# International Journal on Advances in Systems and Measurements

2010 vol. 3 nr. 3&4

The *International Journal on Advances in Systems and Measurements* is published by IARIA.
ISSN: 1942-261x
journals site: http://www.iariajournals.org
contact: petre@iaria.org



Sponsored by IARIA
www.iaria.org

**International Journal on Advances in Systems and Measurements**
**Volume 3, Number 3 & 4, 2010**

- Jean-Henry Morin, University of Geneva - CUI, Switzerland
- Igor Podebrad, Commerzbank, Germany
- Leon Reznik, Rochester Institute of Technology, USA
- Chi Zhang, Juniper Networks, USA

**Sensor Networks**

- Steven Corroy, University of Aachen, Germany
- Mario Freire, University of Beira Interior, Portugal / IEEE Computer Society - Portugal Chapter
- Jianlin Guo, Mitsubishi Electric Research Laboratories America, USA
- Zhen Liu, Nokia Research – Palo Alto, USA
- Winston KG Seah, Institute for Infocomm Research (Member of A*STAR), Singapore
- Radosveta Sokkulu, Ege University - Izmir, Turkey
- Athanasios Vasilakos, University of Western Macedonia, Greece

**Electronics**

- Kenneth Blair Kent, University of New Brunswick, Canada
- Josu Etxaniz Maranon, Euskal Herriko Unibertsitatea/Universidad del Pais Vasco, Spain
- Mark Brian Josephs, London South Bank University, UK
- Michael Hubner, Universitaet Karlsruhe (TH), Germany
- Nor K. Noordin, Universiti Putra Malaysia, Malaysia
- Arnaldo Oliveira, Universidade de Aveiro, Portugal
- Candid Reig, University of Valencia, Spain
- Sofiene Tahar, Concordia University, Canada
- Felix Toran, European Space Agency/Centre Spatial de Toulouse, France
- Yousaf Zafar, Gwangju Institute of Science and Technology (GIST), Republic of Korea
- David Zammit-Mangion, University of Malta-Msida, Malta

**Testing and Validation**

- Cecilia Metra, DEIS-ARCES-University of Bologna, Italy
- Krzysztof Rogoz, Motorola, Poland
- Rajarajan Senguttuvan, Texas Instruments, USA
- Sergio Soares, Federal University of Pernambuco, Brazil
- Alin Stefanescu, University of Pitesti, Romania
- Massimo Tivoli, Universita degli Studi dell'Aquila, Italy

**Simulations**

- Tejas R. Gandhi, Virtua Health-Marlton, USA
- Ken Hawick, Massey University - Albany, New Zealand
- Robert de Souza, The Logistics Institute - Asia Pacific, Singapore
- Michael J. North, Argonne National Laboratory, USA

**International Journal on Advances in Systems and Measurements**
**Volume 3, Numbers 3 & 4, 2010**

**CONTENTS**

# Combining Aggregates of Synthetic Microscale Nanorobots with Swarms of Computer-controlled Flagellated Bacterial Robots to Enhance Target Therapies Through the Human Vascular Network

Sylvain Martel
NanoRobotics Laboratory, Dept. of Computer and Software Engineering, and Institute of Biomedical Engineering,
École Polytechnique de Montréal (EPM), Campus of the Université de Montréal,
Montréal, Canada
E-mail: sylvain.martel@polymtl.ca

*Abstract*—**The field of medical nanorobotics exploits nanometer-scale components and phenomena to enable new or at least to enhance existing medical diagnostic and interventional procedures. The best route for such miniature robots to access the various regions inside the human body is certainly the vascular network which is constituted of nearly 100,000 km of blood vessels. The variations in blood vessels diameters from a few millimeters in the arteries, down to ~4 µm in the capillaries with respective important variations in blood flow velocities, lead to significant challenges in the development of a robot relying on a singe type of propulsion method while being trackable in the human body. This tracking feasibility in a living body was realized experimentally by integrating magnetic nanoparticles (MNP) capable of creating a net field inhomogeneity that could be detected by magnetic resonance imaging (MRI). In such an environment, dipole-dipole interaction between synthetic microscale nanorobots encapsulating MNP can be used to achieve higher magnetophoretic velocities when subjected to a 3D magnetic gradient force generated by an upgraded MRI platform to allow such aggregated nanorobots to travel in the blood circulatory network. Nonetheless, the potential limitations of using an artificial approach especially in the human microvasculature suggests that a biological system such as the molecular motors capable of flagellar propelling thrust force exceeding 4 pN provided by each MC-1 MRI-trackable magnetotactic bacteria swimming as swarms under computer control in blood vessels could enhance targeting efficacy in regions such as tumors. Presently, such artificial and natural approaches are unique and very different from any other methods proposed elsewhere. In this paper, these approaches are described with examples of implementations with the goal of demonstrating the potential advantages of combining them for targeting regions deep in the human body.**

*Keywords-Magnetic nanoparticles; medical nanorobots, magnetic resonance imaging, magnetotactic bacteria, target therapy*

## I. INTRODUCTION

This paper is based on [1] where synthetic microscale nanorobots were compared to flagellated bacterial robots for target intervention in the human vasculature. Here, not only the main differences in both types are compared but there are described in more details with the goal of demonstrating the potential advantages of combining artificial or synthetic microscale robots with natural or biological microscale robots for specific target interventions in the human body. But first, as we refer to such untethered entities or agents as robots, the readers must be aware of what this term means in the context of navigable microscale entities in the blood vessels, considering present technological limits.

There are many opinions of what would define a robot. For instance, several organizations including the Robotics Institute of America (RIA) define a robot as a re-programmable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks. Accordingly, an untethered entity or device capable of programmable motion for the performance of a variety of tasks such as the delivery of therapeutic agents at a specific location in the human body could be viewed as a robot under such a definition. In turn, this definition could be extended to include a microscale robot if its overall dimension would be in the micrometer-range and ideally below 100 µm across. Unlike a nanorobot which can be viewed as a nanometer-scale robot, a microscale nanorobot is defined here as a robot with overall dimensions in the micrometer range that depends on nanometer-scale components (typically less than 100 nm across) to embed characteristics allowing such robot to accomplish its assign tasks. Embedded magnetic nanoparticles (MNP) can be considered as one type of these components allowing propulsion/steering, tracking, and hyperthermic functions to name but the main ones made possible.

Indeed, MNP can be manipulated by an external magnetic field gradient with the capability of disrupting a high intensity local magnetic field creating a net field inhomogeneity that can be picked up by magnetic resonance imaging (MRI). Such signals can then be fed back to an external computer as tracking data to be processed to perform closed-loop navigation or trajectory control along a pre-planned path.

But present implementations for target interventions using MNP alone have major limitations due to the fact that the approach relies on a permanent (or an electro-magnet) that must be located near the targeted region. The higher field intensity towards and very near the external magnet restrict targeting to regions near the skin and when the target is located deeper in the human body, a significant reduction of targeting efficacy is expected. When coupled with the fact that the approach relies on trapping the MNP without trajectory servo-control based on tracking information from the release site and towards the target, the approach seems extremely limited. This observation becomes more evident when we look at the distance between the reachable limits of catheterization and targets reachable through complex microvasculature networks such as the ones near a tumor.

Therefore, when operating in the human vascular network, untethered microscale robots capable of performing such tasks must be capable of directional propulsion and/or steering in a 3D volume while being trackable in the body by a suitable imaging modality such as MRI. When these conditions would be met, effective navigation along pre-planned paths through real-time closed-loop control from the release site typically at the catheterization boundaries to as close as possible to a specific target possibly reachable through the microvasculature, would be feasible.

## II. TRACKING METHOD

Unlike in the cases of most robotic systems, a tracking approach based on direct-line-of-sight is not possible when operating in the vascular network. Among the few imaging techniques available for such application, X-ray or computed tomography (CT) scan, and MRI stand as the two main potential and widely accepted imaging modalities that are presently widely used in clinics and hospitals.

Since the blood vessels represent the routes to be navigated by these microscale robots, it becomes important to image as many blood vessels that are along the planned path using angiography. Presently, X-ray digital subtraction angiography (DSA) still has the best spatial and temporal resolution for imaging blood vessels. However X-ray has poor soft-tissue information and a high radiation exposure, two disadvantages that do not apply with MRI. As such, substantial research efforts attempt to replace X-ray angiography by magnetic resonance angiography (MRA). Despite the recent progresses in MRA, presently DSA is still the preferred and most appropriate technique to image the vascular network.

The fact that DSA could be used to gather data for the pre-navigation/planning phase of the microscale robots does not means that X-ray is the most appropriate imaging to gather real-time tracking information of the robots navigating in the blood vessels. This is a critical issue since the spatial resolution of X-ray although better that MRI is still not sufficient to detect such microscale robots when traveling in smaller diameter vessels (e.g. from the small diameter arterioles to the capillaries). On the other hand, MRI-tracking deep in the human body can be achieved with a proper MRI sequence if the magnetic components embedded in such microscale robots create a sufficiently large local distortion of the magnetic field inside the bore (tunnel) of the MRI scanner. Indeed, the local magnetic field distortion from a magnetic spherical particle for instance can be found at a point $P$ of coordinate $r(x, y, z)$ by that of a magnetic dipole as

$$\vec{B}'(P) = \frac{\mu_0}{4\pi}\left(3\frac{(\vec{m}.\vec{r})\vec{r}}{r^5} - \frac{\vec{m}}{r^3}\right). \tag{1}$$

where $\mu_0 = 4\pi\times10^{-7}$ H·m$^{-1}$ is the permeability of free space such that for a uniformly magnetized object, the dipolar magnetic moment (A·m$^2$) depends on the magnetization saturation of the material and is given by

$$\vec{m} = \frac{4}{3}\pi a^3 \vec{M}_{SAT} \tag{2}$$

where $a$ here represents the radius (m) and $\boldsymbol{M_{SAT}}$ being the magnetization saturation of the material used.

Indeed, in [2] we showed that the susceptibility-based negative contrast in MRI can be used to track such magnetic microscale robots with dimensions as small as 15 μm in the human body. In this case, the same ferromagnetic material with high susceptibility and used for propulsion was used to induce a perturbation in the main magnetic field homogeneity of a 1.5T clinical MRI scanner. This perturbation can be set during the synthesis process of the microscale robots to reach a level much larger than the robot itself. By creating an artifact from the magnetic material embedded in the microscale robot with a size equivalent or slightly larger than a typical MRI voxel (~0.5 × 0.5× 0.5 mm$^3$), the microscale robot will become MRI-trackable.

One method is to use the intra-voxel de-phasing in the Gradient Echo (GE) sequence. This method aims at amplifying the effect of a microscale magnetic object such as a robot that is too small to be visualized in the MR-image or with any other medical imaging modalities.

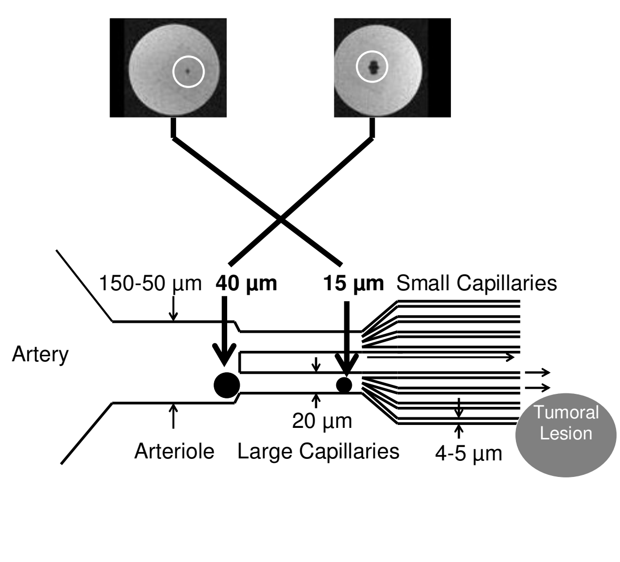


Fig. 1 – Experimental coronal images of a stainless steel bead with an overall diameter of 15 µm imaged using a 1.5 T scanner with a Time Echo (TE) of 30 ms.

## III. PROPULSION AND STEERING

### A. Single Synthetic Microscale Nanorobot

Even if tracking information is gathered from a modern medical imaging modality such as MRI, propelling and/or steering (when blood flow is used for propulsion) MNP sufficiently to direct them effectively along a pre-planned trajectory is not possible or at least extremely challenging due to the amplitude of magnetic gradients being required. Such required gradient amplitude is presently well beyond technological limitations when operating in some regions of the body of a human adult (e.g. inside the torso) because of the distance separating the MNP from the source generating the directional magnetic gradient field. This is also in great part due to the small volume of MNP even when synthesized with materials having the highest magnetization saturation. This is depicted in Eq. 3.

$$\vec{F}_M = R \cdot V \left( \vec{M} \cdot \nabla \right) \vec{B} \cdot \quad (3)$$

The induced magnetic force (N) (Eq. 3) produced by magnetic gradients (T $m^{-1}$) and acting on a ferromagnetic core with a volume $V$ ($m^3$), depends on the duty cycle $R$, i.e. the percentage of time per navigation closed-loop control cycle being dedicated to propulsion (dimensionless) which can be reduced due to longer MRI-tracking acquisition time and/or to allow time for cooling of the coils in particular cases, and the volume magnetization of the core material (A $m^{-1}$). When such magnetization reaches a saturation level ($M_{SAT}$), the highest induced propulsion force per unit volume can be reached for a given ferromagnetic material. This would typically be the case when placed in a high intensity magnetic field such as in the bore or tunnel of a standard 1.5T or higher field clinical MRI system. Furthermore, such gradient force unlike the use of an external magnet is constant within the 3D working space inside the bore of MRI systems.

Nonetheless, our initial studies [3] for the development of what we are referring to as a double-insert, i.e. a set of propulsion coils (to increase gradients beyond the 40 mT/m limitation as found in typical clinical MRI systems) that allow MR-imaging to be performed in a time-multiplexed fashion and designed to be installed inside the bore of a clinical MRI system, would still have limited capabilities. This is mainly due to limitation of modern technologies to dissipate the heat generated by such coils. With an inner diameter capable to accommodate a human adult, our initial studies suggest that the maximum gradient amplitude that could be generated would be ~400-500 mT/m. It is then obvious as stated earlier and as depicted in Eq. 3 that a directional force sufficient to influence the path of independent MNP for achieving efficient targeting, cannot be generated for human subjects unless being restricted only to some body's regions such as the legs or arms for instance. For operating in all regions of the human's body (but without transiting through the blood brain barrier (BBB)), this suggests that a cluster or aggregate of MNP with a sufficiently high magnetization saturation level must be encapsulated within a material as to form an entity or robot that could be sufficient to increase effectively $V$. This would provide an induced magnetic force (Eq. 4) for each robot that would be made of $n$ embedded ferromagnetic cores or superparamagnetic MNP, each with a volume $V$ where $n\,V \leq V_R$ being the volume of each robot, assuming that all embedded MNP are identical.

$$\vec{F}_{MR} = R \cdot \left( \sum_n V \right) \left( \vec{M} \cdot \nabla \right) \vec{B}, \quad nV \leq V_R \cdot \quad (4)$$

Except for transiting through the BBB for targeting inside the brain, such entities do not need to be smaller than approximately half the diameter of the tiniest blood vessels found in humans, i.e. untethered entities should not have an outer diameter much greater than 2 µm and much smaller than 2 µm. Larger diameters would decrease magnetophoretic velocities of such entities in the tiniest vessels due to wall retardation effects while smaller ones would typically lead to a lower charge of magnetic material being embedded. Hence, in order to increase the effective $V$, a relatively large quantity of MNP can be encapsulated within a non-magnetic material. Such synthetic entity or microscale synthetic robot would then have a magnetophoretic velocity expressed as

$$\overrightarrow{v_M} = \frac{\vec{F}_{MR}}{f} \cdot \quad (5)$$

In Eq. 5, $f$ is the friction factor. The value of the friction factor depends on the geometry of the entity being navigated. As stated earlier, the high magnetic field provided by a clinical MRI platform allows us to achieve maximum propulsion force density through magnetization saturation of the MNP while providing an image modality capable of feeding back tracking information to a servo-controller. But when inserted in a high magnetic field, MNP will align to the lines of the magnetic field and maintain the same orientation while operating in such magnetic field. In a clinical MRI system, this homogeneous magnetic field with typical values of 1.5 T or 3.0 T is known as the $B_0$ field. A spherical shape (lack of anisotropy) would then be a good choice since it would provide a constant friction factor that would be independent of the direction of any navigated blood vessels relative to the $B_0$ field. As such, the friction factor for such untethered microscale spherical synthetic robot with embedded MNP would be computed as

$$f = 3\pi \eta \, d_R \,. \qquad (6)$$

In Eq. 6, where $\eta$ is the viscosity of the medium (e.g. 1.0 mPa·s in water at 20 °C), and $d_R$ is the diameter (m) of the spherical robot. Hence assuming that the synthetic microscale robots would have a spherical shape, Eq. 4 can be re-written as

$$\vec{F}_{MR} = R \cdot \left( p\,(4/3)\,\pi \, d_R^{\,3} \right)\left( \vec{M} \cdot \nabla \right) \; \vec{B} \,. \qquad (7)$$

where $p$ is the percentage of ferromagnetic or superparamagnetic material in each untethered microscale spherical robot.

*B. Aggregated Synthetic Microscale Nanorobots*

Considering the aforementioned equations, one can easily deduct that a larger untethered robot based on such method of propulsion will have a higher magnetophoretic velocity (if charged with the same percentage of the same type of MNP) than a smaller robot if the diameter of the blood vessel is large enough to avoid wall retardation effects. But smaller robots would be required to travel in smaller diameter vessels.

One solution to this issue is the use of an aggregation of smaller robots. Indeed, the effective $V$ can also be increased with an aggregation of such microscale robots. The level of propulsion/steering force for such aggregation will depend on the coupling force between neighboured robots or untethered entities. When in a high intensity magnetic field, each untethered microscale robot generates a local magnetic dipole. It is this dipole-dipole interaction between neighboured robots that maintains such aggregation. A strong attracting force through strong dipole-dipole interactions between the microscale robots will help maintaining the entities agglomerated while effectively increasing the volume of magnetic material, leading to higher magnetophoretic velocities. But more loosely coupled interactions caused by a lower level of dipole-dipole interactions and/or the use of surfactants or a surface acting like a surfactant (potentially caused by the addition of some types of functionalized molecules), will allow such aggregate to reconfigure when transiting between various vessels geometries such as transiting from larger to smaller diameter vessels. But such lower interactions may result in unexpected or unwanted breakages when encountering obstacles or forces such as the ones created by vortices, which may result in several smaller aggregates, each having lower magnetophoretic velocities than what would be possible with a single larger aggregate. On the other hand, when the interacting forces are too high, an unwanted embolization may occur when transiting from a larger diameter vessel to a smaller one, preventing such aggregate to pursue its course. Therefore, the right compromise in the level of dipole-dipole interacting forces must be set during the synthesis of such microscale robots, taking into account both physiological and technological characteristics. To help in the design or synthesis of such aggregated microscale synthetic robots, Eq. 8 is used to calculate the dipole-dipole interaction energy $E_D$ between two neighboured robots A and B with respective dipoles $\mu_A$ and $\mu_B$. Such interaction depends on not only on the relative orientations of the dipoles but also on their orientation with respect to the vector $r_{AB}$ joining the center of the two dipoles.

$$E_D = -\frac{\mu_0}{4\pi}\left( \frac{(\mu_A \cdot r_{AB})(\mu_B \cdot r_{AB})}{r_{AB}^{\;5}} - \frac{\mu_A \cdot \mu_B}{r_{AB}^{\;3}} \right), r_{AB} = \left\| r_{AB} \right\| \qquad (8)$$

Since the lowest energy configuration corresponds to the two magnetic moments aligned head-to-tail, neighboured (i.e. closed enough for $E_D$ to be non-negligible) microscale synthetic robots when in a MRI system, will tend to form needle-like aggregates with an elongated axe being oriented in the direction of the $B_0$ field. This phenomenon is taken into consideration with regard to the angle between the $B_0$ field and directional motion of the aggregate.

*C. Self-propelled Bacterial Nanorobots*

A self-propelled entity refers here as an entity that can propel itself without any propulsion force produced by an external source. Flagellated bacteria and more specifically magnetotactic bacteria (MTB) [4] acting under computer control can be described as self-propelled nanorobots [5] since they rely on nanometer-scale components to act as a robot. One example is the chain of magnetite nanoparticles (magnetosomes) embedded in the cell of the MTB. Such chain can act as a miniature magnetic compass needle [6]. When a directional torque is induced on such a chain from an electro-magnetic field under computer control, precise directional control of polar MTB can be achieved. This is referred here to as magnetotaxis [7] control. Our studies showed that the MC-1 MTB is presently the most appropriate bacteria for such target applications. The same magnetosomes can also be used for tracking purpose as explained in Eq. 1. The self-propulsion system of each bacterium in the form of two flagellar bundles allows efficient propulsion in low Reynolds regime without the limitation of gradient forces generated from an external source.

## IV. BACTERIAL PROPULSION – EXPERIMENTAL RESULTS

The advantage of bacterial propulsion over the gradient-based propulsion mentioned earlier becomes evident in low Reynolds regime and in the microvasculature. Again, from Stokes equation, we have a thrust force (N) provided by the two flagellar bundles on the back of the spherical cell with a radius $a_B$ (m) being calculated from observed terminal velocities as

$$\overrightarrow{F_T} = \frac{\overrightarrow{v_T}}{3\pi \eta \, a_B} \cdot \quad (7)$$

From swimming velocities of the MC-1 cells recorded in water at room temperature, the thrust force was estimated at more than 4.0 pN (~10 times the thrust force provided by other well known species of flagellated bacteria). To reach the tumoral lesion, capillaries with a diameter as small as ~4 µm can be expected. Hence, the maximum radius of a spherical synthetic microscale robot would be limited by the wall retardation effect being expressed as

$$\frac{v}{v_\infty} = \left( \frac{1-\lambda}{1-0.475\lambda} \right)^4 \quad (8)$$

where $v$ and $v_\infty$ is the velocity of the spherical synthetic robot or the MC-1 bacterium in the capillary and in open space respectively, and $\lambda = d/d_C$ which is the ratio between the diameter of the synthetic robot or the cell of the MC-1 bacterium and the inner diameter of the capillary respectively. In the capillaries, the optimal ratio for an untethered synthetic microscale robot propelled by magnetic gradient is ~0.5. Hence, for the smallest diameter capillaries $d_C = 4\times10^{-6}$ m, the largest synthetic microscale spherical robot would have $a_R = 1\times10^{-6}$ m. To maximize the effective volume of magnetic material being embedded, $V_R$ needs to be maximized while maintaining the ratio at ~0.5, which suggests that *min* $a_R = 1\times10^{-6}$ m since nothing is gained with a smaller volume except for transiting through the BBB. Considering Eq. 7, when loaded with 100% FeCo MNP for achieving maximum magnetization saturation level, it is obvious that a single synthetic nanorobot with a diameter of 2 µm for travelling in the tiniest blood vessels with the limitation in generating magnetic gradients at the human scale would be much less effective than the minimum thrust force of 4 pN provided by a single 1 to 2- µm in diameter MC-1 MTB. But our preliminary results suggest that when operating in the vascular network at body temperature, unlike for a synthetic version, their motility after a period of time $t$ (in minutes) and hence their swimming speed will be affected according to Eq. 9.

$$v_{B37} = 0.09\, t^2 - 8.10\, t + v_{MTB} \cdot \quad (9)$$

where the initial average swimming velocity $v_{MTB} = 187.85$ µm/s prior to be injected in blood. Hence, the advantage of this particular type of bacterial propulsion is true but of a limited time (approx. 40 minutes) especially in the microvasculature where $V$ is limited due to space constraints and where the formation of aggregates is not possible. On the other hand, our studies showed that in larger vessels especially in arteries and in larger arterioles, synthetic microscale nanorobots will perform better since thrust force above the bacterial thrust force can be achieved.

## V. BACTERIAL STEERING

Unlike the synthetic microscale robots that require higher gradients as their volume decreases in order to induce a sufficiently high propulsion/steering force for targeting or navigation purpose, the fact that the magnetic field for a polar MTB is only used for directional control and not to provide a propelling force, translates into the need for a much lower magnitude of magnetic field with makes the navigation of smaller untethered robots in the human body, technologically possible using much less power. Indeed, the magnetotactic bacteria of type MC-1 and depicted in Fig. 2 is an example of a biological microscale robot where the flagella bundles are the propulsion (propulsive) system and the chain of membrane-based nanoparticles known as magnetosomes embedded in the cell implements an embedded steering system by acting like a miniature magnetic compass needle that can be oriented by inducing a torque from a directional magnetic field.

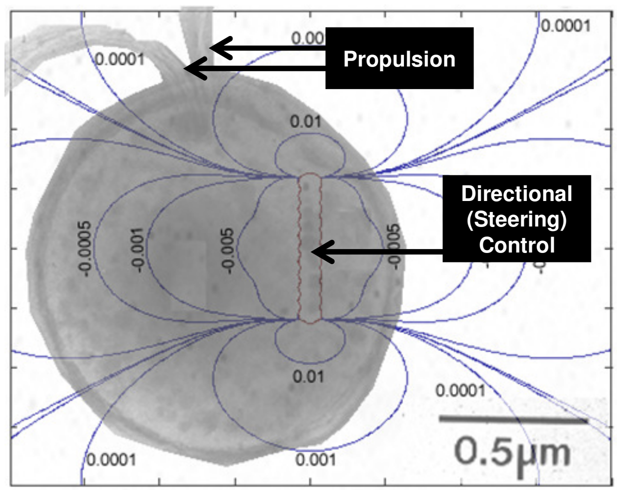



Fig. 2 – Electron photograph of a MC-1 magnetotactic bacterium acting as a biological microscale robot with the two flagella bundles and the chain of magnetosomes used for propulsion and steering respectively.

Indeed, here directional control is performed by inducing a directional torque $\boldsymbol{T}$ by applying a directional magnetic field $\boldsymbol{B}$ as described by the following equation:

$$\vec{T} = V \cdot \vec{M} \times \vec{B} \,. \quad (10)$$

Since for a robot with the same volume and magnetization, the magnitude of the magnetic field required is much less for generating a directional torque compared to a directional displacement (propulsion) force, directional control of such robot capable of providing its own propulsion (propelling) force would require significantly less power.

## VI. COMBINING MAGNETIC AND BACTERIAL PROPULSIONS

Many targets such as tumoral lesions located deeper in the body can be accessed by travelling through a microvascular network including the angiogenesis network. But because of the size of the blood vessels, the injection process is typically performed in larger diameter vessels, typically in an artery. As such, combining magnetic-based propulsion with self-propelled bacterial carriers can lead to enhanced targeting efficacy when travelling through the microvasculature is essential, which is most often the case in tumor targeting. Indeed, since bacterial propulsion based on the MC-1 MTB has a minimum thrust force of 4 pN which is approximately 10 times the thrust force that has been measured for other well known species of flagellated bacteria, suggests that it could be used as a complementary means of propulsion when targeting deep in the vasculature. In other words, such bacterial propulsion may become more effective at a location where magnetic propulsion becomes ineffective since it does not rely on an external source as magnetic propulsion does. The latter can be explained as stated earlier by the fact that the smaller diameters of the blood vessels prevent the use of synthetic microscale nanorobots with a sufficiently large volume of embedded magnetic material for sufficient propulsion within known technological limitations.

But there is another important advantage of using such flagellated bacteria when operating in the microvasculature. Indeed, such bacterial robots have what is referred to here as path finding (PF) capability. When a directional magnetic field is used (outside the tunnel of a MRI scanner) to point in the desired direction of motion for the bacterial robots (e.g. towards a target such as a tumor), the latter will move under the influence of the induced torque generated by the same directional field. When a MTB reaches a sufficiently wide obstacle (which is often the case in the microvasculature, e.g. vessel's walls), it will not typically remain at the location where it reached the obstacle as a synthetic microrobot will until the direction of the magnetic field is changed but instead, it will look for a path that will lead toward the direction of the magnetic field. Since the small blood vessels found in the microvasculature cannot presently be imaged due to a lack of spatial resolution of all existing medical imaging modalities, knowing in which direction the induced force should be applied to navigate synthetic microscale robots in such chaotic maze of blood vessels without such image information is practically not possible. This is why the use of bacterial microscale robots can make a huge difference in targeting efficacy when operating in such physiological environments.

Nonetheless, to take advantage of this PF capability as well as the enhanced propulsive force provided by MTB in the microvasculature, such bacteria loaded with therapeutic agents must be carried from the catheterization boundary in an artery to the entry of the microvascular network. As such, special micro-carriers based on magnetic propulsion and capable of encapsulating such MTB must be used. The general concept is depicted in Fig. 3.

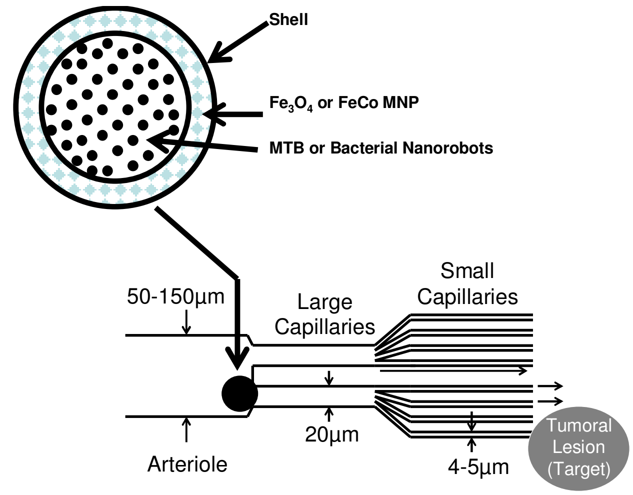



Fig. 3 – Diagram showing the general concept of encapsulating MTB to transport them towards the microvasculature

The synthesis of such carriers with various diameters down to approximately 10 µm and containing MC-1 bacteria capable of free swimming inside the shell as depicted in Fig. 3 has be done successfully by our research group. But more experimental data needed to be recorded before we can validate the idea of transporting drug-loaded MTB. Nonetheless, preliminary experimental results suggest that this concept is very promising.

## VII. REDUCTION OF THE BLOOD FLOW

Another complementary strategy that may enhance targeting in deep regions is to increase the effective propulsive force of MTB in the microvasculature by reducing temporarily the blood flow with the use of synthetic polymorphic microscale robots. Polymorphic microscale robots have the ability to change forms or volumes. For instance, such a robot could travel in a specific blood vessel with a lower overall volume and use a higher volume to create a temporary embolization at a specific site that would eliminate or at least reduce the blood flow to help the MTB navigating more efficiently.

Our first prototypes [8] of such polymorphic microscale robots were made of biocompatible N-isopropylacrylamide (PNIPA) hydrogel. Such hydrogel reduces its size in response to an elevation of temperature above a specific threshold. This temperature threshold referred to as the lower critical solution temperature (LCST) is adjusted slightly above 37$^o$C by the addition of several monomers. The local heat responsible to trigger the volume change is generated by biocompatible ~20 nm single-domain superparamagnetic Iron-Oxide ($Fe_3O_4$) MNP (the use of FeCo MNP could also be envisioned) embedded into the

PNIPA structure and are also used for propulsion and steering. The characteristics of these MNP are adjusted to allow the energy from a magnetic field to drive the magnetic moments capable of overcoming the thermal energy barrier and to allow a rotation and an alignment with the direction of the same magnetic field. After removing the external magnetic field, magnetic moments do not relax immediately, but after a short delay before returning to their original random orientation. During this process known as the Néel relaxation, energy is released in the form of heat that is used to rise the temperature inside the hydrogel-based microscale robot. Several microscale polymorphic robots were tested successfully using a 4 kA.m$^{-1}$ magnetic field modulated at 160 kHz. This approach is summarized schematically in Fig. 4.

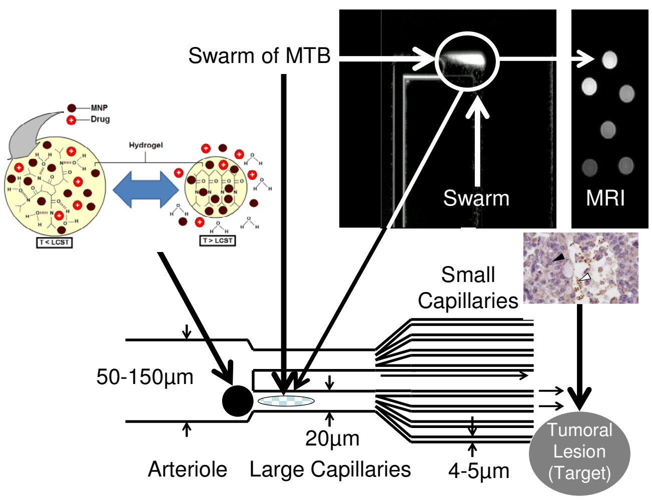



Fig. 4 – Schematic of the use of polymorphic microscale robots to reduce the blood flow in order to enhance targeting using a swarm of bacterial microscale robots

## VIII. Conclusion

An aggregate of synthetic microscale robots will have higher propulsion performance in larger vessels. On the other hand, MC-1 bacterial propulsion proves to be superior in the microvasculature. Combining both approaches should lead to optimum targeting performance in regions located deeply in the human vascular network. One promising approach presented here is to encapsulate self-propelled bacteria acting as biological microscale robots operating under the control of an external computer into special micro-carriers being propelled by magnetic gradients produced by an upgraded MRI scanner. This combination allow complementary propulsion methods necessary to target more efficiently specific regions located deeper and accessible by transiting through the microvasculature after being introduced in the body through an artery. Another complementary approach is based on polymorphic synthetic microscale robots capable of changing volume to produce temporary embolization that would allow a reduction of the blood flow in smaller diameter capillaries and hence, potentially help achieving enhanced targeting efficacy. Although more types of microscale robots could be envisioned, this paper already showed the potential advantages of combining aggregates of synthetic and biologic microscale robots for specific medical interventions such as in cancer therapies.

## Acknowledgment

This project is supported in part by the Canada Research Chair (CRC) in Micro/Nanosystem Development, Fabrication and Validation and grants from the National Sciences and Engineering Research Council of Canada (NSERC), the Province of Québec, the Canada Foundation for Innovation (CFI), and US Grant Number R21EB007506 from the National Institute Of Biomedical Imaging And Bioengineering. The content is solely the responsibility of the authors and does not necessary represent the official views of the National Institute Of Biomedical Imaging And Bioengineering or the National Institutes of Health.

## References

[1] S. Martel, “Aggregates of synthetic microscale nanorobots versus swarms of computer-controlled flagellated bacterial robots for target therapies through the human vascular network,” *The Fourth Int. Conf. on Quantum, Nano and Micro Technologies (ICQNM 2010)*, St. Marteen, Netherlands Antilles, Feb. 10-16, 2010

[2] N. Olamaei, G. Beaudoin, G. Cheriet and S. Martel, "MRI visualization of a single 15 µm navigable imaging agent and future microrobot," *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Buenos Aires, Argentina, Aug. 31 – Sept. 4, 2010

[3] S. Martel, O. Felfoul, J.B. Mathieu, A. Chanu, S. Tamaz, M. Mohammadi, M. Mankiewicz, and N. Tabatabaei, “MRI-based nanorobotic platform for the control of magnetic nanoparticles and flagellated bacteria for target interventions in human capillaries,” *International Journal of Robotics Research (IJRR), Special Issue on Medical Robotics*, vol. 28, no. 9, pp. 1169-1182, Aug. 2009

[4] R. P. Blakemore, “Magnetotactic bacteria,” *Science*, 190, pp. 377-379, 1975

[5] S. Martel, M. Mohammadi, O. Felfoul, Z. Lu, and P. Pouponneau, “Flagellated magnetotactic bacteria as controlled MRI-trackable propulsion and steering systems for medical nanorobots operating in the human microvasculature,” *International Journal of Robotics Research (IJRR)*, vol. 28, no. 4, pp. 571-582, April 2009

[6] R. B. Frankel and R. P. Blakemore, “Navigational compass in magnetic bacteria,” *J. of Magn. and Magn. Materials*, 15-18 (3), pp. 1562-1564, 1980

[7] H. Debarros, D. M. S. Esquivel, and M. Farina, “Magnetotaxis,” *Sci. Progr.*, 74, pp. 347-359, 1990

[8] N. Tabatabaei, J. Lapointe and S. Martel,"Microscale hydrogel-based computer-triggered polymorphic microrobots for operations in the vascular network," *3rd IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, Tokyo, Japan, Sept. 26-30, 2010

# Study of the Boundary Conditions of the Wigner Function Computed by Solving the Schrödinger Equation

Andrea Savio* and Alain Poncet†
Lyon Institute of Nanotechnology – INSA Lyon
7 avenue Jean Capelle, 69621 Villeurbanne, France
*andrea.savio@insa-lyon.fr †alain.poncet@insa-lyon.fr

***Abstract*—In this work, we compute the Wigner distribution from wavefunctions that are generated by solving the Schrödinger equation. Our goal is to propose an avenue of research that may help better understand certain limitations of deterministic Wigner transport equation solvers, such as negative electron densities or limited charge drops in presence of potential barriers. We evaluate the numerical accuracy required by the Schrödinger solver to compute the Wigner function and compare the performance of an analytic and a numerical solver applied to a constant potential profile, as well as to single- and double-barrier one-dimensional structures. Then, we investigate how the Wigner function boundary conditions vary in these same structures as the contact length increases. We also investigate the range of the wave vector grid required to accurately compute the charge from the Wigner function. Finally, we carry out the same study on biased structures.**

***Keywords*—Quantum transport, Schrödinger equation, Wigner function**

## I. Introduction

The constant drive toward increasing integration densities is pushing the size of electronic devices down, ever closer to the nanometer scale. As an example of this trend, the oxide barrier in ULSI MOS transistors was expected to shrink to a thickness below one nanometer by the 45 nm technology node, prior to the introduction of high-permittivity dielectrics. Another example is the channel length of this same type of transistors, which is expected to drop below 10 nm within the next few technology process generations, according to the current ITRS roadmap [2].

As the size of electronic devices approaches the nanometer scale, quantum phenomena begin to affect the charge carriers' distributions and currents, and TCAD simulators must be capable of accounting for these phenomena in order to model device operation accurately. Instead, current commercial simulation software mostly implements classical models such as the drift-diffusion, thermo- and hydrodynamic ones [3]. These models are derived from the phase-state Boltzmann Transport Equation (BTE) and do not take into account the wave nature of charge carriers. Quantum effects are in general treated only tangentially, to simulate parasitic phenomena such as tunneling currents and confinement levels. In order to be capable of accurately simulating next-generation electronic devices, TCAD software needs to implement full-quantum models. This would enable engineers not only to better predict and characterize parasitic effects in current devices, but also to explore innovative quantum-based designs, such as Resonant Tunneling Diodes (RTD) and quantum dots.

The Schrödinger Equation (SE) is the starting point for a number of approaches that model quantum phenomena. In its one-dimensional (1D) transient form, this equation represents carriers as wavefunctions $\psi(x,k,t)$ of energy $E(k)$, which propagate through a lattice potential energy $U(x)$. $x$ indicates the real space and $k$ the wave vector space. The carriers are given an effective mass $m^* = m_r m_0$, where $m_r$ is the relative mass and $m_0$ the electron mass in vacuum. The SE is thus given by:

$$i\hbar\frac{\partial}{\partial t}\psi + \frac{\hbar^2}{2m^*}\frac{\partial^2}{\partial x^2}\psi = U\psi \qquad (1)$$

$\hbar$ is the reduced Planck constant. Although the SE can be solved analytically or numerically through a number of different schemes, it remains ill-suited to simulate carrier transport. A major shortcoming is that it is difficult to match the electron wavefunction to measurable physical quantities at the boundaries of a system. It is also problematic to account for parasitic phenomena such as carrier-carrier interactions.

One way to address these shortcomings is to use the Wigner Function (WF) instead of the SE to compute charge densities and currents. The WF is a quasi phase-space distribution function that is obtained by solving the Wigner Transport Equation (WTE), which is itself derived from the SE. The WTE was first studied by Wigner [4], and was implemented numerically much later by Kluksdahl, to simulate quantum tunneling [5], [6], and by Frensley, to study a 1D RTD device [7], [8]. Frensley's implementation uses a first-order differentiation scheme and assumes a constant effective mass across the structure. Higher-order schemes were later studied by Jensen and Buot [9]–[14], while Tsuchiya [15] and Gullapalli [16], [17] applied a varying effective mass. Implementations on RTD devices are also studied by Miller [18] and Wu [19]. Biegel compares various differentiation and self-consistency schemes and applies them to the simulations of RTD devices [20]. Grubin looks at the resolution of the transient WTE [21], while Nedjalkov analyzes the issue of interactions [22]. Yamada studies a 3D mixed self-consistent scheme applied to a silicon nanowire transistor, by solving the SE across the device's cross-section and the WTE along the transport direction, using a differentiation scheme up to the third order [23]. Finally, Kefi-Ferhane simulates a thin 2D MOS transistor by applying a WTE solver along the channel and a Schrödinger solver perpendicularly [24].

In this paper, we discuss a number of issues that we encountered when implementing the 1D deterministic numerical WTE solver described by Frensely. In order to better understand these issues, we study a method to compute the WF directly from the SE, rather than by solving the WTE. We hope that this approach may give us better insight into the nature of the WF and help us in the future in addressing the problems encountered. In this paper, we present some significant initial results, as we look at the WF boundary conditions in unbiased and biased structures and estimate the minimum contact length that has to be applied to a device in a simulation. In addition, we investigate the minimum range that has to be used for the wave vector mesh in order to accurately compute carrier densities from the WF.

## II. Derivation of the Wigner equation

The WTE is derived from the SE by calculating the Density Matrix Function (DMF) and then carrying out a variable change and a Fourier Transform (FT). In the formulae that follow, the transient nature of the wavefunction is implied. The DMF $\rho(r,s)$ is derived by correlating the wavefunction on $(r,s)$ couples of points in real space. In the case of a 1D structure with entry and exit contacts (the “Emitter” and “Collector” respectively), this gives [8]:

$$\begin{aligned}\rho(r,s) =& \\ &\frac{2m^*_{\text{Emitter}}k_BT}{h^2}\int_0^\infty \psi(r)\overline{\psi(s)}f_{\text{FD}}(E(k))\,dk \\ &+\frac{2m^*_{\text{Collector}}k_BT}{h^2}\int_{-\infty}^0 \psi(r)\overline{\psi(s)}f_{\text{FD}}(E(k))\,dk\end{aligned} \tag{2}$$

$h$ is the Planck constant, $k_B$ the Boltzmann constant, and $T$ the absolute temperature, which is set to 300 K in all the simulations presented in this work. The first term of the formula represents wavefunctions incident at the emitter, i.e., with a positive wave vector $k$, while the second represents wavefunctions incident at the collector, i.e., with a negative wave vector. The wavefunctions are weighed by the carrier energy spectrum density, which is given by a Fermi-Dirac Distribution (FDD) $f_{\text{FD}}(k)$ integrated over transverse momenta:

$$f_{\text{FD}}(k) = \ln\left[1+\exp\left(-\frac{E(k)-E_F}{k_BT}\right)\right] \tag{3}$$

$E_F$ is the Fermi energy level at the contact. Assuming a parabolic band, the carrier energy $E$ is given by:

$$E(k) = \frac{\hbar^2k^2}{2m^*_{\text{Contact}}} \tag{4}$$

By applying the SE to the DMF and then carrying out the following variable change:

$$\begin{aligned}&r = x+y/2 \quad , \quad s = x-y/2 \\ &u(x,y) = \rho(x+y/2, x-y/2)\end{aligned} \tag{5}$$

the Liouville - von Neumann Transport Equation (LNTE) is derived [8], [25]:

$$\frac{\partial}{\partial t}u - i\frac{\hbar}{m^*}\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right)+\frac{i}{\hbar}\left[U\left(x+\frac{y}{2}\right)-U\left(x-\frac{y}{2}\right)\right]u = 0 \tag{6}$$

The WTE is derived by applying a FT to the LNTE [8]:

$$\begin{aligned}\left(\frac{\partial f_W}{\partial t}\right)_{\text{Scattering}} &= \underbrace{\frac{1}{2\pi\hbar}\int_{-\infty}^{\infty}[\delta U(x,k-k')f_W]\,dk'}_{\text{Drift term}} \\ &+ \underbrace{\frac{\hbar k}{m^*}\frac{\partial f_W}{\partial x}}_{\text{Diffusion term}} + \underbrace{\frac{\partial f_W}{\partial t}}_{\text{Transient term}}\end{aligned} \tag{7}$$

In this formula, $f_W(x,k,t)$ designates the WF. The formula includes a scattering term that accounts for carrier interactions. Note that the FT transforms the space variable $y$ into the wave vector $k$. $\delta U(x,k-k')$ is the non-local potential, given by [8]:

$$\delta U(x,k) = 2\int_0^\infty \sin(ky)\left[U\left(x+\frac{y}{2}\right)-U\left(x-\frac{y}{2}\right)\right]\,dy \tag{8}$$

The WF can be either computed by solving the WTE, or calculated directly from the DMF [4], [8]:

$$f_W(x,k) = \int_{-\infty}^{\infty} e^{-iky}\rho\left(x+\frac{y}{2}, x-\frac{y}{2}\right)\,dy \tag{9}$$

The charge $n(x)$ can be computed from either the DMF or the WF [8]:

$$n(x) = \rho(x,x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f_W(x,k)\,dk \tag{10}$$

## III. WTE implementation issues

At present, commercial simulation software typically deals with quantum parasitic phenomena by applying ad-hoc models to the areas of a device that are most affected. As the device size decreases and these areas become relatively larger, full-quantum simulators might eventually come to replace current classical models. Even after repeated shrinks, however, some regions in a device might still behave classically (e.g., the contacts), and quantum models should therefore be capable of smoothly handling the transition between quantum and classical transport.

In the specific case of WTE solvers, when simulating sufficiently large devices with negligible quantum effects, the values of the charge and the current should be consistent with those obtained by solving the BTE. On smaller devices, as quantum effects begin to appear, the simulated characteristics should be consistent with those yielded by other, comparable quantum models.

Fig. 1 tests these consistency constraints by showing an edge-case classical structure that is sufficiently small to let quantum effects begin to appear. The simulated structure is an abrupt silicon $N^+P^+N^+$ double junction. Each region is 15 nm long; the device is not biased. The figure displays the electron

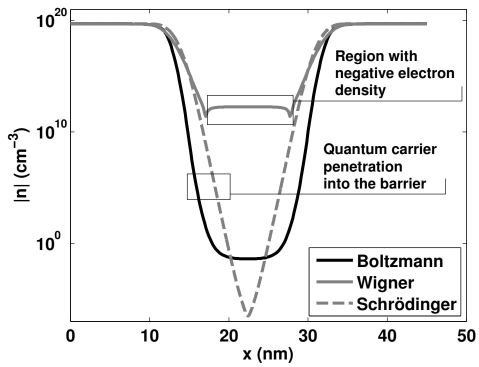



Fig. 1. Absolute-value electron densities obtained on a $N^+P^+N^+$ silicon structure with self-consistent Boltzmann, Wigner and Schrödinger solvers. The dopant profiles are abrupt and each region is 15 nm long. $N^+ = P^+ = 5{\times}10^{19}$ cm$^{-3}$.

density plots obtained by solving the WTE, the BTE and the Schrödinger equations self-consistently with the same mesh numerical parameters. The details concerning the Schrödinger solver implementation are discussed in Section IV. For the time being, note that this solver should be considered as the one providing the most accurate results when quantum effects are taken into account.

The preponderant classical nature of the simulated device can be seen from the large drop in the electron density (over 20 decades) given by both the BTE and Schrödinger solvers. Quantum effects result in a less abrupt slope in the middle-region valley in the Schrödinger plot compared to the BTE one. This can be explained by taking into account the penetration of the electron wave packet into the $P^+$ barrier. The slope in the WTE plot is consistent with the Schrödinger one, which indicates that quantum effects are correctly accounted for. However, the WTE plot shows a glaring artifact, as the electron density in the middle of the valley takes negative values, which has no physical sense. Moreover, the minimum electron density in the WTE plot is 10 decades higher than that in the Schrödinger one.

The shape of the WTE plot seems to suggest that the WTE solver cannot handle a drop in the charge by more than a few decades. This is consistent with the literature on the WTE, which mostly presents simulations displaying limited drops in charge and current. For example, the peak/valley current ratio simulated in RTD devices is generally lower than one decade [20], while the nanowire transistor simulated by Yamada has an $I_{\text{On}}/I_{\text{Off}}$ ratio of about 100 [23]. If the WTE solver is indeed accurate only for small variations of the simulated electric macroscopic quantities, it could be problematic to use it to simulate devices with a mixed quantum and classical character.

Investigating these issues with WTE solvers is somewhat problematic, due to memory constraints. Indeed, as the WTE contains an integral term, it is implemented numerically as a block matrix [20], where the number of non-zero coefficients increases with the square of the mesh density in the wave vector space. The rapidly-growing memory footprint thus limits the resolution at which the WF can be calculated, as well as the ability to investigate its properties. By computing the WF from the SE, rather than from the WTE, we are able to reduce this footprint; this enables us to define much denser meshes than those used in a WTE solver, and thus to thoroughly investigate the issue of boundary conditions in single- and double-barrier, classical and quantum 1D structures.

## IV. Computation of the WF with a Schrödinger solver in unbiased structures

In order to test whether the WTE solver can accurately compute high charge and current drops, one could try to increase the $x$- and $k$-grid resolutions ($N_x$ and $N_k$ respectively). However, memory constraints rapidly limit this technique, as the size of the drift term matrix increases proportionally to $N_x N_k^2$ [20]. In fact, it would be more efficient to apply the solver to only a small region of interest where quantum phenomena take place, e.g., between the two ends of a potential barrier. The memory overhead of meshing the contact regions could thus be avoided. However, this technique poses the problem of what boundary conditions to apply to the solver as, according to Frensley, a FDD can be used only if the boundaries are distant from the quantum region [8]. Moreover, to the best of our knowledge, no study has yet been conducted to evaluate the minimum contact length at which equilibrium conditions can be applied.

In order to investigate the matter of the minimum contact length, we implement a solver that computes the WF directly from the SE. Its ultimate purpose in future studies will be to apply its solution to the WTE as a boundary condition. This section focuses on developing an accurate implementation of this solver, and on assessing its limitations. The Schrödinger solver algorithm is implemented as follows: first, a number of wavefunctions are cast into a given potential profile; then (2) is applied to compute the DMF, and finally (9) is used to calculate the WF. The solver thus works on three different 1D grids to account for the $x$, $y$ and $k$ variables. $N_x$, $N_y$ and $N_k$ denote the respective grid resolutions. The details of the numerical implementation, including grid spacing, are discussed further at the end of this section.

The solver has been applied to three potential profiles at zero bias: (a) a constant potential over a length of 50 nm; (b) one with two contacts separated by a 7 nm-thick rectangular barrier and (c) one with two contact regions and two 1 nm-long barriers separated by a 3 nm well. The optimal length of the contact regions is discussed in Section VI. In all three cases, the Fermi level is set equal to the conduction band energy at both contacts. For the constant potential profile, a relative mass of 0.5 is taken. For the other two, two different combinations are tested: (a) a relative mass of 0.5 and a barrier height of 1.5 eV, which are representative of a generic silicon/nitride structure, and (b) a relative mass of 0.067 and a barrier height of 0.3 eV, which are representative of a generic III-V structure. The solver used in this work is not self-consistent, as the purpose of this study is simply to compute the WF from a given potential profile. However, self-consistency can be implemented, as for the plot in Fig. 1.

Fig. 2 illustrates how wavefunctions are computed. Starting at the emitter contact, plane wavefunctions $\psi_{\text{Emitter}}(x, k)$ of the form:

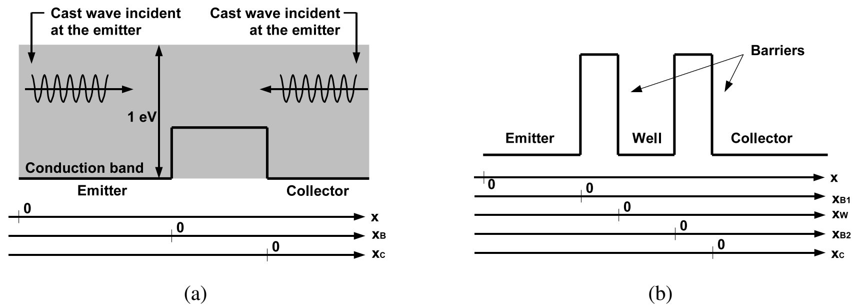



Fig. 2. Schematic view of the SE solver implementation on a single- (a) and a double-barrier (b) potential profile. Wavefunctions are cast into the structure from the emitter and collector contacts, within an 1 eV-wide range from the conduction band upwards. The $x$, $x_B$, $x_W$ and $x_C$ axes, as defined in (11) and (13)-(15), are displayed and their origins are marked.

$$\psi_{\text{Emitter}}(x,k) = e^{ikx} + b_{\text{Emitter}}(k)e^{-ikx} \tag{11}$$

are cast into the device, where $k$ is the positive wave vector and the carrier energy $E$ is given by (4). These wavefunctions are used to compute the first integral term in (2). Each wavefunction contains a normalized incident component with a positive wave vector and a reflected component with a negative wave vector and a complex reflection coefficient $b_{\text{Emitter}}(k)$. For each incident wave vector, the wavefunction is computed at each node on the $x$-grid through either an analytic or a numerical scheme.

The analytic scheme applies the transfer-matrix method. In short, this method is composed of four steps: first, the structure is divided into separate regions, namely the contacts, the barriers and the well. Then, the wavefunction is calculated symbolically in each region by solving the SE. As the SE is a second-degree differential equation relative to space, its solution in region $i$ is the linear combination of two functions $f_i(x,k)$ and $g_i(x,k)$, and has the form:

$$\psi(x,k) = a_i(k)f_i(x,k) + b_i(k)g_i(x,k) \tag{12}$$

$a_i(k)$ and $b_i(k)$ are the wavefunction coefficients in the region. In the third step, these coefficients are evaluated by setting up a system of two equations at each interface between adjacent regions; these equations express the continuity of the wavefunction and of its first derivative at the interface. When all interfaces are accounted for, solving the overall system yields the wavefunction coefficients in each region. Finally, the wavefunction can be evaluated at all points.

The potential profile in each region needs to be regular enough so that there exists a symbolic solution for the SE. This is possible for instance if the potential is constant, as discussed further in this section, or if it varies linearly, as seen in Section VII; in the latter case, the solutions are given by Airy functions, which can be evaluated to a high precision with appropriate numerical libraries. With a flat or linear potential profile, the wavefunction can be calculated symbolically at all points in the structure, and to evaluate it numerically as a last step.

In the barriers, for a constant potential $U(x) = U_B$ the solutions of the SE take the form:

$$\psi_{\text{Emitter}}(x_B,k) = \begin{cases} a_{\text{Barrier}}(k)e^{ik_Bx_B} + b_{\text{Barrier}}(k)e^{-ik_Bx_B} & E(k) > U_B \\ a_{\text{Barrier}}(k)x_B + b_{\text{Barrier}}(k) & E(k) = U_B \\ a_{\text{Barrier}}(k)e^{k_Bx_B} + b_{\text{Barrier}}(k)e^{-k_Bx_B} & E(k) < U_B \end{cases} \tag{13}$$

In this formula, $k_B = \sqrt{(2m^*|E(k)-U_B|)}/\hbar$ and $x_B$ is the $x$ coordinate with the origin set at the left foot of the barrier. In the case of a double barrier, the solution in the well is:

$$\psi_{\text{Well}}(x_W,k) = a_{\text{Well}}(k)e^{ikx_W} + b_{\text{Well}}(k)e^{-ikx_W} \tag{14}$$

The origin of $x_W$ is set at the left end of the well. Finally, the solution at the collector contact is of the type:

$$\psi_{\text{Collector}}(x_C,k) = a_{\text{Collector}}(k)e^{ikx_C} \tag{15}$$

This wavefunction has no negative wave vector, as no wave is incident at the collector. The origin of $x_C$ is set at the start of the collector region.

The numerical scheme solves the SE by applying the Numerov method [26]. Once again, the process can be divided into four steps. The computation starts at the collector, where the wavefunction has only one component. This component is temporarily normalized, i.e., $a_{\text{Collector}}$ is set equal to 1. This makes it possible to use (15) to evaluate the wavefunction at the first two nodes in the collector region:

$$\begin{aligned} \psi_{\text{Collector Normalized}}(x_C = 0) &= 1 \\ \psi_{\text{Collector Normalized}}(x_C = \Delta x) &= e^{ik\Delta x} \end{aligned} \tag{16}$$

Then, from these two data points, the Numerov method is applied to compute the wavefunction backwards into the structure. When the emitter contact is reached, the following step consists in evaluating the wavefunction coefficients $a_{\text{Emitter}}$ and $b_{\text{Emitter}}$ in this region. At the emitter boundary, the function and its first derivative are given by (11) and have the form:

$$\begin{aligned} \psi_{\text{Emitter}}(x = 0) &= a_{\text{Emitter}} + b_{\text{Emitter}} \\ \frac{\partial\psi_{\text{Emitter}}}{\partial x}(x = 0) &= ika_{\text{Emitter}} - ikb_{\text{Emitter}} \end{aligned} \tag{17}$$

Note that $a_{\text{Emitter}}$ is not equal to 1 because the wavefunctions are normalized at the collector instead of the emitter, for the time being. The numerical value of the first derivative is computed by applying a differentiation scheme [26] centered on the emitter boundary node. Having computed both the wavefunction and its derivative, the system (17) can be solved for the two coefficients. Finally, all wavefunction values calculated across the structure are divided by $a_{\text{Emitter}}$, so that they are correctly normalized.

According to Pang [26], the Numerov method can be considered to be of order $O(N_x^4)$. However, its present application limits convergence to the order $O(N_x^2)$, because of the evaluation of the wavefunction derivative at the emitter contact. Once the wavefunction packet incident at the emitter has been evaluated, normalized wavefunctions are similarly cast into the structure from the collector, in order to compute the second integral term in (2).

The numerical implementation of (1), (2) and (9) requires a certain care in order to avoid aliasing of the discrete FT. To be consistent with Frensley's scheme, the condition $\Delta y = 2\Delta x$ is imposed, where $\Delta x$ and $\Delta y$ are the $x$- and $y$-grid spacings respectively. In this way, the $x$-grid vertices can be reused for the $y$-grid. In addition, in (9), the $k$-grid is implemented symmetrically to $k = 0$, such that [8]:

$$k_i = \{-k_{\text{Max}} + (i + 1/2)\Delta k\}_{i=0..(N-1)} \tag{18}$$

where $\Delta k$ is the $k$-grid spacing. The $y$-grid is defined as:

$$y_i = \{-y_{\text{Max}} + i\Delta y\}_{i=0..(N-1)} \tag{19}$$

$k_{\text{Max}}$ and $y_{\text{Max}}$ are the half-ranges of the two grids, which have the same number of nodes $N = N_y = N_k$. $k_{\text{Max}}$ and $y_{\text{Max}}$ are related by:

$$y_{\text{Max}} = \pi/\Delta k = \pi N/(2k_{\text{Max}}) \tag{20}$$

Note that this last condition cannot be fully satisfied, because $y_{\text{Max}}$ is rounded to the nearest node on the $x$-grid. The error on $y_{\text{Max}}$ is, however, equal to $\Delta x/2$ at most, i.e., less than 1% for $N > 50$. Also note that, in order to fully mesh the $y$-grid, the $x$-grid must be extended by $y_{\text{Max}}/2$ beyond the emitter and collector contacts. The $k$-grid defined in (18) should be reused in (2) in order to calculate the wave vectors that are cast into the SE. However, we observed that this is not necessary: in [1], we determine that 500 vectors spaced linearly over a 8 eV wide range from the conduction band upwards are sufficient to compute the DMF and WF without significant aliasing. In a subsequent study, we determine that even lower values (250 vectors over a 1 eV range) can be used [27], [28].

## V. Comparison of the analytic and numerical solvers

Fig. 3 shows the WF computed on the emitter node for a constant potential profile. For such a profile, (2) and (9) resolve to:

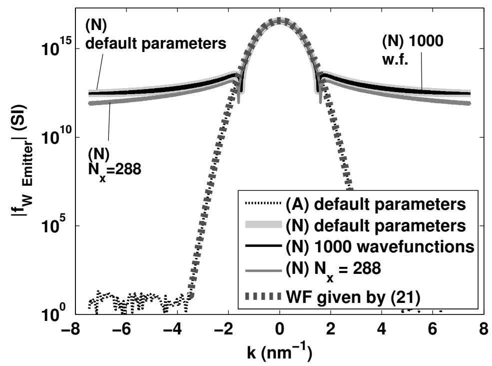



Fig. 3. Absolute WF vs. wave vector plots at $x = 0$, obtained by applying the analytic and numerical SE solver. Default simulation parameters: $m_r = 0.5$, 250 wavefunctions, $k_{\text{Max}} = 10$ nm$^{-1}$, $N_x = 144$, $N_k = 1000$.

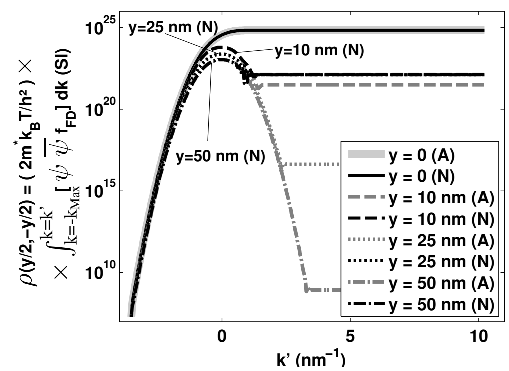
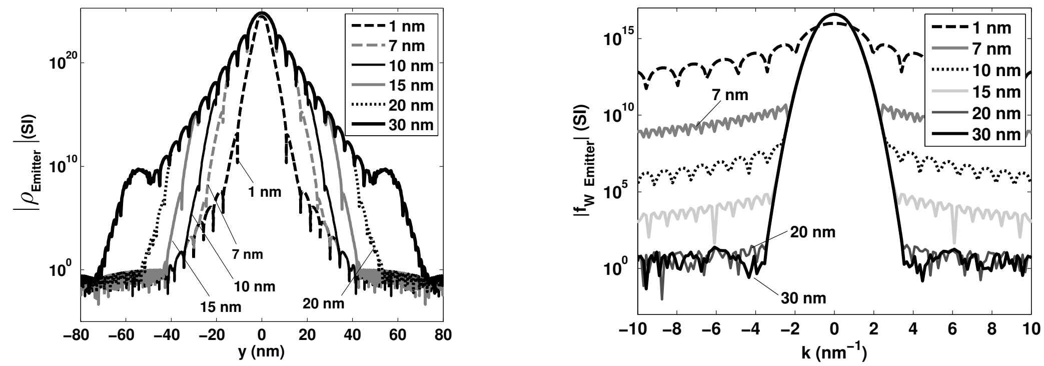



Fig. 4. $\rho(y/2, -y/2)$ integral computed between $-k_{\text{Max}}$ and $k'$ at $x = 0$ for different values of $y$. The letter A denotes the analytic SE solver, while N denotes the numerical one. Note that beyond $y = 10$ nm the value of the integral for the numerical solver remains constant, as the plots for $y = 10$, 25 and 50 nm coincide. Default simulation parameters: $m_r = 0.5$, 250 wavefunctions, $k_{\text{Max}} = 10$ nm$^{-1}$, $N_x = 144$, $N_k = 1000$.

$$\begin{gathered}\psi(x,k) = e^{ikx} \\ \rho\left(x+\tfrac{y}{2}, x-\tfrac{y}{2}\right) = \frac{2m^* k_B T}{h^2}\int_{-\infty}^{\infty} e^{iky} f_{\text{FD}}(E(k))\,dk \\ f_W(x,k) = \frac{m^* k_B T}{\pi\hbar^2} f_{\text{FD}}(E(k))\end{gathered} \tag{21}$$

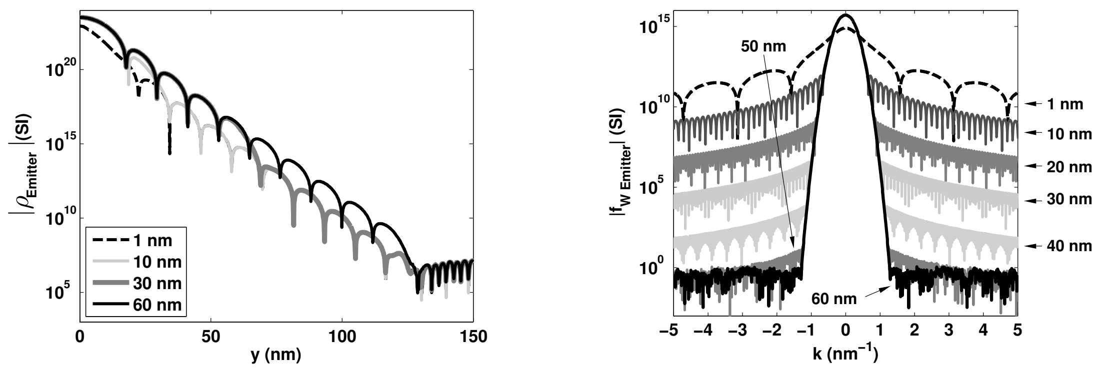
The first two plots in the figure are generated by casting 250 wavefunctions. A 1000-point $k$-grid is then used to calculate the WF. The first plot is obtained from the analytic SE solver. Consistently with (21), the WF is proportional to a FDD over a range of 15 decades, i.e., within machine precision of the IEEE 754 double data types used in the computations. This result confirms that FT aliasing is negligible, even though the number of wavefunctions is only half the $k$-grid resolution.

Three plots in the figure show a much smaller drop: these are obtained from the numerical SE solver. Because aliasing is negligible, the lobe-like artifacts observed in these plots can only be caused by numerical error. The figure shows that doubling the number of wavefunctions or the $x$-grid resolution does not significantly reduce this error. Note however that the lobes have only a minor impact on the charge, as they




(a) (b)

Fig. 5. Absolute-value DMF (a) and WF (b) computed at the emitter contact of a silicon-based single-barrier structure with a varying contact length. For a contact length between 20 and 30 nm, the WF is equal within machine precision to that obtained with a constant potential profile.




(a) (b)

Fig. 6. Absolute-value DMF (a) and WF (b) computed at the emitter contact of a III-V-based single-barrier structure with a varying contact length. For a contact length between 50 and 60 nm, the WF is equal within machine precision to that obtained with a constant potential profile.

begin to separate from the central FDD peak about 4 decades below the WF maximum. The charges given by the analytic and numerical solvers are thus consistent within 1%.

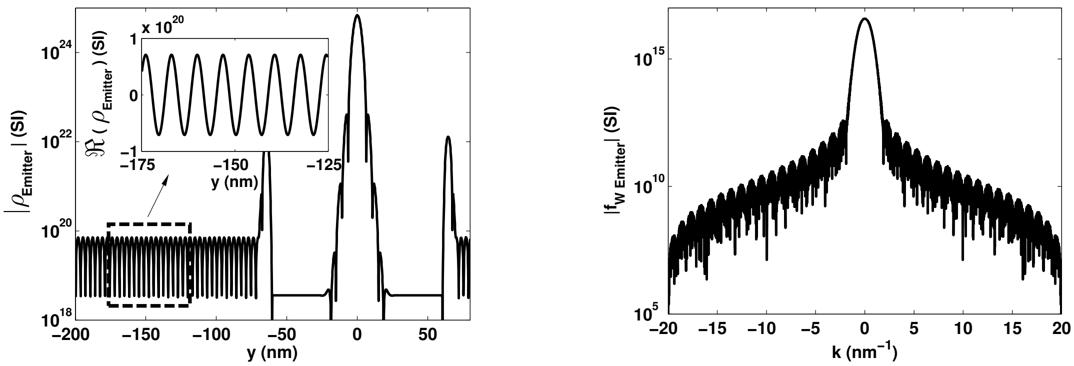
Fig. 4 provides some insight into the nature of the numerical error, by highlighting the computation of the DMF $\rho(y/2, -y/2)$ at $x = 0$. The different plots show how the value of the partially-computed $\rho$ integral varies as the upper bound $k'$ increases up to its maximum value $k_{\text{Max}}$. The plots obtained from the numerical and analytic solvers are compared. For $y > 10$ nm, the value of the integral peaks at $k' = 0$ and then drops as $k'$ increases. For $y = 50$ nm, the drop for the analytic solver spans about 15 decades. By looking at this specific plot, one realizes that the integral is subject to a variation of 15 orders of magnitude as it is computed. This means that, in order for the DMF to be calculated accurately, the integrand, i.e., the wavefunctions, must be evaluated to a relative accuracy of the same order.

The analytic solver is shown to be capable of this level of accuracy, as the three plots obtained for $y > 10$ nm drop to clearly distinct values. On the contrary, the numerical solver is not, because the same three plots are indistinguishable once they drop by only two decades below their peak. In the case of $y = 50$ nm, the accuracy of the numerical solver should be improved by more than 10 orders of magnitude to match the analytic one. Because of the insufficient accuracy of the numerical solver, all simulations presented in the following sections are based on the analytic one.

## VI. Study of the minimum contact length and of the wave vector grid range in unbiased structures

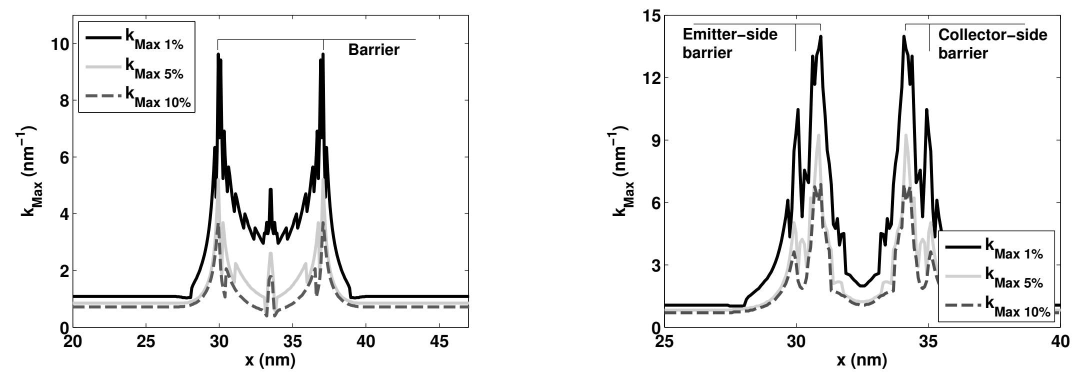
The first part of this section presents a study of the length of the emitter and collector contact regions. Its purpose is to determine the minimum contact length where the WF at the boundaries is equal to the equilibrium FDD within machine precision. The second part discusses the range of the wave vector grid that has to be applied to the WF in order to accurately evaluate the charge densities at all points in a given structure.

Fig. 5 shows the DMF and the WF computed at the emitter contact of a silicon-based single-barrier structure with varying contact lengths. The plots show that a minimum length between 20 and 30 nm should be used. Fig. 6 shows similar plots on a III-V based structure. In this case, the WF at the boundary converges to a FDD profile for a contact length of




(a) (b)

Fig. 7. Absolute-value DMF (a) and WF (b) computed at the emitter contact of a silicon-based double-barrier structure with a contact length of 30 nm.




(a) (b)

Fig. 8. Minimum wave vector grid half-range, as defined in (18), required to compute the charge within a threshold error of 1, 5 and 10% in silicon-based single- (a) and double-barrier (b) structures. The contact length is 30 nm. As $k_{\mathrm{Max}}$ is almost constant in the contact regions, these are left out of the figures.

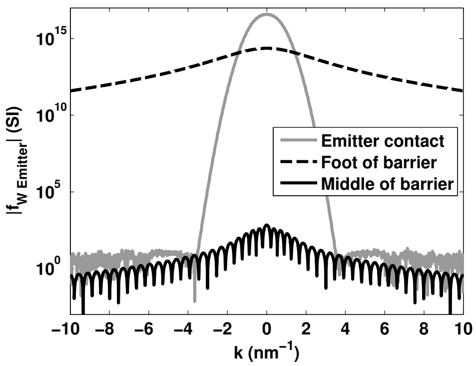
about 60 nm. As we explain in [27], this is due to the smaller relative mass in the III-V materials.

As for double barriers, Fig. 7 shows the DMF and WF computed at the emitter boundary of a silicon device. While in a single-barrier structure the plot drops to the level of numerical noise within less than 100 nm from the origin of the $y$-axis, in the case of a double-barrier it keeps oscillating with an amplitude that remains about constant over several hundred nanometers. The WF plot is also quite different from its single-barrier equivalent, as oscillating lobe-like artifacts separate from the central FDD peak about 4 decades below its maximum. This behavior is unaffected by the contact length. In [1], we hypothesized that this behavior was caused by an inaccurate computation of the wavefunctions. However, further analysis in [27] shows that the oscillations in the DMF are due to the very sharp peaks in the transmission spectrum that occur when the well resonates. While the amplitude of the oscillations is expected to drop eventually as $y$ increases, it does so very slowly: indeed, the $y$-grid range would have to be extended by orders of magnitude in order for the oscillations to fall below the level of numerical noise, which is not feasible due to computational resource constraints.

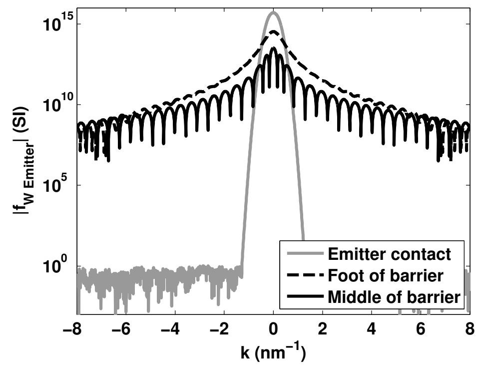
Aside from the contact length, there is another parameter that has a significant effect on the accuracy of a simulation, namely, the range of the wave vector grid that is applied to the WTE solver. Indeed, because the charge is computed by integrating the WF over the wave vector space, a too-narrow range can cause it to be underestimated. Here, we evaluate the minimum ranges that have to be used in order to compute the charge within error margins of 1, 5 and 10%. The reference charge used in the error computation is evaluated by applying the very large range $k_{\mathrm{Max}} = 20$ nm$^{-1}$. Fig. 8 shows how the wave vector range varies across a silicon-based structure for each error threshold. Note that, even if the double-barrier WF is affected by lobes as shown in Fig. 7, these cause an error in the integrated charge by less than 1% and are thus not expected to significantly affect the minimum range plots, at least for the 5 and 10% error thresholds.

For a single barrier, the minimum range plots show distinct spikes at the barrier end points. In fact, Fig. 9 shows that the WF at the foot of the barrier has a much gentler slope than in the contact region, thus requiring a wider wave vector range to accurately evaluate the charge. Also note that the WF in the middle of the barrier is much more oscillatory, which means that a finer mesh has to be applied. A similar behavior is observed in the double-barrier structure, as the plots reach their maximum values at the two ends of each barrier.

Finally, Fig. 10 shows a similar behavior in a III-V based




(a)




(b)

Fig. 9. Absolute-value WF computed at the emitter contact, as well as at the foot and in the middle point of the barrier in silicon- (a) and III-V-based (b) single-barrier structures. The lengths of the contact regions are 30 and 60 nm respectively.

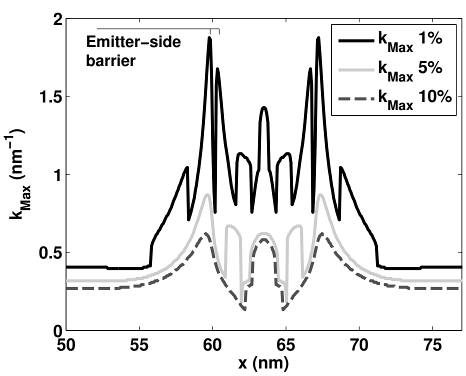
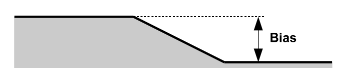
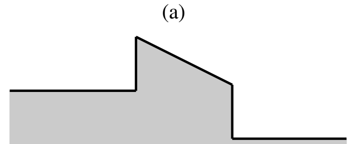


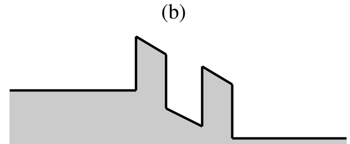

Fig. 10. Minimum wave vector grid half-range required to compute the charge within a threshold error of 1, 5 and 10% in a III-V-based double-barrier structure. The contact length is 60 nm.




(a)



(b)



(c)

Fig. 11. Schematic view of the potential profiles with a bias applied for a structure with no barrier (a) and one with a single (b) and a double (c) barrier.

double-barrier device, but with lower peaks. This is consistent with (3) and (4), where the lower effective mass in III-V devices results in narrower FDD and WF values at the contacts, as seen in Fig. 8.

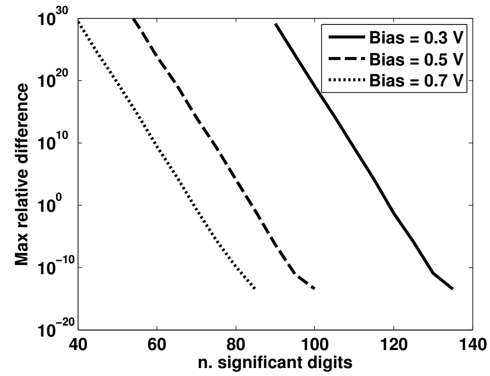
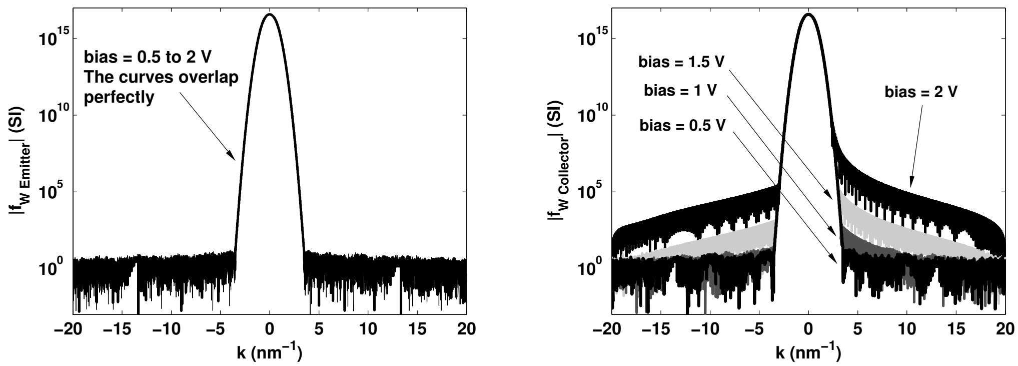



Fig. 12. Relative error in the wavefunction module for a single-barrier biased silicon structure, as a function of the number of significant digits of numerical precision used in the computation and at different bias values. The values plotted indicate the maximum error measured on 250 wavefunctions distributed over a range of 1 eV. The error is relative to reference wavefunctions calculated with a numerical precision of 200 significant digits: the reference wavefunction values and those at a lower precision are both converted to standard double precision, then the maximum relative error between their absolute values is evaluated.

## VII. Computation of the WF with a Schrödinger solver in biased structures

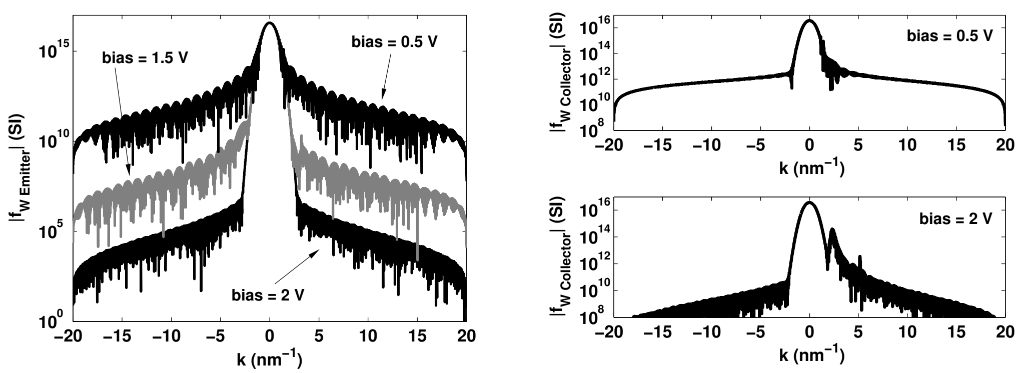
In this section, we look at the WF at the emitter and collector boundaries of structures where a bias is applied. Our goal is to observe whether lobes appear, and to make a rough estimate of their height. Three configurations are studied: the devices either have no barrier, or one, or two, as shown in Fig. 11. Each structure is composed of 30 nm long contacts that enclose a middle region which has a thickness of 7 nm in the no-barrier and single-barrier configurations and of 5 nm in the double-barrier one. In the single-barrier device, the middle region contains the barrier; in the double-barrier one, it contains both the barriers, which are 1 nm thick, and the well, which is 3 nm thick. The applied material parameters are representative of a silicon-based structure.

The potential profile is constant at the contacts and falls linearly in the middle region. This simplification makes it possible to solve the SE symbolically using Airy functions.




(a) (b)

Fig. 13. WF at the emitter (a) and collector (b) of a single-barrier silicon structure, at different bias points.



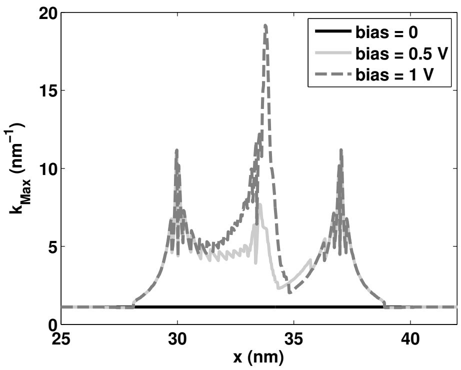

(a) (b)

Fig. 14. WF at the emitter (a) and collector (b) of a double-barrier silicon structure, at different bias points.

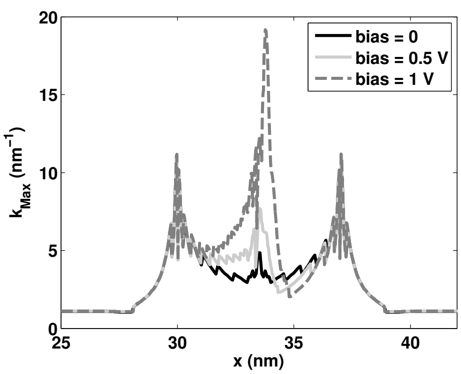
The symbolic wavefunctions formulae are very complex and are not optimized to reduce the numerical error. When evaluating them, it is therefore usually necessary to use a higher computing precision than the IEEE 754 standard machine double one. The MPMATH arbitrary-precision Python library is used [29]. Fig. 12 displays the level of numerical precision needed to calculate wavefunctions in single-barrier structures. It plots the numerical error in the wavefunction module at different working precisions, relative to a reference of 200 significant digits. The wavefunctions are computed across the structures at different bias points, by casting 250 incident wave vectors within a 1 eV energy range. The modules of the wavefunctions are then computed, and they are finally converted to double machine precision. The plots in the figure trace the maximum relative difference compared to the 200-digit reference. The plots all end as the relative difference reaches a value of about $10^{-15}$: this is due to the conversion to double precision, which does not allow to measure relative errors smaller than about 15 decades. The end point of each plot indicates the minimum working precision that is needed to compute the wavefunctions to double data type accuracy.

It can be seen that the required minimum precision is much greater than for unbiased structures, where standard double precision suffices. For double-barrier structures, the numerical precision required in the computation is similar to that for a single barriers; on the contrary, for no-barrier structures, double precision is once again sufficient. It should be stressed that optimizing the symbolic wavefunction formulae to reduce numerical error propagation might be beneficial in lowering the minimum required working precision; nevertheless, standard machine precision may still not be enough.

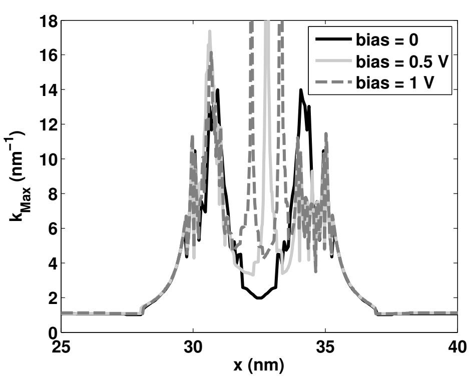
Fig. 13 shows the WF computed at the emitter and collector boundaries on a biased silicon single-barrier structure. The plots are virtually indistinguishable from those obtained on a device with no barrier. The emitter WF looks very similar to the symmetric FDD profile obtained on an unbiased device. In fact, the left side ($k < 0$) of the curve bulges a little less than the right one; this effect is however difficult to spot visually and can only be seen by looking at the numerical values of the WF. An asymmetric WF curve is expected, as it indicates a current flow between the emitter and the collector. In this structure, however, the high and thick barrier insulates well the two contacts. This occurs even at high bias points, as the different curves overlap almost perfectly. On the other hand, on a no-barrier structure, wavefunctions that are incident at the emitter can propagate freely towards the collector, once again independently of the bias point. On the collector side, the asymmetry is more evident, yet it still occurs many orders




(a)




(b)




(c)

Fig. 15. Minimum wave vector grid half-range required to compute the charge within a threshold error of 1% in silicon-based no- (a), single- (b) and double-barrier (c) structure.

of magnitude below the peak.

Fig. 14 plots the WF at the emitter and collector boundaries of a double-barrier structure. On the emitter side, one notices that the asymmetry between the left and right lobes is more evident, especially at the 1.5 V bias point. On the collector side, the asymmetry of the WF plot is again very evident, and lobes again appear about 5 decades below the peak. The plots in III-V-based devices are similar.

Fig. 15 plots the minimum wave vector grid range required to compute the charge within an error margin of 1% at bias voltages of 0, 0.5 and 1 V in the three structure types. As a bias is applied, the plots loose their symmetry. Similarly to the unbiased structures, they peak at the two ends of the middle

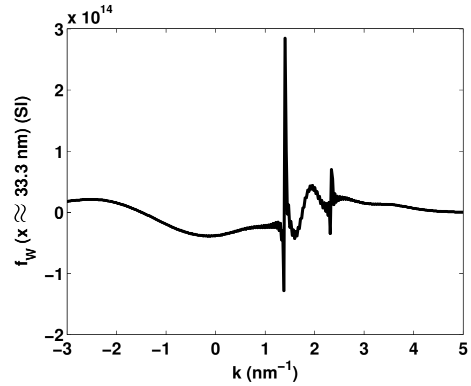



Fig. 16. WF in a double-barrier silicon structure with a 1 V bias at the $x$-node where the minimum wave vector range required to compute the charge within a threshold error of 1% is highest and close to 60 nm$^{-1}$.

region; the left peak (high-bias contact) is however higher in general than the right one (low-bias contact). The difference in height is especially marked in the case of the double-barrier structure, although it is no greater than 30%. In both single- and double-barrier structures, the difference in the heights of the peaks with and without an applied bias is also quite small.

The biggest difference compared to the no-bias plots occurs next to the middle point of the structure, where a very high peak appears. This peak varies considerably with the applied bias and is especially high on the double-barrier structure, where it goes up to 60 nm$^{-1}$ at a bias of 1 V. In fact, in the case of the double-barrier device, two distinct peaks appear near the middle of the structure, with the right one being higher. Fig. 16 shows the WF at the point where the peak is highest: while the plot is markedly asymmetric, it oscillates about the origin, with the negative and positive areas being very similar in size and canceling each other out when the charge integral is evaluated. Similar trends are observed when applying III-V material parameters.

As the WF is computed from the wavefunctions, scattering effects are not accounted for. Normally, interactions have the effect dissipating the charge carriers' energy, thus screening the electric field [30]. It is thus possible that the high-energy peaks observed in Fig. 15 may be considerably lower if interactions are simulated, as the distribution profiles would be pushed back toward the origin of the wave vector space.

## VIII. Conclusion

In this work, we have studied the Wigner Function (WF) by computing it directly from the Schrödinger Equation (SE), rather than by solving the Wigner Transport Equation (WTE). We have shown that, in order to accurately compute the WF over a large wave vector range, an extremely high machine precision is required, often higher than the IEEE 754 double data type. In fact, the numerical solver which we implemented, despite being able to computing charge densities well within 1% accuracy, generated lobes on the WF curves that could not be eliminated by applying denser grids.

This difficulty in computing the WF accurately comes from the density matrix integration step, followed by the application of the Fourier transform. Because these two operations are an inherent part of the WTE, the problems encountered in their implementation might explain those met in solving the WTE itself, namely the negative charge densities and the low charge drop-offs in presence of large barriers. These points have to be investigated further, and the SE solver may be of help, as it allows to compute accurate boundary conditions in the quantum region, at least for a single rectangular barrier structure. These boundary conditions can then be applied to the WTE solver, thus eliminating the memory overhead of meshing large contact regions.

This approach has also helped investigate some basic geometry parameters and numerical implementation conditions that must be applied to the WTE solver in order to accurately simulate 1D single- and double-barrier structures. As for the device geometry, the minimum contact length has been investigated. In single-barrier structures, it has been found that the WF at the boundaries follows a Fermi-Dirac Distribution (FDD) if the contact length is greater than 30 nm in silicon-based devices and about 60 nm in III-V-based ones; in double-barrier ones, the WF separates from the FDD reference profile a few decades below the peak and forms oscillating lobes that are not significantly affected by either the contact length or the numerical computing precision. As for the simulation numerical parameters, this work investigates the range of the WF in the wave vector space required to accurately compute the charge. In silicon-based structures, this range is estimated between 10 and 15 $nm^{-1}$ for an error smaller than 1%, and five to ten times lower in III-V equivalent structures.

This work also presents WF plots in silicon biased structures. The potential profiles used are simplified in order to allow for a symbolic solution of the SE, and carrier interactions are not taken into account. Nevertheless, it is still possible to observe some trends at different bias points, namely the increasing asymmetry in the lobes for positive and negative wave vectors in the WF plots at the collector of the different structures.

## REFERENCES

[1] A. Savio and A. Poncet, "Study of the Wigner function computed by solving the Schrödinger equation," in *Proc. 4th International Conference on Quantum, Nano and Micro Technologies (ICQNM 10)*, 2010, pp. 59–64.

[2] *Process Integration, Devices, & Structures (PIDS)*, International Technology Roadmap for Semiconductors, 2009. [Online]. Available: http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009Tables_FOCUS_C_ITRS.xls

[3] *Sentaurus Device User Guide*, Synopsys, Inc., Mountain View, CA, Sep. 2008, Version A-2008.09.

[4] E. Wigner, "On the quantum correction for thermodynamic equilibrium," *Phys. Rev.*, vol. 40, no. 5, pp. 749–759, 1932.

[5] N. Kluksdahl, W. Pötz, U. Ravaioli, and D. K. Ferry, "Wigner function study of a double quantum barrier resonant tunnelling diode," *Superlattices and Microstructures*, vol. 3, no. 1, pp. 41–45, 1987.

[6] N. Kluksdahl, A. M. Kriman, D. K. Ferry, and C. Ringhofer, "Self-consistent study of the resonant-tunneling diode," *Phys. Rev. B*, vol. 39, no. 11, pp. 7720–7735, 1989.

[7] W. R. Frensley, "Transient response of a tunneling device obtained from the Wigner function," *Phys. Rev. Lett.*, vol. 57, no. 22, pp. 2853–2856, 1986.

[8] W. R. Frensley, "Wigner-function model of a resonant-tunneling semiconductor device," *Phys. Rev. B*, vol. 36, no. 3, pp. 1570–1580, 1987.

[9] K. L. Jensen and F. A. Buot, "Numerical simulation of transient response and resonant-tunneling characteristics of double-barrier semiconductor structures as a function of experimental parameters," *J. Appl. Phys.*, vol. 65, no. 12, pp. 5248–5250, 1989.

[10] K. L. Jensen and F. A. Buot, "The effects of scattering on current-voltage characteristics, transient response, and particle trajectories in the numerical simulation of resonant tunneling diodes," *J. Appl. Phys.*, vol. 67, no. 12, pp. 7602–7607, 1990.

[11] K. L. Jensen and F. A. Buot, "Numerical simulation of intrinsic bistability and high-frequency current oscillations in resonant tunneling structures," *Phys. Rev. Lett.*, vol. 66, no. 8, pp. 1078–1081, 1991.

[12] F. Jensen, K.L. Buot, "The methodology of simulating particle trajectories through tunneling structures using a Wigner distribution approach," *IEEE Trans. Electron Devices*, vol. 38, no. 10, pp. 2337–2347, 1991.

[13] K. L. Jensen and A. K. Ganguly, "Numerical simulation of field emission and tunneling: A comparison of the Wigner function and transmission coefficient approaches," *J. Appl. Phys.*, vol. 73, no. 9, pp. 4409–4427, 1993.

[14] F. A. Buot and K. L. Jensen, "Lattice Weyl-Wigner formulation of exact many-body quantum-transport theory and applications to novel solid-state quantum-based devices," *Phys. Rev. B*, vol. 42, no. 15, pp. 9429–9457, 1990.

[15] H. Tsuchiya, M. Ogawa, and T. Miyoshi, "Simulation of quantum transport in quantum devices with spatially varying effective mass," *IEEE Trans. Electron Devices*, vol. 38, no. 6, pp. 1246–1252, 1991.

[16] K. K. Gullapalli and D. P. Neikirk, "Incorporating spatially varying effective-mass in the Wigner-Poisson model for AlAs/GaAs resonant-tunneling diodes," in *Proc. 3rd International Workshop on Computational Electronics*, Portland, OR, 1994, pp. 171–174.

[17] K. K. Gullapalli, D. R. Miller, and D. P. Neikirk, "Simulation of quantum transport in memory-switching double-barrier quantum-well diodes," *Phys. Rev. B*, vol. 49, no. 4, pp. 2622–2628, 1994.

[18] D. R. Miller and D. P. Neikirk, "Simulation of intervalley mixing in double-barrier diodes using the lattice Wigner function," *Appl. Phys. Lett.*, vol. 58, no. 24, pp. 2803–2805, 1991.

[19] G. Y. Wu and K.-P. Wu, "Electron transport in a resonant-tunneling diode under the effect of a transverse magnetic field: A quantum theory in the Wigner formalism," *J. Appl. Phys.*, vol. 71, no. 3, pp. 1259–1264, 1992.

[20] B. A. Biegel and J. D. Plummer, "Comparison of self-consistency iteration options for the Wigner function method of quantum device simulation," *Phys. Rev. B*, vol. 54, no. 11, pp. 8070–8082, 1996.

[21] H. Grubin and R. Buggeln, "RTD relaxation oscillations, the time dependent Wigner equation and phase noise," *J. Comput. Electron.*, vol. 1, no. 1, pp. 33–37, 2002.

[22] M. Nedjalkov, H. Kosina, R. Kosik, and S. Selberherr, "A Wigner equation with quantum electron-phonon interaction," *Microelectron. Eng.*, vol. 63, no. 1, pp. 199–203, 2002.

[23] Y. Yamada, H. Tsuchiya, and M. Ogawa, "Quantum transport simulation of silicon-nanowire transistors based on direct solution approach of the Wigner transport equation," *IEEE Trans. Electron Devices*, vol. 56, no. 7, pp. 1396–1401, 2009.

[24] J. Kefi-Ferhane and A. Poncet, "Deterministic simulation of transport in MOSFETs by computing Wigner, Poisson and Schrödinger equations," *Phys. Status Solidi A*, vol. 201, no. 11, pp. 2518–2521.

[25] A. Jüngel, *Transport equations for semiconductors*. Berlin: Springer, 2009.

[26] T. Pang, *An introduction to computational physics*, 2nd ed. New York: Cambridge University Press, 2006.

[27] A. Savio and A. Poncet, "Study of the Wigner function at the device boundaries in 1D single- and double-barrier structures," *J. Appl. Phys.*, accepted for publication.

[28] A. Savio and A. Poncet, "Study of the Wigner function computed by solving the Schrödinger equation," in *15th International Conference on Simulation of Semiconductor Processes and Devices (SISPAD 10)*, 2010.

[29] F. Johansson. (2010) MPMATH Python library for arbitrary-precision floating-point arithmetic. [Online]. Available: http://code.google.com/p/mpmath/

[30] W. R. Frensley, "Boundary conditions for open quantum systems driven far from equilibrium," *Rev. Mod. Phys.*, vol. 62, no. 3, pp. 745–791, 1990.

# Assessment of Simulator Fidelity and Validity in Simulator and On-the-road Studies

Andreas Riener
*Institute for Pervasive Computing, Johannes Kepler University Linz, Austria*
*Tel. +43/(0)732/2468–1432, Fax. +43/(0)732/2468–8426*
*Email: riener@pervasive.jku.at*

***Abstract*—A lot of research groups all over the world have tried to relate results from driving simulator studies to real driving behavior. A solution, e. g. in form of a conversion table, would be of great value. Unfortunately, status quo is that even with expensive, high fidelity simulators the validity of results cannot be guaranteed. One reason for this is that a person's behavior cannot be described by mathematical rules and depends, beside the task of interaction, on several subsidiary influence factors. Starting with an elaborate review of driving validity and fidelity constraints, the aim of this paper is to summarize on our research responding to the question to what extent driving simulators can be used to serve as cheap and easy realizable environments for simulating on-the-road behavior. The purpose of the studies was to determine (i) whether or not it is in general possible to approximate real driving with simulator studies, (ii) situation and modality dependent correction or scale factors to deduce real reaction times from simulation, and (iii) further requirements, parameters, and restrictions to be satisfied for succeeding high fidelity studies. Two user studies were conducted, a low fidelity trace-driven simulation in a lab environment and a on the road driving experiment. Recorded reaction times were compared in order to assess the validity of data generated in these experimental series. The events were, in the case of simulation, triggered trace-driven or, in the real driving experiment, manually activated by the experimenter and notifications were forwarded to the driver using a random assignment of one of the modalities vision, hearing, or touch. Results indicate that drivers responds faster to steering requests in the driving simulator compared to real driving. The explanation for this difference can most likely be derived from the fact that test persons were less demanded in the first (artificial) compared to the second (real) setting. When analyzing data on individual notification channel basis, it can be observed that (i) the order of channels with respect to average response times is the same in both settings (vibro-tactile, visual, auditory) and (ii) the reaction time differences are almost uniformly distributed. Prospective work in comparative studies is projected to happen in two directions, on improvements in the behavior of the low fidelity simulator to become as close to reality as possible and in the utilization of high fidelity driving simulators to directly relate real driving results to.***

***Keywords*—Simulator validity/fidelity, Real-driving studies, Trace-driven simulation, Driver-vehicle interaction (DVI), Feedback modalities, Performance evaluation.**

## I. MOTIVATION: SIMULATION IN THE CAR DOMAIN

Simulation or driving simulator studies have become, for several reasons, state-of-the-art methods in the development process of vehicles. One factor for this is *economically justified* – research and development expenses for a new generation of vehicles is just about 5% on the overall costs of car production; however, car manufacturers are increasingly requested to operate as efficient in terms of cost as possible – for the area of development and design this would only be possible when applying computer assisted simulation techniques to all stages of development. Particularly in user interface design, the strong interrelationship between the driver, his/her personal preferences and the different control and assistance systems in a vehicle poses problems to be considered, and necessitates, in excess of pure simulation, a more detailed treatment by application of user studies and/or driving experiments.

Another factor is *time driven* – the car domain is today requested to shorter and shorter time-to-market cycles. The design life cycle for the automotive market has continuously decreased from over 4 years, approaching now 15 to 18 months – a delay of only one or two months can cost a manufacturer up to 30% of market loss [2]. This shorter cycles are caused by $(i)$ driver assistance systems and control instruments catching on more and more into the dashboard, $(ii)$ new forms of driver-vehicle interfaces established from one vehicle generation to the next, and $(iii)$ trade rivalry by car manufacturers to outsell competitors which forces developers to make their own cars more attractive with regard to functionality, comfort, gadgets, etc., to a particular target market.

The third is *technology driven* – beside the before mentioned production-centric issues other challenges are expected to arise in the near future. Damiani *et al.* [p. 95][3] stated in the context of a discussion on attributes of future cars "*[..] Maybe new, unexpected needs and fashions will arise, but in any case the design and development of new technologies and devices will have to face the challenges opened by the new paradigms*". Overlooking the current research in the field of driver-vehicle interaction shows that it is moving toward enabling a full interconnection between drivers, vehicles, and infrastructure in order to increase driving safety and efficiency [4], [5], and in a next step to vehicles moving fully autonomic ("*[..] It is expected that men and women of the future when moving will continue their normal life, leisure and work while the car will take care of their safety*") [3].

With respect to these three pillars a performance and/or usability evaluation of user interfaces for (future generations of) vehicles is already today (and will be so even more in the future) infeasible in on-the-road experiments. This is to a lesser extent due to pure economical reasons, but most likely caused by $(i)$ safety problems, may affecting test persons and other road participants, $(ii)$ confirmed too long development, preparation and execution times to adhere to the strict production plans, and $(iii)$ unknown real-time behavior of new interaction/communication concepts. Therefore, we (and many other research groups all over the world) recommend to use driving simulators to fulfill all of the specified requirements and basic conditions. In this way manufacturer should achieve both a successful launch of new vehicle models as well as a optimum, "relaxed" use while in operation.

*Simulation in Automotive Applications*

Our main motivation followed in this work stems from the third issue, *technological challenges*, as we are, for instance in the project SOCIONICAL [6], interested in the development of socio-technical systems on large scale. One of the investigated domains is the field of transportation, where we are amongst other things dealing with new concepts for driver-vehicle-infrastructure operation. (The task of driving can be described as a highly dynamic and local feedback loop between a driver and a car or its operational controls and assistance systems [7, p. 5]; the interaction between one person and the technology integrated into a single car can be denominated as the basic building block of a socio-technical system.) Within this field of research it would almost be impossible to conduct on-the-road studies evaluating the collective effect of hundreds or thousands of drivers operating their car in a certain region of interest.

Research branches currently under development and products already introduced into market confirm the need for simulation to avoid safety issues for both test persons and other road participants as well. The head-up display (HUD) available for years, for example in BMW's premium type cars [8], serves as one instance. Ward *et al.* [9] discovered that the head-up display requires both a higher mental effort and a higher cognitive load from the drivers while the results of Nakamura *et al.* [10] and others showed reduced workload, decreased response times, more consistent speed, and increased driving comfort. A second example legitimating simulation is the emerging interest in olfactory interaction. Such systems have only be used rarely in the car to date [11], [12, p. 30]; however, it would be feasible to display a scent of burning oil in the passenger compartment to warn the driver or even to systematically employ odors to calm down the driver or strengthen his/her energy to increase driving safety – as it has been evidenced that the odor of jasmine or lavender elicits sedative or relaxant effects [13]. Nevertheless, the application of the same is questionable – on the one side as it still remains, most likely due to its subtle and imprecise perception, a developing research branch [14], [15] and at the other side, as particular fragrances won't "work" for everyone [11]. Aside from this, it is known that the emotional state of healthy subjects has a clear effect on olfaction [16, p. 287]. As one example, a negative emotional state would reduce olfactory sensitivity (emotional states are likely to change quickly and uncontrolled during vehicle operation, e. g., in congested situations or on vehicles cutting in).

*Research Hypotheses*

The main reason arguing for simulation in the car domain is, that it can be applied in many critical areas and allow issues to be addressed before they actually become problems. Therefore, it is not "just a technology" because it forces one to consider global terms of system behavior, most frequently represented by complex models behaving in more than the sum of their components [17, p. 1].

Within this work we have engaged in monitoring, recording, and interpreting dynamic driving activities applied to a trace-driven simulation approach, and afterwards repeated in a real driving experiment. The focus of our research attempt was the investigation and classification of disparities using different feedback channels in the two settings. Moreover, we wanted to give answers to the question to what extent driving simulators can be used to serve as cheap and easy realizable environment for simulating on-the-road behavior, thus facilitating later user experiments to be laboratory based only. The declared aim of this work was to provide a metric for response time differences between simulation and the real world to be used as a conversion table to replace future on-the-road studies with simulation experiments. This suggestion can be assumed feasible as it has been shown for the automotive domain that simulation is already today a useful approach for data collection and driver behavior analysis [18], [19].

In particular, we hypothesize that

*(i)* Reaction times detected in real driving tests as well as in simulation studies are in the same order of magnitude for notifications with a certain sensory channel, so that a situation dependent correction factor can be tabulated first (separately for each feedback modality), and later used for deducing real reaction times from simulation (*relative validity*).

*(ii)* The different notification modalities available in the car (only the three main sensory channels vision, hearing, and touch were analyzed) behave, with regard to a single driver-vehicle interaction cycle, similar in their position/rank in on-the-road studies and simulator experiments.

As in detail elaborated below, the quality of a simulator is normally defined by its *validity* and *fidelity* – the stated research hypotheses are in line with these two qualitative simulator characteristics.

*Outline*

The rest of this paper is organized as follows: Section II gives an overview of the state-of-the-art of driving simulators, required for the development of the simulation environment used within our studies and furthermore summarizes predeterminations and regulations for both the simulated and the real driving experiments. Section III examines several aspects of user interface evaluation in the car domain while Section IV gives insight into both the driving simulator and the on-the-road experiments, their execution, and details on the logging of user data. Section V presents and discusses the findings of the comparison between these two experimental series with regard to qualitative aspects. The concluding Section VI summarizes the paper and gives suggestions for future work to improve the quality of the driving simulator in order to successfully replace real driving studies later.

## II. Driving Simulators: Review of the State-of-the-Art

Driving simulators have a long history in the automotive domain, providing a means of precise design and control of scenarios, allowing for instrumentation and logging, and ensuring repeatability, e. g., for re-running situations with anomalous behavior. Simulators have been successfully applied for the evaluation of driving safety ([20], [21]) and driver behavior, such as influence on fatigue ([22], [23], [24]) or drunk driving ([25], [26]), education in driving schools ([27], [28], [29], [30]), for the design of new functions of vehicles ([31], [32], [33], [34]), and in particular for user interface evaluation ([35], [36], [37], [38], [39], [40], [41], [42], [43], [44]). To directly address the issue of ever decreasing production cycles, simulation has also been successfully applied for a long time to crash ([45], [46], [47]) and wind tunnel tests ([48], [49], [50], [51], [52]).

One of the greater problems in simulation is, that driver behavior (cognitive resources that drivers devote to both steering tasks and side activities) in simulated driving experiments may differ significantly compared to those deployed in real studies [20, p. 590], [53, p. 89] – most likely due to the fact of $(i)$ complex person behavior representation ([54], [55], [56]), $(ii)$ a discoverable lower focus on the tasks due to the risk free environment [1, p. 30], [53], and $(iii)$ vehicle movement dynamics ([57], [58]) or motion cuing ([59], [60], [61]).

### A. *Qualitative Simulator Development*

In the simulation domain the two aspects $(i)$ fidelity and $(ii)$ validity are used for characterizing the quality of simulators. Engströem *et al.* [62] indicated that there is a discoverable relationship between these two metrics as high fidelity simulators offer, per definition, a more realistic driving environment and, thus, a higher validity of obtained results as compared to low fidelity driving simulators with its accredited lower validity of data. For the implementation of a concrete simulator a trade-off between fidelity (the better the higher the development and running costs) and validity (greater validity is attributed to provide results more close to reality; however, some effects observed in low fidelity simulators may be are not appearing in high fidelity simulators) has to be set based on trade-off between the requirements of the studies to be conducted later and the budget available for implementing/constructing the simulator.

*(A) Fidelity:* Fidelity is often equated to the level of realism represented in the simulation [53, p. 89] (it has been reported that the closer a simulator approximates the real world in terms of design and layouts of control, realism of the shown scene, etc. the greater is the fidelity of this simulator). For a better classification of simulators, Rehmann [63] has proposed to use four interrelated fidelity dimensions.

As stated above, high fidelity often goes along with high driving simulator costs; particularly for human factors research the costs of driving simulators are often very high, prohibiting their application so that automotive human factors research related to the evaluation of interface design is usually to date still done in on-the-road experiments [64].

*Consequences (for this work):* With respect to the designated fidelity and the high costs expected for simulators required for human factors research, we designed the driving simulator as used in the studies explained below in a opposite direction, achieving relatively high fidelity at low cost: High *equipment fidelity* was achieved by using a real car for our simulator tests (the same as later used in the real driving experiments), *environmental fidelity*, such as motion cues and sensory information obtained from the real world environment, was, however, fulfilled to a lesser extent (motion cues were not percepted at all as the car was parked (jacked up) in a garage, sensory information from the outer world was either not available or not related to the scene shown in the simulator). *Objective fidelity* is said to be related to dynamic cue timing and synchronization issues, e. g., between steering input and corresponding visual cues. This point was almost fulfilled as the shown driving scene, the notification patterns delivered to the driver (visual, auditory, vibro-tactile), and the vehicle controls used by the driver to react on perceived stimuli were all connected and synchronized one with the other on a single computer (a correct timely behavior was one of the most important design requirements). The last dimension of Rehmann's four-staged classification proposal, *perceptual fidelity*, is concerned with $(i)$ the degree to which the driver perceives the simulation to be a reproduction of the real driving task and $(ii)$ the degree to which a driver's interaction with the driving environment corresponds to real-world driving [63]. Both sub-points are fulfilled to a medium to high degree as the scene used in the simulator was a prerecorded video trace of a real journey through the city which was synchronized to stimulations (vehicle feedback). The corresponding feedback was collected from the user via the real control instruments in the car (i. e., turn indicators, light switch).

*(B) Validity:* The second qualitative aspect to describe a simulator is its validity [65]. It typically refers to the degree to which behavior in a simulator corresponds to behavior in real-world environments under the same conditions and maybe is, as stated before, affected by the level of fidelity. Validity is much harder to achieve as fidelity as it is dependent on many (human) factors like rewards and punishments for "appropriate" driving behavior [66], cognitive workload levels and psychological environment [67, p. 315], different levels of stress [68], etc. Furthermore, it has been shown that individuals can experience symptoms of sickness (like dizziness or headache) in driving simulators [66], making it difficult to relate results to driving behavior observed in reality.

According to Young *et al.* [53], the best method for determining the validity of a simulator is to compare driving performance in the simulator to driving performance in real vehicles under the same driving tasks. For validity, two types can be differentiated, $(i)$ absolute and $(ii)$ relative validity. The former is achieved if the numerical values obtained from driving simulator studies and on-the-road tests are identical (or near identical), the latter when driving tasks in the simulator and in real driving studies have a similar affect (e. g., a similar direction of change) [67]. The feasibility of this recommendation was shown by Lee [69], who confirmed $(i)$ the validity of driving simulator studies and it's on-the-road driving counterpart and $(ii)$ that it is a safer and more economical method than the on-road testing to assess the driving performance of (older adult) drivers.

*Consequences (for this work):* In order to determine the validity of our approach on simulated driving we have conducted a second experiment using an on-the-road driving study. For that the same (physical) car, the same notification patterns, and the same operational tasks to be completed by the test drivers were used. Beside these basic factors we selected a similar length of the route (with regard to driving time) for the real driving experiment and used days with low traffic to avoid distraction from other road participants as good as possible.

To come to the point, the results obtained in our studies when comparing the two experimental series are satisfying (a detailed investigation is given in Section IV). Although absolute validity of the simulation environment has not been demonstrated, relative validity has been achieved. In order to further strengthen the plausibility of a relationship between the two approaches, several improvement potentials have been identified during experimental conduction – these are summarized in the last section of this paper.

## III. Vehicle UI's: Evaluation using Simulators

Research on new automotive systems currently relies on car driving simulators, as they are a cheaper, faster, and safer alternative compared to tests on real tracks. However, there is increasing concern about the fidelity of results provided from the simulator and their influence on the validity of these studies in the reality. Especially for motion cuing, and here for high-speed curve driving, the provision of large sustained acceleration would be difficult to be reproduced in the limited motion space of simulators.

User interface (UI) composition and evaluation is a challenging task in the design phase of a new vehicle generation, particularly today where the time-to-market cycles decreases steadily, reaching 15 to 18 months after up to 48 months a few years ago [70]. This reduction in development time would only be possible when applying simulation to all stages of the vehicle manufacturing cycle, even to the design and evaluation of user interfaces [71]. Solutions like Virtual Product Development (VPD) and Computer Aided Engineering (CAE) techniques [70] are applicable to "hardware design", nevertheless these approaches are not suitable for the development of user interfaces without further considerations regarding the user as they are highly dependent on users preferences – and a person's behavior does simply not follow mathematical rules or physical laws.

However, (full-motion) flight simulators, which are situated between "model-based" simulation and tests in the real world, have been successfully showed its applicability for pilot training in the past decades (see, for instance, [72], [73], [74] or [75]). Following this approach, driving simulators should be established for user interface testing in the automotive domain as well. Like in the flight simulator, tests with vehicle simulators can be conducted in a quiet, controlled test environment with the following advantages: $(i)$ mistakes can be reviewed immediately, $(ii)$ a failed task can be repeated by rewinding and replaying the scenario, $(iii)$ the "driver" is secured from accidents and other road participants do not need to be endangered, and $(iv)$ user's concentration is on the task, not on the noisy, stressful environment [76].

It is supposed, and has already been shown for the evaluation of single driver-single vehicle interaction issues, that driving simulators can be successfully applied for user interface evaluation (at least for a certain level of validity) [77], [78], [79] or in the automotive electronics Journal [80].

Summarizing the considerations regarding simulation poses the problem of a still missing "reality effect", as in real situations aggregated, e. g., from motor noise, environmental sound (raindrops pattering on the front windshield, honking cars, etc.), road vibrations (providing implicit knowledge about driving behavior, such as drifting on gravel or snowy roads), penalties for driving violations, danger of road accidents, etc. – even if the replayed scenery in the simulator looks highly realistic and the experiments are processed in a "real" physical car (as done in the trace-driven simulation experiment described later in this work). For driving experiments using a simulator instead of real driving studies it

Figure 1. Experimental setting for the trace-driven experiment with vehicle control signals taped via Atmel $\mu$C and forwarded to the control application (leftmost image), screen shot of the prerecorded test run as shown the participants during the experiment (rightmost picture).

would be rather important to know these missing parameters or at least their joint impact on the distortion of results. If it is possible to provide evidence for a common or individual “correction factor(s)” or to build a “correction-model”, it would no longer be required to conduct expensive (in terms of cost and time) real-driving studies. Nevertheless, in any case, and according to findings obtained so far, it will be almost impossible to cover the full range of errors, and thus to fully replace on-the-road studies with simulation.

*Authenticity: Example of alcohol impaired driving*

Before replacing user studies with simulation experiments on a broader basis it has to be approved that the results of the latter are equal or comparable to that of the former, at least for the key issues. If authenticity for both test persons and achieved results can be confirmed it would be much easier (or even yet feasible) to perform specific tests by simulation, and to give expressive statements on the results for the reality. For instance, it is not recommended to study alcohol impaired driving or the influence of fatigue in on-the-road studies, however, an investigation by using a driving simulator would be possible at no risk and is assumed to provide meaningful results directly transferable “back to the real traffic”. To substantiate this implication, the research conducted by Robbe [81] can be quoted, performing one laboratory and three on-road driving studies (restricted highway, normal highway, urban traffic) concerning the effects of marijuana smoking on actual driving performance, and detecting almost the same results.

## IV. The Experiments

A driving simulator was developed initially, measuring response times from a driver on a limited number of vehicle control operations, notified either visually, auditory or vibro-tactile. The experiment was processed in a real car parked in a garage, with “real” vehicle controls such as turn indicator or light switch connected to a microcontroller (see Figures 1, 4). Instead of a computer-generated scenery, a prerecorded journey of about $30min.$ in length across the city of Linz was replayed on a large projection screen placed in front of the windscreen. The detailed setting of the experimental system as well as a in-depth description of the conducted experiments including evaluation and results is given in Riener *et al.* [31].

After this first experiment – which has already been defined considering a later reuse in a real driving study – a similar test series (using the same car, the same test participants (a subset of them), the same tasks, the same notifications) was prepared and performed in an on-the-road scenario as described below. This second series was conducted several month later so that dependencies between the two experiments like learning effects of test participants should not have been observed.

As the detailed results of both the simulation experiment and the on-the-road study are published ([31], [1]), we will subsequently focus more on the specialties of the two settings, accounting differences between the two approaches (wherever applicable), and conclude with a elaborate interpretation of achieved results – the last two Sections V and VI are dedicated to this objective. The two experiments will be there compared in detail and, more interesting, possible effects will be discussed.

*Geographic Similarity of Experiments*

Both the simulation experiment (actually the scene shown in the simulator) and the on-the-road driving tests were conducted in the greater area of Linz, Austria. For the former, a $21km$ (original driving time approximately $30min.$) long trip cross-town the city of Linz was recorded, containing controlled and uncontrolled crossings, road tunnels and freeway components (the video camera was mounted in the car in order to tape the view of the driver). Waiting times on crossings and unsubstantial or pointless driving sections were removed from the taped run (in a way indiscernible for the test driver), afterwards the remaining video was tagged with action/notification points. On the other side, the real driving study was conducted in and around the city of Perg, $25km$ east of Linz, mainly due to the lower volume of traffic and thus, a reduced risk of incidents. All the test runs were processed here on a predefined, fixed course (see Figure 2) with a circuit length of $25.79km$; an entire run lasts on average $34min$. It has to be noted that vehicle specific data from the controller area network (CAN) bus (via ElmScan5 USB ELM327 on board diagnostics (OBD) interface) and global positioning system (GPS) position as well as vehicle acceleration data (from RaceTechnology [82] DL2 data logger with integrated $20Hz$ high accuracy GPS tracker and IMU06 six degree of freedom inertial measurement unit) were gathered for a different purpose, and not for evaluation or data interpretation in the context of this work (to be correct, they were only be used for visual inspection; additional data correctness verification was not applied).

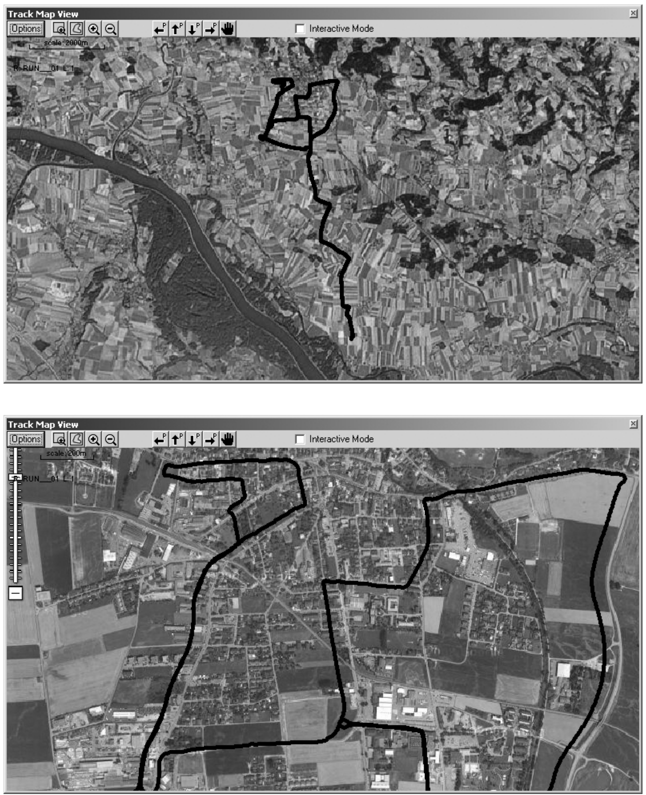



Figure 2. GPS traces of the predefined driving route (length of $25.79km$) with subjacent satellite maps (the upper image shows the overall journey (similar initial and final points), the lower displays a detailed view of the trip across the city of Perg).

### *Differences in Data Acquisition*

The main difference between the two settings is grounded in the initiation of notifications. In the experiment using simulation it is trace-driven and time aligned to the video of a prerecorded journey, in the real study it is executed manually by the experimenter according to predefined positions in the driven route (a person initiating the feedback was seated on the back seat behind the driver so that he/she – and the actual task of activation – could not be seen, neither be guessed by the driving person; once the activation key was pressed (=task notification) one sensory channel out of the three available modalities was chosen randomly by the software).

For both the trace-driven experiment and the on-the-road series exactly the same setting was used. Notifications about required driving activities were delivered to a particular driver using either a visible, audible or vibro-tactile feedback signal. Reaction times from the driving person were then collected from the real control instruments of a car (e. g., turn indicator, light switch) connected to a Atmel AVR ATMEGA 8 microcontroller and forwarded to the capturing software.

```
------------------------------------------------------------
Real-Driving Journey started at 30 March 2009 10:04:51
------------------------------------------------------------
Visual Task TurnRight 0 created. (10:05:01734)
Visual Task TurnRight 0 finished: 1,141 ms (10:05:02875)
Visual Task TurnLeft 0 created. (10:06:1393)
Visual Task TurnLeft 0 finished: 935 ms (10:06:02328)
Vibro-tactile Task TurnLeft 1 created. (10:07:42734)
Vibro-tactile Task TurnLeft 1 finished: 1,016 ms (10:07:43750)
Auditory Task LightsOn 0 created. (10:08:05234)
Auditory Task LightsOn 0 finished: 1,412 ms (10:08:06646)
.
.
.
Visual Task TurnRight 14 created. (10:37:01343)
Visual Task TurnRight 14 finished: 1,094 ms (10:37:02437)
Auditory Task TurnLeft 13 created. (10:38:15640)
Auditory Task TurnLeft 13 finished: 860 ms (10:38:16500)
------------------------------------------------------------
Real-Driving Journey stopped at 30 March 2009 10:38:35
Stopping unfinished tasks...
Total number of tasks: 31
------------------------------------------------------------
```

Figure 3. Abstract of the log file of one particular driving experiment. Time flags in parenthesis are used for synchronization with other vehicle-specific data on basis of GPS time.

*Logging/Recording:* Log files were compiled for each trip in the two experimental series for evaluation purpose. Figure 3 exemplarily shows an abstract of such a log file for a real driving trip employing the following list descriptors.

*(i) Selected notification channel:* In the actual studies one out of *visual*, *auditory* or *vibro-tactile* (unimodal feedback). The software was designed modularly expandable, so that this list could later be extended, for instance with notifications using the olfactory channel (*smell*) or by combining notifications from more than one sensory modality at a time to a multimodal feedback system.

*(ii) Kind of activity:* This is an indicator for the action to be performed by the test person. Within the here discussed experiments we used the activities $(i)$ turn right, $(ii)$ turn left, $(iii)$ lights on, and $(iv)$ lights off. Each activity is followed by an enumerator counting the actual number of occurrences of that activity (starting from zero).

*(iii) Indicated command:* This field is one of either "start" (*created*) where user notification is initiated, or "stop" (*finished*) where the response from the user was recognized.

*(iv) Response time:* This field is defined only for the *finished* command and indicates a driver's reaction

Table I
Statistics on reaction times using a 5% confidence interval (time values given in ms).

| Trait | Min $x_{min}$ | Max $x_{max}$ | Mean $\overline{x}$ | Median $\widetilde{x}$ | Std.Dev. $\sigma$ | Min $x_{min}$ | Max $x_{max}$ | Mean $\overline{x}$ | Median $\widetilde{x}$ | Std.Dev. $\sigma$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | Trace-driven simulation (CI 5%, 752 datasets) | | | | | On-the-road studies (CI 5%, 353 datasets) | | | | |
| Combined | 281.0 | 1,985.0 | 889.2 | 812.0 | 349.9 | 203.0 | 1,750.0 | 1,003.2 | 1,000.0 | 331.9 |
| Visual | 391.0 | 1,922.0 | 784.3 | 703.0 | 295.8 | 360.0 | 1,750.0 | 978.7 | 940.0 | 328.7 |
| Auditory | 641.0 | 1,984.0 | 1,129.6 | 1,078.0 | 269.6 | 203.0 | 1,719.0 | 1,179.5 | 1,195.0 | 298.6 |
| Vibro-tactile | 281.0 | 1,625.0 | 690.6 | 641.0 | 255.9 | 265.0 | 1,672.0 | 879.9 | 828.0 | 299.3 |

time from one particular stimulation activity. It can be calculated simply by using the formula

$$t_{response} = t_{finished} - t_{created} \tag{1}$$

*(v) Actual time:* This field contains Internet synchronized time (UTC) to allow data alignment between measures recorded with the local acquisition system connected to the notebook computer with vehicle specific measurements and GPS data recorded with the RaceTechnology [82] data logging equipment.

*User Studies*

Two user studies were carried out to prove the research hypotheses $(i)$ reaction times are in the same order of magnitude and $(ii)$ ranks of channels are similar. The lab-based simulation experiment was conducted with eighteen volunteers, the later executed field study with twelve voluntary test persons.

*Trace-driven simulation:* Eighteen persons (15 male, 3 female subjects) in the age range from $age_{min}$=18 to $age_{max}$=38 years ($\overline{age}$=25.00 years, $\sigma_{age}$=5.12 years) participated in the first experiment on trace-driven simulation (Figure 4). All of the test persons were relatives, colleagues at the university and students with a valid driving license and a driving experience of on average more than seven years. Not a single one was in a relationship with our department, had previous knowledge about the aim of the experiments nor experiment related interaction with participants driven previously. Male subjects vary in age from $age_{min}$=18 to $age_{max}$=38 years ($\overline{age}$=24.80 years, $\sigma_{age}$=5.41 years). Female test persons vary in age from $age_{min}$=22 to $age_{max}$=30 years ($\overline{age}$=26.00 years, $\sigma_{age}$= 4.00 years).

*On-the-road study:* In contrast, the second study was conducted on a smaller group of twelve test persons whereof 7 subjects were male (58.33%) and 5 participants were female (41.67%). Most of the attendees of this second experiment already participated in the first experiment. Male subjects vary in age from $age_{min}$=25 to $age_{max}$=55 years ($\overline{age}$=33.43 years, $\sigma_{age}$=10.63 years). Female test persons vary in age from $age_{min}$=26 to $age_{max}$=52 years ($\overline{age}$=39.20 years, $\sigma_{age}$= 11.21 years), the overall mean age is 35.83 years and the standard deviation 10.78 years.

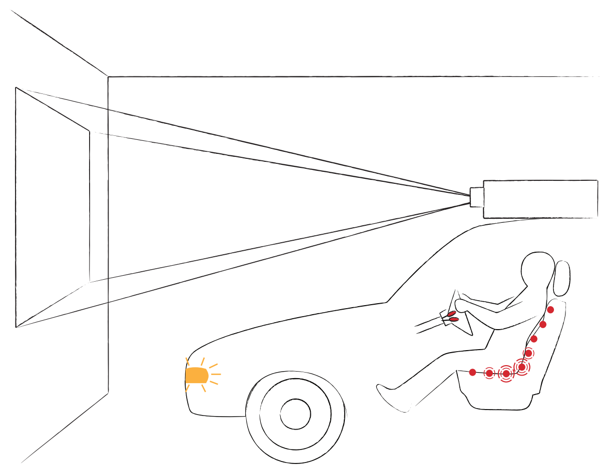


Figure 4. Scribble of the driving simulator setting as used for trace-driven simulation. The car was parked in a darkened garage with vehicle controls connected to a microcontroller. A video projector was used to generate a close-to-reality street view in the front shield.

Although the first experiment was conducted on a larger group compared to the second, results of the two series should be directly comparable. Nevertheless, one particular issue, age dependency, has to be considered accessorily as it has been evidenced earlier that the age of test persons has a major impact on the measured reaction time [7, p. 224] ("*[..] the ability to respond decreases with age.*"). From the difference in the mean age of the two test series (35.83-25.00=10.83 years) we can already prior to experimental results draw the conclusion that the average reaction time is higher in the real driving study (=slower reaction) compared to the simulator experiment conducted with younger people.

*Trace-Driven Simulation versus Real Driving Studies*

According to [53, p. 91] our simulator belong, as the bigger part of all simulators used in this domain, to the class of simulators with *relative* validity (*absolute* validity is the "ultimate goal" to achieve, but can only be reached using high fidelity, high priced driving simulators – and even for those, this criterion has only been demonstrated rarely to date). Nevertheless, the success of any simulation model or simulator is based on how effective a simulator can translate real-world situations and the manner that physical elements for the real world that plays an active role in the choice process are represented [83]. For the comparative study in this work a trace-driven simulation approach was chosen, operating, rather than as a common driving simulator, as tool for real-time driving decision-making.

The "simulator" was deployed to investigate the performance of driver response times using different notification channels on a prerecorded, typical trip across Linz – detailed findings from the tests are given in [7], [31].

Beside the accepted motivation for processing simulation experiments, such as safety, feasibility, independency, repeatability, or comparability, the aim in these series was more to assess a "kind of baseline" for the on-the-road studies to be performed later.

After extensive tests with the simulated driving environment and several modifications in the driver-vehicle interaction loop with the aim that the simulator was intuitively understood by all test participants, the final setting of the simulation experiment, as used in the first study, was transferred to and repeated in a similar designed on-the-road scenario. The goal in the real driving experiment was to provide evidence for similar system behavior embodied by a comparable reaction performance or workload of the driver. On successful proof this would legitimate further engagement in improving the driving simulator by considering parameters influencing the real driving performance to behave as realistic as possible in terms of cognitive workload, distraction, and reaction time in order to finally replace any real driving study with an equivalent simulation run.

Tests in real traffic situations requires a substantially higher effort for preparation compared to experiments with a simulator. Before starting to drive each test person got a detailed initial training ("dry simulation") in order to avoid (or at least reduce) the probability of later accidents or danger situations due to misconceived action triggers to a minimum. This preparatory stage lasts about $20min.$ per person, the driving experiment started immediately afterwards and took, on average, $34.3min.$ for the $\approx 26km$ round trip.

Basically, the setting was similar to that of the simulation environment; however, the number of action points was with 35 lower than in the first series (44). Each test person had to drive exactly the same predefined route (which was also different to that used in the trace-driven simulation) with notifications delivered on specific points of the route using random feedback channels. Visual notifications were given on small displays ("jumbo LEDs") placed left and right on top of the dashboard, auditory information was delivered via headphones, and vibro-tactile information was transmitted via sixteen tactor elements (two strips of eight each) integrated into the car seat (Figure 5). For measuring reaction times, the signals from the control elements activated in reality (light switch, turn indicator) were captured by a microcontroller and forwarded to and post processed in the data analysis unit (standard notebook).

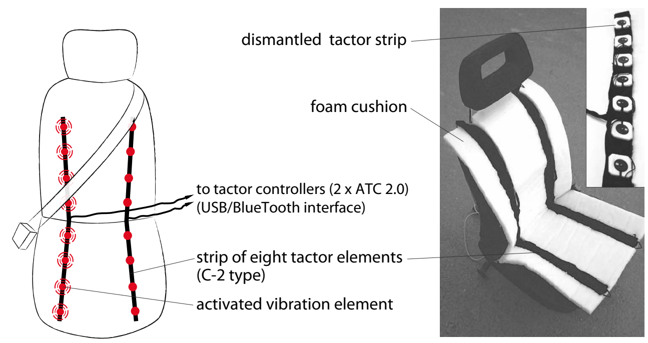



Figure 5. Vibro-tactile notifications were delivered via the driver seat. Therefore, two strips of eight voice coil transducers each were embedded into the seat. The foam cushion was applied to ensure comfortable sitting.

## V. Findings

Table I gives a summary of data analysis separated for the two experimental series. Particularly the mean reaction time ($\overline{x}$) and the SD ($\sigma$) are of interest for further examinations with respect to the comparability of the experiments (see Table II).

Table II
Results show both increased average reaction times and standard deviation for real-driving journeys compared to trace-driven simulation experiments.

| Attribute | Reaction Time, Increase in% | | | Order |
|---|---|---|---|---|
| $TD \rightarrow R$ | $\overline{x}$ | $\widetilde{x}$ | $\sigma$ | $TD, R$ |
| CI 5% (752, 353 datasets) | | | | |
| Combined | 12.8 | 23.2 | -5.1 | -, - |
| Visual | 24.8 | 33.7 | 11.1 | 2, 2 |
| Auditory | 4.4 | 10.9 | 10.8 | 3, 3 |
| Vibro-tactile | 27.4 | 29.7 | 16.9 | 1, 1 |

### A. Variance

From Table I it can be assessed that the standard deviation of reaction times is similar for both test series ($349.9ms$ trace-driven, $331.9ms$ on-the-road; difference of $5.14\%$ in favor of real-driving studies). Data inspection on modality level (Table II) shows more representative results – the differences are here $10.8\%$ for auditory notifications, $11.1\%$ for visual, and $16.9\%$ for vibro-tactile stimulation, in each case in favor of trace-driven simulation.

*Interpretation:* The variance of reaction time over all participants for a certain experiment is relatively stable at $300ms$ (see Table I). Cross *et al.* [84] observed increased stability of (finger) reaction times as an inverse function of age – examining the individual notification channels confirms this finding as the mean age in the case of simulation is $25.0$ years compared to $35.8$ years for the real driving study.

It can be assumed that the variance of reaction time is rather independent from the channel of notification and whether the experiment is processed in the real or as simulation – the only reason for the larger variability should be based on the age difference of person groups.

For definitive confirmation it would be essential to conduct further studies with test persons in a broader range of age, e. g., age group $18$ to $65$. Initial evaluations with respect to age were presented in Riener [85] for the trace-driven experiment – the order of variance ($364.94ms$ for the group of persons aged 25 years or below, $337.37ms$ for the group older than 25 years) follows the findings presented by Cross and Luper [84], and are confirmed within this work.

### B. Reaction Time

Reaction time is attributed to cover $(i)$ the time required to perceive the need for an action, $(ii)$ thoughts about how to solve the problem, $(iii)$ the selection of a solution, and $(iv)$ the initiation of motor actions. The mean reaction time for the two experiments differ by $12.8\%$ (Table II) in favor of simulation. When comparing the on-the-road studies to the simulation experiment based on individual modalities, the reaction time increase is $4.4\%$ for auditory, $24.8\%$ for visual, and $27.4\%$ for vibro-tactile delivered notifications (visually represented in Figure 6).

*Interpretation:* The significant increase in reaction time for on-the-road studies, discoverable over all modalities, can be explained by several reasons and is confirmed, at least to some extent, by surveys carried out in connection with the experiments.

The driver, while steering a “real car”, is (or should be, for safety reasons) focused on the main task of driving. Therefore, Erp [86, p. 2] proposed the safety strategy “*[..] vehicle operation should allow the driver to have both eyes on the road, both hands on the steering wheel, and both feet on or near the pedals*”. In the last time it can be increasingly observed that he/she is distracted by secondary (operation of driver assistance and/or information systems) or tertiary tasks (communication with passengers, adjusting car stereo, and operation of other comfort/entertainment services) in the car [87], [88]. Even if these additional distraction factors were limited as good as possible for the conducted experiments (e. g., no car passengers, car stereo switched off, air conditioning system fixed, etc.), the driver is distracted to some extent, and has therefore limited capacity free for perceiving and reacting to actions.

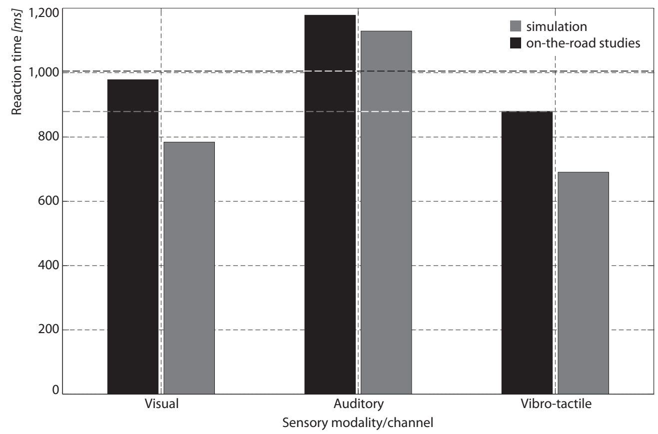



Figure 6. Mean reaction times for the three feedback channels visual, auditory, and vibro-tactile, isolated for simulation (gray) and real journeys (black). Dotted lines at 889.2 and 1,003.2ms indicate the overall mean reaction times.

The large increase in the mean reaction time between simulation and on-the-road study for the visual notification channel ($24.8\%$) (Table II) can be explained by the fact that driving is mostly a visual task, demanding much higher attention when driving in real traffic compared to controlling a simulator. The behavior of the driver in the simulation environment has no impact to the real world (there is no “real danger” – neither for the driver, nor for other road participants, pedestrians or the infrastructure), so that the driver can completely focus on the task of vehicle control. Furthermore, visual notifications were overlayed to the replayed video in the simulation while this information was provided around the dashboard for the real experiment. This requires glances with re-focusing the eye for the latter study, which is well known to require some extra time. Results would be definitely better comparable when showing information in the on-the-road experiment in a more natural way, using, for instance, a head-up display to provide the driver with information without taking his/her eyes from the windscreen.

The vibro-tactile stimulation channel, ascribed to be uninfluenced from the cognitive load of visual and auditory senses, was added accessorily. It is supposed that the increase in the mean reaction time between simulation and real driving tests ($27.4\%$, see Table II) results from the uncommonness of using the sense of touch as information carrier. For the trace-driven simulation, test persons are willing to trust this modality, but in real-driving studies users are more cautious as operating errors are prone to run unnecessary risks. The histograms shown in Figure 7 reflect these assumptions as the reaction time is on average lower with less variance for the first (simulation) compared to the latter experiment (on-the-road studies). We are confident that the great difference in reaction time declines (or even disappears) with common utilization of the sense of touch

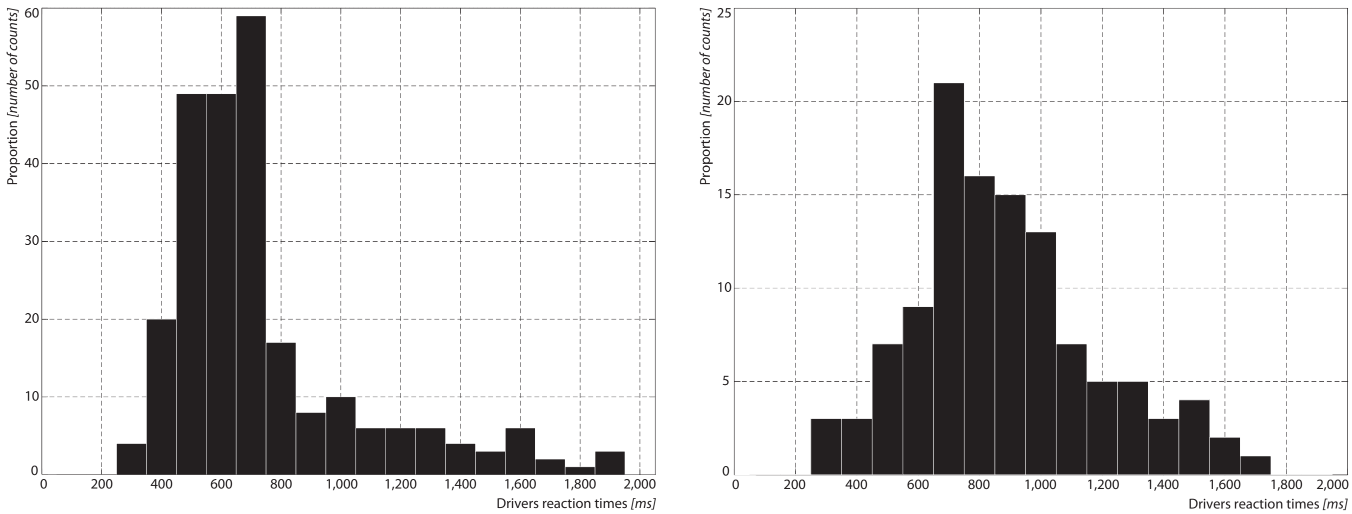



Figure 7. Histograms for trace-driven simulation (left) and on-the-road experiments (right) for the vibro-tactile modality. The shape of the plot shows that reaction times are lower with less variance for the case of simulation compared to real-driving studies.

in driver-vehicle interfaces (the characteristic of touch-based feedback was already discussed before, e. g., in [7, p, 194]).

### C. Further Results

The surreal behavior of a driving simulator is another factor influencing the performance compared to on-the-road studies. For simulation, we found in particular that $(i)$ a discoverable lower concentration is on the task of driving due to the risk-free environment, $(ii)$ traffic rules (road signs such as speed limits) and road traffic regulations can be (and are) ignored by reason of no punishment on delicts, and $(iii)$ the general unreal behavior of the simulation environment (absence of engine and environmental noise, road vibrations, etc.) has a negative impact on the validity of results.

It should also be noted that the comparison lacks of equal preconditions, for instance in the simulated experiment – contrary to the real driving study – test persons were actually not involved in a driving task (they had only to watch the video and react on a requested feedback using the real control instruments of the car). Future simulation settings, desired to provide a more realistic behavior, should be designed under the guidelines to cover the identified issues.

## VI. Conclusion and Future Work

With increasing pressure regarding production cost and time, automobile manufacturer are requested to apply simulation to all stages of product development including user interface evaluation. The application of driving simulators is, particularly for the last item, a great challenge as a person's behavior cannot be described by mathematical equations or physical rules. Moreover, and beside the primary task of interaction with the vehicle, driver behavior depends on several subsidiary factors of influence.

In order to gain insight in vehicle steering performance, two driver-car interaction experiments of similar type have been designed and processed, first, a *labor study* (trace-driven simulation) and second, a *real driving experiment* (on-the-road journey). Comparing the average reaction times of drivers based on notifications using different sensory channels confirmed that the two instances perform similar; for both studies, response time from vibro-tactile delivered commands were lowest, followed by responses from visual and auditory stimulation. Furthermore, results have shown that the reaction time in real driving situations is on average $13\%$ higher compared to simulated driving. The main reason for this large difference is, that the simulation environment is only a imprecise, low detailed copy of the real environment (static car in a garage with a video of the prerecorded track displayed on the front shield; no real driving task). Therefore, the difference cannot be considered as universal, linear correction factor to obtain real response times from driving simulator studies.

According to the detailed elaboration of simulator *validity* and *fidelity*, there are two strategies for future improvements. The first is to use a more sophisticated simulator for upcoming experiments, providing an immersive environment with close-to-reality behavior (road vibrations, engine noise, penalty models for speeding, etc.). With such a simulator, using enhanced settings to cover the discussed problems, it should be feasible to estimate the differences in reaction time between simulator and on-the-road study in a more precise way. Such high end simulators must not be developed in-house. There are several research institutions owning (and leasing) simulators, such as, for instance, simulators at TNO Netherlands [89], DLR Germany [90], TRW Automotive [91], or IZVW Wurzburg/Germany [92]. Nevertheless, high fidelity simulators are very expensive (even to rent) and it has been shown that also these systems cannot guarantee validity. From this state of knowledge, it should be clear to put not more effort into the development of high fidelity simulators.

The second option is to improve both quality and comparability of the existent systems based on the findings from the initial experiment series summarized in this work. One issue to cope with is, for example, to adjust the cognitive workload the driver is exposed to in both settings to be similar, e. g., by adding a task to the simulation experiment or by increasing the complexity of the simulator to behave more realistic. It is supposed, that simulator outcome would then be better comparable to real driving results. If approved, a metrics table could be provided, containing rows for notification channels and columns for the level of simulator fidelity, e. g. low fidelity simulator, visual notification, add $x\%$ to get real world behavior or high-tech simulator, visual channel, add $y\%$ (with $y < x$); values should be provided for all the modalities.

## Acknowledgments

This work is supported under the FP7 ICT Future Enabling Technologies program of the European Commission under grant agreement No. 231288 (SOCIONICAL).

## References

[1] Andreas Riener. Simulating On-the-Road Behavior Using a Driving Simulator. In *Third International Conference on Advances in Computer-Human Interactions (ACHI '10)*, pages 25–31, February 2010. ISBN: 978-1-4244-5693-2.

[2] W. Fornaciari, M.V. Angiolillo, and R. Farina. Seamless design space exploration for automotive systems. In *Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on*, pages 241–244, 2007.

[3] Sergio Damiani, Enrica Deregibus, and Luisa Andreone. Driver-vehicle interfaces and interaction: where are they going? *European Transport Research Review*, 1(2):87–96, July 2009.

[4] Alois Ferscha and Andreas Riener. Pervasive Adaptation in Car Crowds. In *First International Workshop on User-Centric Pervasive Adaptation (UCPA) at MOBILWARE 2009, Berlin, Germany*, page 6. Springer-Verlag Berlin Heidelberg, April 27 2009.

[5] Simon Nestler, Marcus Tnnis, and Gudrun Klinker. Common interaction schemes for in-vehicle user-interfaces. *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*, pages 159–168, 2009.

[6] SOCIONICAL. Complexity Science based modeling, prediction and simulation methods for large scale socio-technical systems. [Online]. last retrieved January 9, 2011. http://www.socionical.eu.

[7] Andreas Riener. *Sensor-Actuator Supported Implicit Interaction in Driver Assistance Systems*. Vieweg+Teubner Research, Wiesbaden, Germany, 1st (January 14, 2010) edition, January 2010. ISBN-13: 978-3-8348-0963-6.

[8] BMW. BMW Technology Guide: Head-Up Display. [Online]. last retrieved January 9, 2011. http://www.bmw.com/com/en/insights/technology/technology_guide/articles/head_up_display.html?source=index&article=head_up_display.

[9] N.J. Ward and A. Parkes. Head-up displays and their automotive application: An overview of human factors issues affecting safety. *Accident Analysis & Prevention*, 26(6):703–717, 1994.

[10] K. Nakamura, J. Inada, M. Kakizaki, T. Fujikawa, S. Kasiwada, H. Ando, and N. Kawahara. Windshield Display for Safe and Comfortable Driving. *SAE World Congress on Intelligent Vehicle Initiative (IVI) Technology 2005, Advanced Controls and Navigation Systems*, 2005.

[11] Tony Swan. 2009 Tokyo Auto Show: At Nissan, Driving Pleasure Includes Olfactory Gratification. [Online.], October 22, 2009. last retrieved January 17, 2011. http://blog.caranddriver.com/2009-tokyo-auto-show-at-nissan-driving-pleasure-includes-olfactory-gratification/.

[12] Gran Kecklund, Torbjrn Akerstedt, David Sandberg, Mattias Wahde, Tania Dukic, Anna Anund, and Magnus Hjlmdahl. State of the art review of driver sleepiness. Deliverable 1.1, Intelligent Vehicle Safety Systems (IVSS) program project, April 2007. [Online]. last retrieved January 17, 2011. http://www.vti.se/11934.epibrw.

[13] Kyoko Kuroda, Naohiko Inoue, Yuriko Ito, Kikue Kubota, Akio Sugimoto, Takami Kakuda, and Tohru Fushiki. Sedative effects of the jasmine tea odor and (r)-(–)-linalool, one of its major odor components, on autonomic nerve activity and mood states. *European Journal of Applied Physiology*, 95(2):107–114, October 2005.

[14] Joseph Nathaniel Kaye. Symbolic Olfactory Display. Master thesis, School of Architecture and Planning, Program in Media Arts and Sciences, MIT, May 2001. [Online]. last retrieved January 11, 2011. http://alumni.media.mit.edu/~jofish/thesis/symbolic_olfactory_display.html.

[15] Adam Bodnar, Richard Corbett, and Dmitry Nekrasovski. AROMA: ambient awareness through olfaction in a messaging application. In *Proceedings of the 6th international conference on Multimodal interfaces (ICMI'04)*, pages 183–190, New York, NY, USA, 2004. ACM.

[16] Olga Pollatos, Rainer Kopietz, Jennifer Linn, Jessica Albrecht, Vehbi Sakar, Andrea Anzinger, Rainer Schandry, and Martin Wiesmann. Emotional Stimulation Alters Olfactory Sensitivity and Odor Judgment. *Chem. Senses*, 32(6):583–589, 2007.

[17] Alan M. Christie. Simulation: An Enabling Technology in Software Engineering. In *The Journal of Defense Software Engineering*, page 6. Software Engineering Institute, Carnegie Mellon University, 1999.

[18] Jeffrey Adler, Michael McNally, and Will Recker. Interactive Simulation for Modeling Dynamic Driver Behavior in Response to ATIS. TR UCI-ITS-TS-WP-93-4, Institute for Transportation Studies, UCI, 1993.

[19] J. Baujon *et al.* A new low-cost tool for driver behavior analysis and studies. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 569–574, 2000.

[20] Ying Wang, Wei Zhang, Su Wu, and Yang Guo. Simulators for Driving Safety Study – A Literature Review. *Virtual Reality*, pages 584–593, 2007.

[21] Yi-hu Wu, Dan Yu, and Xiaomei Yin. A simulation of car-following driving based on coordinated model. In *ICICTA '08: Proceedings of the 2008 International Conference on Intelligent Computation Technology and Automation*, pages 219–224, Washington, DC, USA, 2008. IEEE Computer Society.

[22] S.F. Zhao, G.H. Xu, and T.F. Tao. Modeling human vehicle driving which integrate models of fatigue. In *4th IEEE Conference on Industrial Electronics and Applications (CIEA 2009)*, pages 3493–3497, 2009.

[23] T. Von Jan, T. Karnahl, K. Seifert, J. Hilgenstock, and R. Zobel. Dont sleep and drive – VW's fatigue detection technology. In *Proceedings of 19th International Conference on Enhanced Safety of Vehicles, Washington, DC*, 2005.

[24] Aladino Amantini and Pietro Cacciabue. A simple simulation predicting driver behavior, attitudes and errors. *Digital Human Modeling*, pages 345–354, 2009.

[25] Jiangpeng Dai, Jin Teng, Xiaole Bai, Zhaohui Shen, and Dong Xuan. Mobile phone based drunk driving detection. In *4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 1–8, March 22–25, 2010. ISBN: 978-963-9799-89-9.

[26] M.E. Rakauskas, N.J. Ward, E.R. Boer, E.M. Bernat, M. Cadwallader, and C.J. Patrick. Combined effects of alcohol and distraction on driving performance. *Accident Analysis & Prevention*, 40(5):1742–1749, 2008.

[27] Y. Wang and W. Zhang. A Driver Training Platform Prototype Based on Distributed Simulation. In *International Conference on Cyberworlds 2008*, pages 506–510. IEEE, 2008.

[28] L.P. Lonero. Trends in driver education and training. *American journal of preventive medicine*, 35(3S):316–323, 2008.

[29] J.M. Leitão, A. Moreira, J.A. Santos, A.A. Sousa, and F.N. Ferreira. Evaluation of driving education methods in a driving simulator. *Computer Graphics and Visualization Education'99*, 1999.

[30] Chao Sun, Feng Xie, Xiaocao Feng, Mingmin Zhang, and Zhigeng Pan. A Training Oriented Driving Simulator. *Entertainment Computing (ICEC 2007)*, pages 1–9, 2007.

[31] Andreas Riener and Alois Ferscha. Simulation Driven Experiment Control in Driver Assistance Assessment. In David Roberts et al., editor, *12th IEEE International Symposium on Distributed Simulation and Real Time Applications, Vancouver, BC, Canada*, page 10. IEEE CS Press, P3425, October 27–29 2008. ISBN: 978-0-7695-3425-1.

[32] Paul F. Smith, Jie Chen, and Hongxing Hu. Model-Based Design Study and Evaluation of New HMI Concepts for Vehicle Multimedia, Climate Control and Navigation Systems. *SAE World Congress & Exhibition, April 2007, Detroit, MI, US*, page 9, 2007.

[33] Wan Jian, Chu Xiumin, Wu Yong, and Zhang Rui. The Design of Autonomous Smart Car Used in Simulation of Vehicle Platoon. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 1, pages 885 –890, 19-20 2008.

[34] P. Cheevarunothai, Y. Wang, and N.L. Nihan. Development of a loop detector simulator (LOOPSIM) for in-laboratory traffic research and education. *Computer Applications in Engineering Education*, 16(1):45–54, 2008.

[35] J. Santos *et al.* The interaction between driving and in-vehicle information systems: Comparison of results from laboratory, simulator and real-world studies. *Transportation Research Part F: Psychology and Behaviour*, 8(2):135–146, 2005.

[36] F. Panerai *et al.* Speed and safety distance control in truck driving: comparison of simulation and real-world environment. In *Proceedings of Driving Simulation Conference*, pages 91–107, 2001.

[37] C. Lange, M. Wohlfarter, and H. Bubb. Vergleichbarkeit von Usability Lab und Realversuch zur Bestimmung der ergonomischen Guete und der Ablenkungswirkung von Nebenaufgaben im Kfz. *Proceedings GfA Gesellschaft fuer Arbeitswissenschaften Kongress, Stuttgart*, 2006.

[38] T.Y. Koo, K.J. Park, B.Y. Kim, H.J. Kim, and M.W. Suh. A study on drivers workload of telematics using a driving simulator: A comparison among information modalities. *International Journal of Precision Engineering and Manufacturing*, 10(3):59–63, 2009.

[39] J. Sodnik, C. Dicke, S. Tomazic, and M. Billinghurst. A user study of auditory versus visual interfaces for use while driving. *International journal of human-computer studies*, 66(5):318–332, 2008.

[40] G. Costagliola, S. Martino, and F. Ferrucci. A Simulation Environment to Assess Driving Performances while Interacting with On-board Telematics Systems. *Enterprise Information Systems*, pages 439–451, 2008.

[41] T.Y. Koo, C.H. Bae, B.Y. Kim, Z. Rowland, and M.W. Suh. Development of a driving simulator for telematics human–machine interface studies. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 222(11):2077–2086, 2008.

[42] D.D. Salvucci. Rapid prototyping and evaluation of in-vehicle interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(2):1–33, 2009.

[43] J. Barceló, J.L. Ferrer, and R. Martin. Simulation assisted design and assessment of vehicle guidance systems. *International Transactions in Operational Research*, 6(1):123–143, 1999.

[44] H.J. Bullinger and M. Dangelmaier. Virtual prototyping and testing of in-vehicle interfaces. *Ergonomics*, 46(1):41–51, 2003.

[45] P. Spethmann, C. Herstatt, and S.H. Thomke. Crash simulation evolution and its impact on R&D in the automotive applications. *International Journal of Product Development*, 8(3):291–305, 2009.

[46] H. Burg, A. Moser, and H. Steffan. *Handbuch Verkehrsunfallrekonstruktion*, chapter Simulation und Animation, pages 585–592. Springer, 2009.

[47] R. Thomson, M. Edwards, T. Martin, C. van der Zweep, R. Damm, and G. della Valle. Car-car crash compatibility: development of crash test procedures in the VC-Compat project. *International Journal of Crashworthiness*, 12(2):137–151, 2007.

[48] E. Guilmineau and F. Chometon. Effect of side wind on a simplified car model: Experimental and numerical analysis. *Journal of Fluids Engineering*, 131:021104, 2009.

[49] E. Mercker and H.W. Knape. Ground simulation with moving belt and tangential blowing for full-scale automotive testing in a wind tunnel. In *Society of Automotive Engineers international congress and exposition*, volume 27, 1989.

[50] W. Eckert, N. Singer, and J.D. Vagt. The Porsche Wind Tunnel Floor-Boundary-Layer Control–A Comparison With Road Data and Results From Moving Belt. 1992.

[51] K. Burgin, P.C. Adey, and J.P. Beatham. Wind tunnel tests on road vehicle models using a moving belt simulation of ground effect. *Journal of Wind Engineering and Industrial Aerodynamics*, 22(2-3):227–236, 1986.

[52] H. Kwon, Y.W. Park, D. Lee, and M.S. Kim. Wind tunnel experiments on Korean high-speed trains using various ground simulation techniques. *Journal of Wind Engineering and Industrial Aerodynamics*, 89(13):1179–1195, 2001.

[53] Kristie L. Young, Michael A. Regan, and John D. Lee. *Driver Distraction: Theory, Effects, and Mitigation*, chapter Measuring the Effects of Driver Distraction: Direct Driving Performance Methods and Measures, pages 85–105. CRC Press, 2009. ISBN: 978-0-8493-7426-5.

[54] M. Pantic, A. Pentland, A. Nijholt, and T. Huang. Human computing and machine understanding of human behavior: a survey. *Artifical Intelligence for Human Computing*, pages 47–71, 2007.

[55] B.G. Silverman, M. Johns, J. Cornwell, and K. O'Brien. Human behavior models for agents in simulators and games: part I: enabling science with PMFserv. *Presence: Teleoperators & Virtual Environments*, 15(2):139–162, 2006.

[56] Giuseppe Andreoni, Marco Rabuffetti, and Antonio Pedotti. Simulation of Complex Human Movement Through the Modulation of Observed Motor Tasks. *Digital Human Modeling*, pages 3–12, 2007.

[57] F. Aparicio, F. Jiménez, and J. Sánchez. Development and use of vehicle dynamics simulation software as support for road vehicles theory teaching. *Computer Applications in Engineering Education*, 17(4):467–478, 2009.

[58] Dieter Ammon, Michael Gipser, Jochen Rauh, and Jrgen Wimmer. High performance system dynamics simulation of the entire system tire-suspension-steering-vehicle. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 27(5):435–455, 1997.

[59] F. Colombet, M. Dagdelen, G. Reymond, C. Pere, F. Merienne, and A. Kemeny. Motion Cueing: what's the impact on the driver's behaviour. In *In Proceeding of the Driving Simulator Conference (DSC 2008)*, pages 171–181, 2008.

[60] G. Reymond and A. Kemeny. Motion cueing in the Renault driving simulator. *Vehicle System Dynamics*, 34(4):249–259, 2000.

[61] A. R. Valente Pais, M. Wentink, M. M. van Paassen, and M. Mulder. Comparison of three motion cueing algorithms for curve driving in an urban environment. *Presence: Teleoper. Virtual Environ.*, 18(3):200–221, 2009.

[62] J. Engström, E. Johansson, and J. Östlund. Effects of visual and cognitive load in real and simulated motorway driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8(2):97–120, 2005.

[63] Albert J. Rehmann. A Handbook of Flight Simulation Fidelity Requirements for Human Factors Research. Technical Report DOT/FAA/CT-TN95/46, U.S. Department of Transportation Federal Aviation Administration, December 1995.

[64] Matthew P. Reed and Paul A. Green. Validation of a Low-Cost Driving Simulator Using a Telephone Dialing Task. Final Report UMTRI-95- 19, University of Michigan, Transportation Research Institute, June 1995.

[65] Nico Kaptein, Jan Theeuwes, and Richard Van Der Horst. Driving simulator validity: Some considerations. *Transportation Research Record: Journal of the Transportation Research Board*, 1550(-1):30–36, January 1996.

[66] Gary Burnett, A. Irune, and A. Mowforth. Driving Simulator Sickness and Validity: How Important Is It to Use Real Car Cabins? *International Conference on Road Safety and Simulation (RSS 2007), Rome, Italy*, page 6, November 7–9 2007.

[67] B. Reimer, L.A. D'Ambrosio, J.F. Coughlin, M.E. Kafrissen, and J. Biederman. Using self-reported data to assess the validity of driving simulation data. *Behavior research methods*, 38(2):314, 2006.

[68] H. Ashton, R.D. Savage, J.W. Thompson, and D.W. Watson. A method for measuring human behavioural and physiological responses at different stress levels in a driving simulator. *British Journal of Pharmacology*, 45(3):532, 1972.

[69] H.C. Lee. The validity of driving simulator to measure on-road driving performance of older drivers. *Transport Engineering in Australia*, 8:89–100, 2003.

[70] Hurshkumar G. Donde, Rakesh K. Lad, Prabal Kumar Biswas, Praveen B. Patil, and Gordon Stace. Reduction in Time to Market of Automotive Seating System Using LS-DYNA . In *Proceedings of the 7th International LS-DYNA Users Conference, Detroit*, pages 9–17 – 9–30 (14), Electronics City, Bangalroe 561 229, India, July 2002. Infosys Technologies Limited.

[71] Larry Gould. GM Speeds Time to Market Through Blistering Fast Processors. *Automotive Design & Production, Gardner Publications, Inc.*, July 2004. [Online]. last retrieved January 11, 2011. http://www.autofieldguide.com/articles/070407.html.

[72] Alfred T. Lee. *Flight Simulation: Virtual Environments In Aviation*. Ashgate Publishing Limited, Hampshire, England, 2005. ISBN: 0-7546-4287-9.

[73] Federal Aviation Administration. Airplane Simulator Qualification. Advisory circular 120-40b, U.S. Department of Transportation, Washington, D.C., July 1991.

[74] Michael D. Proctor, Maria Bauer, and Thomas Lucario. Helicopter Flight Training through Serious Aviation Gaming. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology (JDMS)*, 5(1):18, January 2008.

[75] Thomas Longridge, Judith Brki-Cohen, Tiauw H. Go, and Andrew J. Kendra. Simulator fidelty considerations for training and evaluation of today's airline pilots. In *Proceedings of the 11th International Symposium on Aviation Psychology*, page 7, March 2001.

[76] Dave Hirschmann. Simulation nation: Flight simulator training for general aviation. *Aircraft Owners and Pilots Association (AOPA) Flight Training*, page 3, January 2009.

[77] Samantha Jamson, Frank Lai, Hamish Jamson, Anthony Horrobin, and Oliver Carsten. Interaction between Speed Choice and the Environment. Road safety research report 100, Department for Transport, Great Minster House, 76 Marsham Street, London SW1P 4DR, 2008. ISBN: 978-1-906581-39-8.

[78] Andrew Kun, Tim Paek, and Zeljko Medinca. The effect of speech interface accuracy on driving performance. In *Proceedings of 8th Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)*, Antwerp, Belgium, August 27–31 2007. Microsoft Research, USA and University of New Hampshire, USA.

[79] Susan Chisholm, Jeff Chaird, Lulie Lockhart, Lisa Fern, and Elise Teteris. Driving Performance while Engaged in MP3 Player Interaction: Effects of Practice and Task Difficulty on PRT and Eye Movements. In *Proceedings of the 4th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, pages 238–245 (8), Skamania Lodge, Stevenson, Washington, USA, July 9–12 2007.

[80] N. N. Making a driving simulator even better – Improved imagery and simulation have increased safety and performance at Ford. *automotive electronics*, pages 10–12, August/September 2008.

[81] Hindrik W.J. Robbe. Marijuana use and driving. *Journal of the International Hemp Association*, 1:44–48, 1994.

[82] Race Technology Ltd. Automotive Technical Excellence – Products. [Online]. last retrieved January 9, 2011. http://www.race-technology.com/products_2_3.html.

[83] Reiner Suikat. The new Dynamic Driving Simulator at DLR. In *Driving Simulator Conference, Orlando, Florida*, pages 374–381 (8), November 30 – December 2 2005.

[84] Douglas E. Cross and Harold L. Luper. Relation between finger reaction time and voice reaction time in stuttering and nonstuttering children and adults. *J Speech Hear Res*, 26(3):356–361, September 1983.

[85] Andreas Riener. Age- and Gender-Related Studies on Senses of Perception for Human-Vehicle-Interaction. In *Proceedings of the 2nd Workshop on Automotive User Interfaces and Interactive Applications, September 7-10, 2008, Lübeck, Germany*, page 8. GI-Edition 2008, September 2008.

[86] J.B.F. Van Erp and H. Van Veen. Vibro-tactile information presentation in automobiles. In *Proceedings of Eurohaptics*, volume 2001, pages 99–104, 2001.

[87] Juergen Schwarz et al. Code of Practice for the Design and Evaluation of ADAS. PReVENT Report 11300, v1.6, Response 3, a PReVENT Project, August 2006. Preventive and Active Safety Applications, Integrated Project, Contract number FP6-507075.

[88] Marcus Toennis, Verena Broy, and Gudrun Klinker. A Survey of Challenges Related to the Design of 3D User Interfaces for Car Drivers. In *Proceedings of the 3D User Interfaces (3DUI'06)*, pages 127–134, Washington, DC, USA, 2006. IEEE Computer Society.

[89] Netherlands Organisation for Applied Scientific Research (TNO). Automotive. [Online]. last retrieved January 9, 2011. http://www.tno.nl/groep.cfm?context=markten&content=markt&laag1=59&item_id=59.

[90] German Aerospace Center (DLR). Institute of Transportation Systems – Dynamic Driving Simulator. [Online]. last retrieved January 9, 2011. http://www.dlr.de/fs/en/desktopdefault.aspx/tabid-1236/1690_read-3257.

[91] TRW Automotive. Cognitive Safety Systems. [Online]. last retrieved January 9, 2011. http://www.trwauto.com.

[92] Center for Traffic Sciences (IZVW). Driving simulator with motion platform. [Online]. last retrieved January 9, 2011. http://www.psychologie.uni-wuerzburg.de/methoden/forschung/technik/fahrsimulator.php.en.

[93] T. Bock, M. Maurer, and G. Farber. Validation of the Vehicle in the Loop (VIL) – A milestone for the simulation of driver assistance systems. In *2007 IEEE Intelligent Vehicles Symposium*, pages 612–617, 2007.

[94] Yaman Barlas. Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, 12(3):183–210, 1996.

[95] Gian Beeli, Susan Koeneke, Katja Gasser, and Lutz Jancke. Brain stimulation modulates driving behavior. *Behav Brain Funct.*, 4(34):7, August 6, 2008. [Online].

[96] Monica Bordegoni, Umberto Giraudo, Giandomenico Caruso, and Francesco Ferrise. Ergonomic interactive testing in a mixed-reality environment. In *Proceedings of the 2nd International Conference on Virtual Reality (ICVR 2007), Held as part of HCII'07, Beijing, China*, pages 431–440, July 22–27 2007.

[97] N. Dahlstrom, S. Dekker, R. van Winsen, and J. Nyce. Fidelity and validity of simulator training. *Theoretical Issues in Ergonomics Science*, 10(4):305–314, 2009.

[98] Paul Green. How Driving Simulator Data Quality Can Be Improved. In *Driving Simulation Conference North America 2005, Orlando, Florida*, page 11, November 2005.

[99] Yaron Hollander and Ronghui Liu. The principles of calibrating traffic microsimulation models. *Transportation*, 35(3):347–362, May 2008.

[100] A.H. Hoskins and M. El-Gindy. Technical report: Literature survey on driving simulator validation studies. *International Journal of Heavy Vehicle Systems*, 13(3):241–252, 2006.

[101] Maura Houtenbos. *Expecting the unexpected – A study of interactive driving behaviour at intersections*. Hs-043 950, Technical University Delft, The Netherlands, January 2008. pp. 216.

[102] Jack P. C. Kleijnen. Verification and validation of simulation models. *European Journal of Operational Research*, 82(1):145–162, 1995.

[103] Michael A. Regan, John D. Lee, and Kristie L. Young. *Driver Distraction: Theory, Effects, and Mitigation*, chapter Measuring the Effects of Driver Distraction, pages 85–106. CRC Press, 2009. ISBN: 978-0-8493-7426-5.

[104] Kwon Son, Kyung-Hyun Choi, and Sung-Sook Eom. Virtual prototyping simulation for a passenger vehicle. *Journal of Mechanical Science and Technology*, 15(4):448–458, April 2001.

[105] Tomer Toledo and Haris Koutsopoulos. Statistical validation of traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board*, 1876(-1):142–150, January 2004.

[106] Marcel B.F. Uhr, Daniel Felix, B.J. Williams, and Helmut Krueger. Transfer of Training in an Advanced Driving Simulator: Comparison Between Real World Environment and Simulation in a Manoeuvring Driving Task. In *Driving Simulation Conference, North America 2003 (DSC-NA 2003)*, page 11, 2003.

[107] Ying Wang, Bruce Mehler, Bryan Reimer, Vincent Lammers, Lisa A. D'Ambrosio, and Joseph F. Coughlin. The validity of driving simulation for assessing differences between in-vehicle informational interfaces: A comparison with field testing. *Ergonomics*, 53(3):404–420, 2010.

[108] Garrett Weinberg and Bret Harsham. Developing a low-cost driving simulator for the evaluation of in-vehicle technologies. In *AutomotiveUI '09: Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 51–54, New York, NY, USA, 2009. ACM.

# Establishing a Measurement System for IT Service Management Processes: A Case Study

Marko Jäntti
*University of Eastern Finland*
*School of Computing*
*Software Engineering*
*P.O.B 1627, 70211*
*Kuopio, Finland*
*marko.jantti@uef.fi*

Antti Lahtela
*Regional State Administrative*
*Agency for Eastern Finland*
*Development and Steering Unit*
*for the Local Register Offices*
*P.O.B 1348, 70101*
*Kuopio, Finland*
*antti.lahtela@avi.fi*

Jukka Kaukola
*Tieto Finland Oy, Energy*
*P.O.B 1199, 70211*
*Kuopio, Finland*
*jukka.kaukola@tieto.com*

***Abstract*—IT service providers need effective and efficient methods how to design, manage, support and measure IT services. IT Infrastructure Library (ITIL) is the most widely used IT service management framework. It consists of best practices that can be used in implementing, for example service support processes, such as incident management and problem management. Although IT service management frameworks and standards provide some guidelines for measuring IT services, many IT organizations consider service measurement as a difficult task. The research problem in this paper is how the measurement of the IT service support processes can be improved? The main contribution of this study is to 1) describe the implementation process of the ITIL-based IT service management measurement system (ITSM-MS), 2) describe the system architecture and the main functions of the ITSM-MS, 3) propose a framework for measuring IT services and 4) present the lessons learnt from the ITSM-MS implementation process. The ITSM-MS can be used to measure the performance of IT service support processes. The measurement system was developed in cooperation between a software engineering research project and a large IT service provider company in Finland.**

***Keywords*-IT service management; metrics; measurement; system**

## I. INTRODUCTION

IT service providers are continuously looking for more efficient methods to improve the performance of customer support processes and to reduce the support and maintenance costs. Process improvement actions should be based on the reliable and up-to-date measurement data. Service measurement plays an important role in IT service management. This paper is based on the published conference paper [1].

The business objective of this study is to decrease the manual work related to measurement and reporting of IT service performance. Currently, thousands of IT organizations are implementing IT service management processes. Effective service management requires that there is a measurement system that enables measuring each service management process.

This paper provides a deeper literature review on the measurement of IT service support processes, includes a more detailed description how we implemented an IT service management measurement system (ITSM-MS) for IT service support processes in a large IT service provider organization and proposes a systematic framework for measuring IT services. The study is valuable because it describes a unique case that combines both IT service management and dynamic, real-time measurement tool for IT service processes.

The focus in this paper is on process metrics. According to IEEE Standard for a Software Quality Metrics Methodology a process metric is "a metric used to measure characteristics of the methods, techniques, and tools employed in developing, implementing, and maintaining the software system." [2]. We extend this definition to involve also IT services.

Many IT service organizations consider the measurement of IT service management processes, especially service support processes, as a difficult task. Difficulties are mainly due to the following four reasons: 1) IT organizations do not have a structured approach for measuring IT services and service management processes, 2) tools used by service support teams do not enable effective measurement, 3) IT service management standards and frameworks do not provide practical examples how to measure support processes, and 4) there are too many options what to measure in service management.

First, in many IT organizations the measurement activities regarding customer support are still carried out as ad hoc activities without specified business gseoals for measurements, a measurement manager that is responsible for improving the measurement process and tools or clear description of the used metrics. Instead of an ad hoc approach, organizations need a structured approach for measuring IT service support processes. The IT organization should consider measurement and reporting as a systematic process that is managed and improved by a process manager and frequently reviewed and where each metric is linked to business objectives.

Second, service support processes are very tool-oriented processes. Unfortunately often, the service reporting and measurement functions are the weakest parts of service desk tools. The tool should enable effective measurement of both resolution times and volumes of support requests. In the ideal situation, the service measurement system provides real-time measurement data about the process performance.

Third, there is a wide selection of IT service management frameworks and standards that IT organizations can use to improve and manage their processes, such as IT Infrastructure Library (ITIL) [3], ISO/IEC 20 000 [4], Control Objectives for Information and related Technology (COBIT) framework [5], Capability Maturity Model Integration (CMMI) for Services [6], and Microsoft Operation Framework (MOF) [7]. These frameworks and standards should provide IT organizations with more practical examples how to measure IT service support processes.

Fourth, IT organizations can measure their IT operations from many different perspectives. Different stakeholders require different metrics and reports. A typical IT service provider organization deals with software products, software projects, and IT services. Products, projects and services are produced by following organization's processes. IT customers, service managers, customer service managers, product managers, process managers, and business managers have all their own requirements regarding measurement and reporting. It is challenging to create a measurement system that satisfies everybody's requirements.

The remainder of the paper is organized as follows. In Section III the research problem and research methods of this study are described. Section II describes the background and related work for our study. In Section IV, we describe the implementation process of the measurement system, the system architecture, the main functions of the system, and the proposed IT service measurement framework. Section V presents the lessons learnt from the ITSM-MS project. Finally, the conclusions are given in Section VI.

## II. Background and Related Work

Surprisingly few academic studies have dealt with measurement of service support processes (incident management and problem management) from the IT service management perspective. There are studies that have focused on predicting incident volumes through statistical methods [8] and discussing service level agreements [9]. Service monitoring and measurement should begin immediately after the service level agreement is agreed and accepted and start producing service achievement reports [10].

The ITIL version 2 completely ignores the establishment of a measurement process. The Continual Service Improvement (CSI) of the ITIL version 3 is a step to a better direction. It proposes a 7-step improvement process to support IT service management measurement activities [11]. In the CSI, the measurement is based on three basic concepts: critical success factors (for example, reducing IT costs), key performance indicators (for example, 10 percent reduction in the costs of handling printer-related incidents) and metrics (for example, cost of the improvement effort). Unfortunately, the 7-step improvement program seems to be very abstract and difficult to adopt in practice. A potential canditate for a measurement process is Six Sigma but a lot of work is needed to convert it into IT service management purposes.

Six Sigma is a process improvement model that enables organisations to streamline processes by reducing the number of defects [12]. The Six Sigma approach has two key methodologies: 1) DMAIC (Define, Measure, Analyze, Improve or optimize, Control) that can be used to improve an existing business process and 2) DMADV (Define, Measure, Analyze, Design, Verify) that is used to create a new product or a process.

The measurement of the service support processes will be painful if the service desk tools do not include effective measurement functions. Advanced service desk tools enable monitoring whether service level requirements are met in resolving service desk cases. The tool should inform users if the service level agreements are close to breach (yellow warning code) or have already breaches (red code). The service desk tool must be capable of producing both time-based and volume-based performance reports.

If the service reporting and measurement function does not work properly, it will remarkably increase the manual work in producing process performance reports. In many IT organizations, process managers still have to use Microsoft Excel to produce monthly reports regarding customer support processes. For example, Jäntti, Miettinen and Vähäkainu report in their study that time-based performance metrics were difficult to implement due to tool difficulties [13].

It is surprising that although there are various IT service management frameworks and standards available, IT organizations still have problems in creating metrics and measuring the service management processes. The IT Infrastructure Library (ITIL) is the most widely used framework for IT service management. The support processes of ITIL include incident management, problem management, change management, configuration management, and release management [14]. In this paper, we focus on the first two processes that are often called front-end support processes [15].

The main objective of incident management process is to deal with all incidents including failures, questions and queries reported by the users [16]. This process is related to corrective software maintenance [17], [18]. Problem management in turn focuses on preventing problems and incidents, eliminating recurring incidents and minimizing the impact of incidents. Many organizations have difficulties to implement ITIL-based problem management activities [19]. Jäntti has reported that difficult ITIL terminology causes IT organizations challenges especially when the improvement target is the problem management process [20]. Although ITIL has introduced a selection of IT service support metrics both in version 2 [14] and version 3 [16], it does not provide sufficient information how IT service management process measurements should be done in practice.

The incident management process of the ITIL version 3 contains 15 potential metrics for incident management (e.g., total numbers of incidents, number and percentage of major incidents, mean elapsed time to achieve incident resolution) and 10 metrics for problem management (e.g., the total number of problems/period, the percentage of problems resolved within service level agreement targets, the average cost of handling a problem, the number of major problems). Instead of long lists of metrics, IT service management frameworks should provide more practical examples how to use metrics. Additionally, metrics could be divided by priority into primary metrics and secondary metrics.

Besides ITIL, there are several other IT service management standards and frameworks that address the need for monitoring and measuring service management and provide their own set of metrics. Control Objectives for Information and related Technology (COBIT) framework is designed for IT governance purposes [5]. COBIT provides both process metrics and maturity level metrics for each delivery and support (DS) process, such as DS8 Manage Service Desk and Incidents and Manage Problems. Examples of metrics include first-line resolution rate, % of incidents reopened, % of problems recorded and tracked, and % of problems that recur (within a time period) by severity.

ISO 20 000 is ITIL-compliant auditable standard for IT service management that consists of two parts: specification for service management [4] and code of practice for service management [21]. One of its requirements is that the organization shall apply suitable methods for monitoring and measurement of the service management processes. ISO 20 000 also requires that the organizations produces reactive reports, proactive reports, and scheduled reports regarding IT service management activities.

However, ISO 20 000 does not tell which metrics should be used to measure the processes. It defines very generic requirements, for example, "the organization shall apply suitable methods for monitoring and, where applicable, measurement of the service management processes". According to ISO 20 000, service reporting should focus on performance against service level targets, non-compliance and issues (SLA and security breaches), workload (volume and resource utilisation), performance reporting on major events (major incidents and changes), trend information and satisfaction analysis [4].

IT organizations have difficulties in deciding what to measure. Even in a small IT organization there are hundreds of measurement targets to choose. Naturally, each measurement target requires a different type of measurement approach. Software quality metrics can be divided, for example, into efficiency metrics (transaction time), correctness metrics (complexity, MTBF), reliability metrics (down times), and maintainability metrics (number of modules, number of errors per unit) [22]. IEEE Standard Dictionary of Measures to Produce Reliable Software [23] divides metrics simply into process metrics and product metrics where product measures are applied to software objects and process measures are applied to the activities of software development, test and maintenance. Marik, Kral and Marik [24] have examined software validation and verification metrics from the testing viewpoint. Testing-related metrics can be used in the release testing activity of the IT service release management process [25].

We propose that there are three key issues that IT service support providers should measure. First, they should measure the performance of any IT service management process, such as resolution times and volumes for service incidents, problems and change requests categorized by customers, business priority [26], request type etc. A very basic software engineering metric that measures the quality of the software or service is the number of errors that relate to a configuration item.

Second, the IT service provider organization should measure the maturity of IT service management processes or an IT organization, for example, using the process maturity model of the Control Objectives for Information and related Technology (COBIT) [5], Capability Maturity Model Integration (CMMI) for Services [6] that is an extension to CMM model [27] or other maturity assessment models, such as self-assessment model of IT Service Management Forum [28], software maintenance maturity model [29] or corrective maintenance maturity model [30]. Third, it is very important to know the customer satisfaction rate on IT services and processes. Other metrics can be implemented after the three first metrics have been introduced, for example, service business performance metrics including the costs of service unavailability.

There seems to be a clear need for better IT service management measurement frameworks that would define the goals, roles, activities and would be easy to adopt in practice. Additionally, support process managers could use the practical measurement examples of some existing software quality measurement frameworks, such as the Defect Management Framework of Quality Assurance Institute [31], the Software Quality Measurement a Framework for Counting Problems and Defects [32] and Personal Software Process [33].

This case study is a part of the results of KISMET (Keys to IT Service Management and Effective Transition of Services) project [34] and MaISSI (Managing IT Services and Service Implementation) project [35] at the University of Eastern Finland, School of Computing, Software Engineering Research Group, Finland. The KISMET project focuses on improving IT service transition processes (change, configuration and release management) while the emphasis of MaISSI project work was in the service support (service desk, incident and problem management).

The work in our research project has been divided into eight subprojects (MaISSI pilot projects). Improving the IT service management measurement was one of the pilot projects and it was carried out during years 2008 - 2009. The measurement framework was created later in 2010 during KISMET project.

The main contribution of this paper is to:

- describe the implementation process of the ITIL-based IT service management measurement system (ITSM-MS),
- describe the system architecture and the main functions of the ITSM-MS,
- propose a framework for measuring IT services and
- present the lessons learnt from the implementation process.

## III. Research Questions & Methodology

The main research problem of this study is: how the measurement of the IT service support processes can be improved? Measurement and continous improvement are hot topics in the IT service provider companies at the moment. There is an urgent need both for a systematic IT service measurement process and for easy-to-use, dynamic measurement tools that enables effective and efficient performance reporting. This study focused on tool improvement providing valuable information on a unique case where a dynamic, real-time measurement tool was implemented for measuring IT service processes.

The measurement challenge was addressed by the application service manager of the case organization. Both constructive methods and case study methods [36] with action research features were used in this study. Constructive methods were used to build the measurement system and the measurement framework. A case study method is “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and the context are not clearly evident” [36]. Eisenhardt reports that a case study is “a research strategy which focuses on understanding the dynamics present with single settings” [37]. The main goal of the case study method was to analyze the current state of the customer support in the case organization. Action research methods [38] were used in the design and implementation meetings. Researchers were active participants in the implementation process.

### A. Case Organization and Data Collection Methods

Our case organization is a medium size business unit (around 120 employees) of a bigger organization that is one of the leading IT service companies in Northern Europe with over 16 000 employees. The company provides IT, R&D and consulting services for various industries, such as banking and insurance, energy, telecom and media, and healthcare. Our pilot was implemented together with the case organization and the MaISSI research team. Improvement of measurement activities was considered as a very important improvement target in the case organization and was selected to the main topic for the pilot project.

IT service support processes, such as incident management, problem management are part of the case organization’s business framework WayToExcellence (W2E). Service desk acts as a single point of contact for customers and users. The service desk is an extended version of the help desk. While the help desk focuses mainly on resolving incidents (software and hardware failures), the service desk provides a wider range of services. Besides resolving incidents by using various knowledge repositories, the service desk can handle service requests, license issues, change requests etc. The service desk acts as a Single Point of Contact (SPOC) for customers and users and records each contact to the incident database.

The service desk assigns the incident to the back office (second-line support) which in turn assigns the case to the product support team, if necessary. The back office is responsible for managing and resolving the service requests (for example, handling requests for database queries). The first level support and back office can escalate incidents to the 3rd-level teams (product development) if program code fixes are needed.

The organization uses a java-based tool for handling all the incidents. The incident management tool supports the ITIL-based service management processes. The organization has been quite satisfied with the tool functions and its configuration options. However, they stated that producing measurement reports regarding the performance of service support processes is not effective and a lot of manual work is related to producing those reports to customers. Every month 1-2 days were spent by service managers to generate service performance reports with MS Excel.

The pilot project between the MaISSI research team and the case organization was carried out between years 2008 and 2009. The main goal of the pilot project was to establish a system that enables better measurement of IT service support processes. The process included the following steps:

- 8 August 2008: The kickoff meeting of the pilot project.
- 1 September 2008: The 1st requirement specification meeting.
- 6 October 2008: The 2nd requirement specification meeting.
- 23 October 2008: The 3rd requirement specification meeting.
- October - November 2008: The design phase of the ITSM-MS.
- January - April 2009: The implementation phase of the system.

- 9 April 2009: The final review meeting of the system.
- April - May 2009: Trainings in the case organization.
- May 2009 - July 2009: Initialization of the system.

The case organization was selected from the pool of MaISSI research project's industrial partners. The research team had had cooperation with the case organization in earlier projects. The role of the MaISSI research team (a project manager, a research assistant) was to help the case organization in the design and the implementation of the measurement system. Figure 1 describes the context of this study.

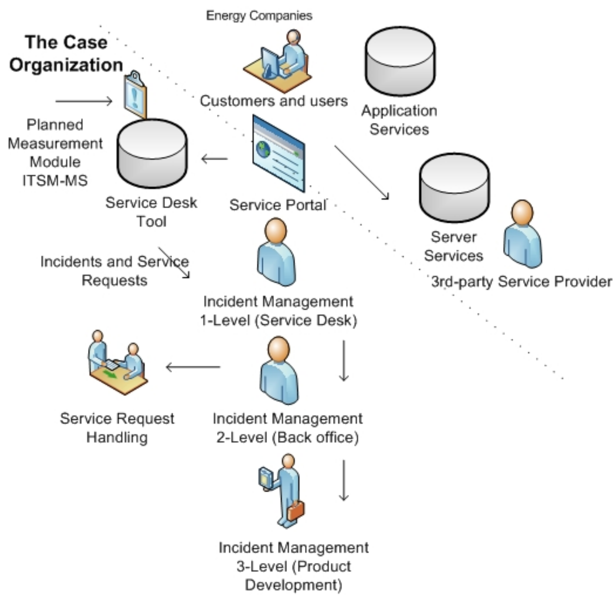



Figure 1. The case study context

The following data collection methods were used in the study:

- Participative observation (field visits and ITSM-MS work meetings)
- Interviews and discussions (application service manager, IT service manager, technical specialists during the implementation phase)
- Internal documentation (a service desk tool user guide, data on existing metrics)
- Access to the case organization, intranet and the service desk system

The MaISSI research team had access to the case organization's facilities as well as to the service desk system. The organization provided workstations to the research team.

### B. Data Analysis Method

In data analysis, we used a within-case analysis method that examines a case carefully as a stand-alone entity [37]. The case study database (a Windows folder with access to MaISSI team) was created to ensure the traceability between data sources, meetings and findings. The case study database included memos from meetings with a case organization, internal documents received from the case organization regarding the measurement of IT service support processes, and design and implementation documents created by the MaISSI research team.

## IV. Implementing an IT Service Measurement System

In this section, we will introduce how the ITSM-MS was established between the case organization and the MaISSI research team. The work was divided into six phases: kickoff, requirements specification, design, implementation, training and introduction and learning.

### A. Kickoff Phase

The kickoff meeting of the pilot project was arranged in 8th August 2008. In that meeting, the representatives (the application service manager and the IT service manager) of the case organization reported that they have problems in measurement and reporting activities regarding IT service support processes: Creating process performance reports with the current tools is mainly manual work and takes too much time from managers. They showed the research team one of the existing process performance reports: the relation of open incidents to closed incidents per day and stated that the organization needs more that type of reports that are easy to understand and provide important information for the business decision makers.

Additionally, the goals for the pilot project were specified. The main goal of the pilot project was to help the case organization in the development and introduction of IT service support metrics. The application service manager and the IT service manager stated that it would be nice to implement a couple of simple process metrics with modern web technologies. The task would mean in practice implementing a dynamic SQL query that combines many search parametres together such as customers, products, priority, and case type.

### B. Requirements Specification Phase

Few weeks after the kickoff meeting, the first requirement specification meeting for the ITSM-MS was arranged and the first requirements for the system were defined. Requirements were gathered together with the application service manager and the IT service manager of the case organization. The following general requirements were identified:

- The system must provide real-time measurement information about IT service support processes
- The system must be easy to use (produced measurement reports must be clear and easy to understand).
- The system must support ITIL-based IT service support processes, especially incident management and problem management.

- The system must provide a dynamic user interface (for example the user can select different values for producing graphs).
- The system must be implemented with the technology provided by the case organization.

Additionally, the MaISSI research team introduced some examples of IT service support metrics based on ITIL version 2, ITIL version 3, and an ITIL-based IT service management metrics book [39]. In the 2nd and the 3rd requirement specification meetings, the requirements for the ITSM-MS were clarified. The concepts in the requirements specification were based on the case organization's service management framework and the ITIL framework.

### C. Design Phase

The design phase of the system was carried out during October - November 2008 and it was divided into two stages. The first stage (October 2008) was performed as an action research where one researcher from the MaISSI project team was working intensively in the case organization. The researcher took part in the case organization's IT service management team and collected data from the incident management tool.

The second stage (November 2008) aimed at preparation of the system specification based on the data that was collected earlier. For example, the following measurement targets were identified:

- Throughput times
  - by products
  - by customers
  - by request types
  - by urgency
- Request volumes
  - by products
  - by customers
  - by priority level

Additionally, the MaISSI project team defined the process metrics that could be created with the new measurement system. Metrics were defined based on ITIL service support metrics. Draft versions of metrics were designed with MS Excel.

- Number of new and closed incidents by priority level
- Number of new and closed problems by products
- Number of new change requests by a customer and a product
- Number of all service desk cases
- Incident throughput time by priority level
- Average incident resolution time

During the second stage, a drop-down menus for the user interface of the ITSM-MS were designed by identifying search parametres and their values (see Figure 2).

In this study, the ITSM-MS was constructed for the internal use of the case organization and its employees. The system was targeted to process managers, product managers and project managers to work as a real-time measurement tool that is connected to every IT service support process.

| Incident Creator | Priority | Customer |
|---|---|---|
| All | All | All |
| Customer | 1 - Urgent | Specified |
| Case Organization | 2 - High | |
| | 3 - Medium | |
| | 4 - Low | |

Figure 2. Design view of the parametres

Figure 3 shows the general system architecture of the ITSM-MS. The service desk (SD) of the case organization receives an incident from the customer, a ticket is entered into the incident management tool and a new case is opened.

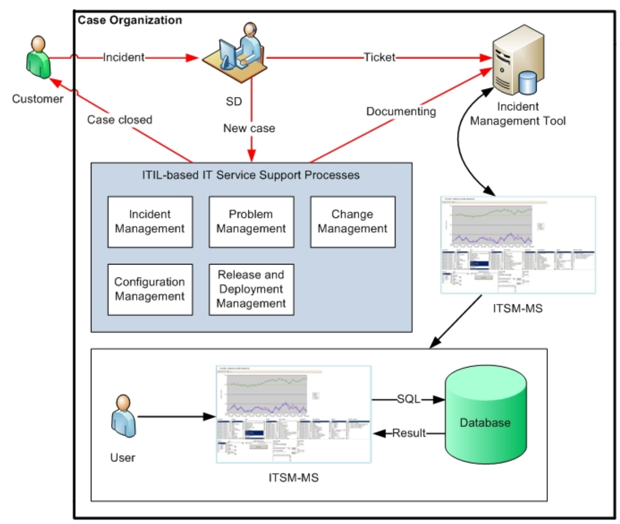



Figure 3. The system architecture

The case organization uses ITIL-based IT service support processes for resolving the case and all the data that is used during these processes is documented into the incident management tool. When the case is resolved, it is sent back to the customer and the case is closed.

As a result of the design phase, the system specification was documented and reviewed at 24th November 2008.

### D. Implementation Phase

The implementation phase of the ITSM-MS was executed in January - April 2009. This phase was also performed as an action research where a member of the MaISSI research project team worked in the case organization's facilities. Based on the system specification, the ITSM-MS was implemented with the help of the case organization and its employees.

From the technical point of view, the ITSM-MS was implemented using Microsoft .NET environment, Microsoft Visual Studio, C# programming language and Oracle database. The ITSM-MS uses the database of the incident management tool. Microsoft Visual Studio was used to implement the user interface and the system functions while Oracle database stores all the data from the IT service support processes.

The ITSM-MS uses the data that is documented in the incident management tool. The user can use the system from the case organization's Intranet and make real-time graphs about different cases. The user's input is transformed into an SQL-query which returns a result from the database. Based on this result, the ITSM-MS draws a graph of the metric into the user's computer screen in the Intranet (see Figure 4).

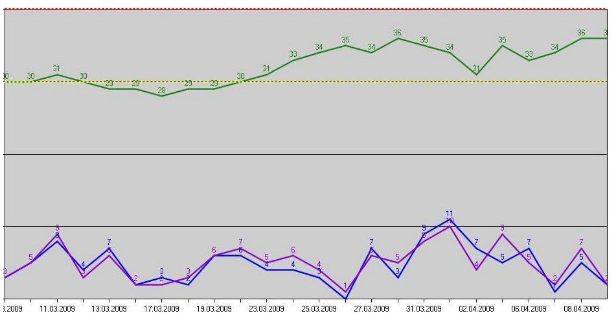


Figure 4. The process performance graph.

The user interface of the ITSM-MS contains different parametres and functions that the user must select for generating the graph and saving the user profile (see Figure 5).

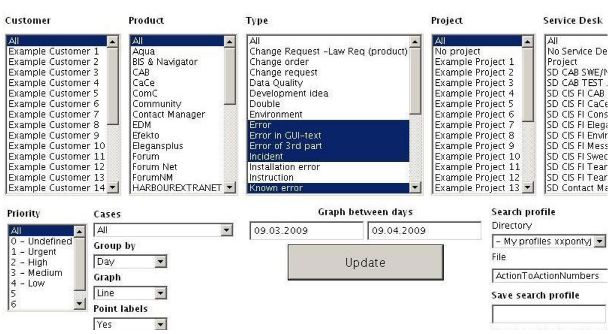



Figure 5. Setting parametres in the ITSM-MS.

The user profile is later used for generating the graph based on saved values and functions. In that way, the user does not have to enter all the values again for the graph. The user can select a specified value or multiple values depending on what type of graph is needed to represent.

- Customer: All customers of the case organization.
- Product: All products of the case organization.
- Type: Different types of cases (for example incident, problem, change request or known error).
- Project: All projects of the case organization
- Service Desk: SD of the case organization.
- Assigned to: The person who is responsible of the case.
- Classification: All cases are classified (for example error in program, hardware problem or error in documentation).
- Impact: The impact of the case (for example standard impact, major impact or note).
- Business impact: How case affects the service or the business (for example no business impact and minor error or esthetic).
- Priority: Priorities of cases: 0 - undefined, 1 - urgent, 2 - high, 3 - medium and 4 - low.
- Cases: Some particular cases.
- Group by: Days, weeks, years.
- Graph: The graph of the metric can be presented as a line graph or a bar graph.
- Point labels:Yes or no.
- Graph between days: The starting point and the ending point for the graph.
- Search profile: Search for a particular saved user profile.
- Save search profile: Save the current user profile.
- Target limits: This is used for the traffic lights (explained later in this paper).

The final review meeting of the system was arranged in 9th April 2009. Participants at that meeting were the application service manager and the IT service manager of the case organization and the MaISSI research project team. After this, the system was deployed to the operational environment.

### E. Training and Introduction Phase

Trainings for the employees were carried out during April - May 2009 by the application service manager of the case organization. The user of the ITSM-MS can select target limits for the metric and save them to a user profile. Limits are used to show the current situation of different cases by traffic lights that are showed in Figure 6.

The pointer on the green section means that the situation is fine and there are not so many unsolved cases. The pointer on the yellow section means that the situation needs more attention and there are few unsolved or non-closed cases to resolve.

Finally, the pointer on the red section means that there are too many unsolved cases in the incident database. This means that the case organization must take action for resolving those cases and turn the pointer from the red section back to the green section. Numbers below the traffic lights describe cases that are currently open and the change on previous measuring period, for example last month. Numbers above the traffic lights are the target limits.

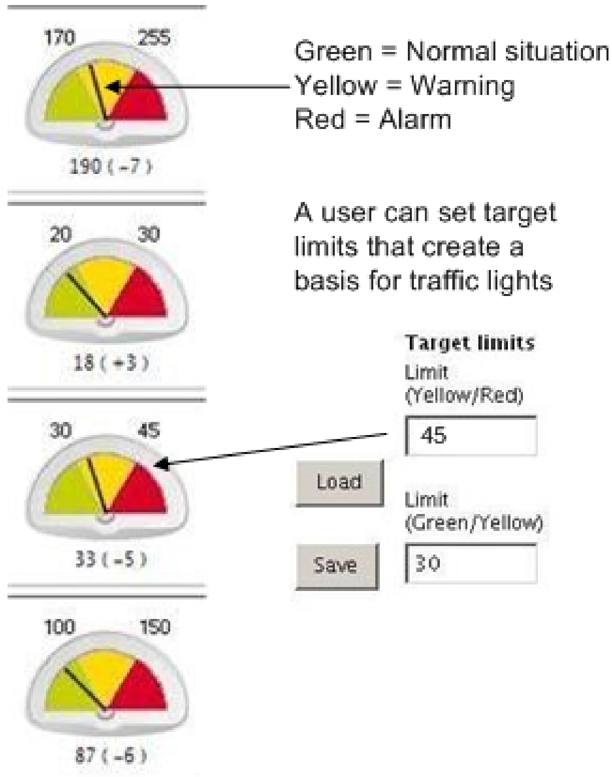



Figure 6. The traffic lights report

The ITSM-MS can also be used as a miniature version besides of the full version that was showed earlier in this paper. Figure 7 presents the mini window of the ITSM-MS system where a saved user profile is showed on the screen. The miniature version enables users to create measurement reports from a company's intranet page.

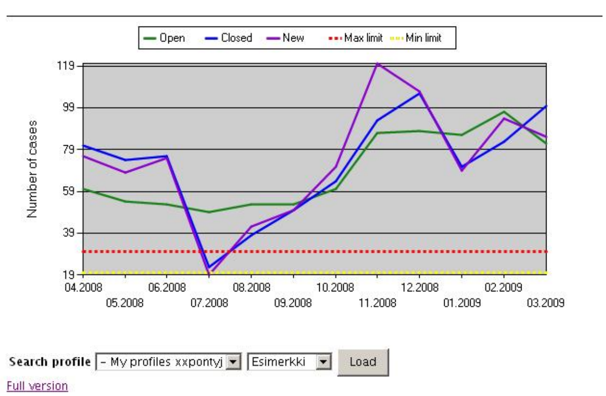



Figure 7. The mini window of the ITSM-MS.

The mini window can show all the saved user profiles that the user has created. Thus, the user does not have to open the full version of the ITSM-MS, if the required graph is saved on a profile. The mini window enables the faster use of the ITSM-MS. After trainings, the initialization phase of the system was performed and now the ITSM-MS is currently in use in the case organization.

*F. Learning Phase*

During the design and implementation of the IT service management measurement system, many things were learned. First, besides the well-designed and easy-to-use measurement tool, an IT service provider organization must have a systematic measurement process that defines why, how, when, to whom metrics and reports are generated.

Second, metrics should be based on the business objectives. The linkage between metrics and business objectives can be built by using critical success factors, key performance indicators and metrics. Thus, we can measure the things that are related to business strategies. Third, IT service providers should invest in how to use the collected measurement data. The data can be used to identify trends and deviations/exceptions in service quality. Information on trends can be used as an input for other service management processes, such as problem management (proactive problem management) and continuous service improvement.

It is hard to calculate what was the exact cost of building the ITSM -MS system. The work effort consisted of a design phase (1 month) carried out by a university researcher, an implementation phase (two months) by a computer science student, and supervision hours given by a project manager.

As a part of the learning phase, we created a process framework for IT service management measurement (see Figure 8) based on the 7-step improvement model of the ITIL Continual Service Improvement book.

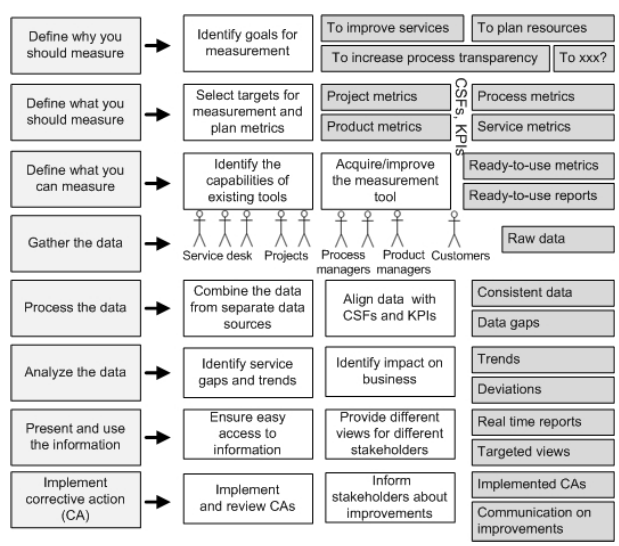



Figure 8. The measurement framework for IT service management.

Our measurement model is based on the following principles:

- Metrics should be linked to business objectives by using three elements (example given in parentheses):
  - Critical Success Factor (Quickly resolve incidents)
  - Key Performance Indicator (Percentage reduction in average time to respond a service desk call)
  - Metric (Call response time in seconds)
- Each phase of the measurement process should have clearly defined outputs. Outputs are marked with dark grey colour in our model.
- The measurement system should enable rapid decision making for business managers. Instead of complex reports one should use "traffic light" reporting or set maximum/minimum limits to charts.
- The measurement system should enable real time reporting instead of pdf reports. There is always a gap between time when a report is created and time when a report is read.
- Targeted measurement reports should be provided to stakeholders. Because stakeholders' needs for reporting vary a lot, one should provide them a possibility to create reports by themselves.

This study was focused on the implementation of the measurement system. The above presented measurement framework has not yet been validated with the case organization but it is presented here as an outcome of the learning process.

## V. Analysis

The IT service management measurement system created in the MaISSI project provides several benefits for the case organization's IT service support. We analyzed the benefits by comparing the state of measurement in the case organization after the ITSM-MS vs. situation before ITSM-MS.

- The work effort required to produce performance reports: 15-20 minutes vs. 1-2 days per person
- The format of reports: dynamic real-time reports vs. fixed pdf reports
- Usability of the report tool: good, in most cases does not require additional training vs. difficult to use in many points.
- Sharing and use of measurement reports: dynamic reports accessible from websites vs. reports must be created from a service desk tool.

First, it remarkably decreased the amount of manual work in measurement. Instead of 1-2 days, the service performance reporting takes now 15-20 minutes. No Excel-based reporting is needed anymore. Second, it provides real-time reports about the process performance enabling faster business decision making regarding service support issues. Third, the usability of report/chart generator is better in the new system than in the existing service desk system. For example, the system can show the id numbers of service desk cases that have been open for a long time. Thus, process manager is able to click the id number and the system shall show the details of the service desk case that requires more attention.

Finally, the system enables an effective knowledge sharing of measurement data to appropriate stakeholders. Measurement queries can be installed into organization's intranet pages and charts can be displayed even as a personal screensaver. The case organization is also planning to provide the reports from the ITSM-MS to customers in near future. Regarding the limitations of our system, the ISM-MS is an add-on module and cannot work without a database of a service desk tool. Additionally, there is no installation software for ITSM-MS available. Thus, the system might be difficult to transfer to new environment.

The following lessons learnt were identified during the implementation of the IT service management measurement system. The implementation phase that the lesson is based on is coded as follows: K=Kick off; R=Requirements specification; D=Design; I=Implementation; T=Training and introduction. The coding of lessons learnt helped us to maintain a chain of evidence between data sources and results.

- The incident management process is a good area to start the IT service management measurement activities (K). If the organization has a help desk or a service desk, it should likely have already several metrics that provide easy start for the improvement of measurement activities. In case of other support processes it might be that there are no existing metrics.
- Create a sense of urgency for the implementation of the measurement system (K) if people need motivating for measurement . In our case, the level of urgency was quite low. However, the improvement of reporting and measurement methods were considered actual internal development target. Perhaps more urgency for the measurement system would have given if the customers had been unsatisfied with the process performance reports.
- Time-based metrics are more difficult to implement than volume-based metrics (R,D). Calculating the resolution time was harder than expected. It should take account in the holidays, weekends and other times when support engineers are not resolving cases.
- Keep measurement reports as simple as possible (T). Although we started from really simple metrics, we found that it was difficult to interpret graphs that somebody else had done with the ITSM-MS.
- Even the top-quality measurement system is dependent on the quality of the collected data (D,I,T). We noticed that our system showed in the drop-down menu all the request types (around fifteen) that were created in the service desk tool. If teams do not have unified classification rules, measurement reports do not give reliable results.

- Selecting appropriate metrics for IT service support is difficult (R). IT service management frameworks have introduced dozens of metrics for one process. However, the frameworks do not tell which metrics are obligatory and which are more like nice-to-have requirements. We recommend Top3 approach for defining metrics. In the Top3 approach, IT organization selects, describes and implements three most important metrics for each IT service support process. Other metrics are considered as nice-to-have metrics.
- The IT organization needs a systematic measurement approach in addition to the measurement system (K,R). However, we observed in this study that the IT service management frameworks seem to lack a clear measurement and reporting process. Our initial goal was to define the measurement process for the case organization before the measurement tool but we failed to achieve this goal.
- Managers love traffic lights (I,T). Traffic lights were perhaps one of the best functions of the system. Simple colour-coded function provides business managers or process managers a rapid overview what is happening in the support and maintenance and where are the pain areas.
- Focus in the early phase of the measurement project how to deploy and use the collected measurement data instead of solely thinking how to create reports (T). In our system, the user does not have to open the full version of the ITSM-MS, if the required graph is saved on a user profile. This enables easy access to the measurement data and likely increases the system usage.
- Implementation of a measurement system does not require a large development team (I). In our project, the implementation of the ITSM-MS was carried out by one person that was a very good programmer. The system design document was created by a MaISSI researcher.

Both the research team and the case organization considered the pilot project and its main result, IT service management measurement system, as a success. After the implementation of the measurement system the cooperation between MaISSI and the case organization continued as another pilot project the goals of which was to increase the transparency of the organization's service support processes to customers and to improve the release management process. The ITSM-MS can be used to provide measurement information regarding all the support processes: incident management, problem management, change management, release management and configuration management.

## VI. Conclusion

This study aimed to answer the following research problem: How the measurement of the IT service support processes can be improved? The main contribution of this study was to 1) describe the implementation process of the ITIL-based IT service management measurement system (ITSM-MS), 2) describe the system architecture and the main functions of the ITSM-MS, 3) propose a framework for measuring IT services and 4) present the lessons learnt from the implementation process.

In this paper, we described the four phases of the implementation process: requirements specification, design, implementation, training and introduction. During these phases we identified the functional and data requirements for the measurement system, described the high-level system architecture, and explained how the system works in the practice. Additionally, we presented ten most important lessons learnt from the implementation process.

Data for this study were collected using case study methods and action research methods. Additionally, constructive methods were used in designing and implementing the prototype of the ITSM-MS. IT service management measurement system was implemented together with a business unit of the large IT service provider organization in Finland. We have received very positive feedback from the case organization regarding the system. The system is considered very useful in the case organization and it has remarkably decreased the amount of manual work in creating process performance reports.

There are several limitations to this study. First, data were collected from one IT service provider company within a relatively short time period. The customers of the case organization did not participate in this study because the improvement focus was on the internal perspective. We also used three important principles of data collection [36] to increase the quality of our study 1) use multiple sources of evidence, 2) create a case study database, and 3) maintain a chain of evidence between data sources and results.

Second, we did not focus much on other non-functional requirements than usability and the fact that the system must work real-time. Third, because the case organization was selected from the partner pool of the MaISSI project, the selection method was the convenience sampling method. Finally, the case study does not allow us to generalize our research results. However, we can use our results to expand the theory of IT service management measurement.

Further research is needed to examine establishment of service support measurement systems in IT service companies and implementation of a systematic measurement process.

## Acknowledgment

This paper is based on research in KISMET (Keys to IT Service Management and Effective Transition of Services) project and MaISSI (Managing IT Services and Service Implementation) project funded by the National Technology

Agency TEKES, European Regional Development Fund (ERDF), and industrial partners.

## REFERENCES

[1] A. Lahtela, M. Jäntti, and J. Kaukola, "Implementing an itil-based it service management measurement system," in *Proceedings of the 4th International Conference on Digital Society*. St. Maarten, Netherlands Antilles: IEEE Computer Society, February 2010, pp. 249–254.

[2] IEEE Standard 1061-1998, *IEEE Standard for a Software Quality Metrics Methodology*. IEEE, 1998.

[3] Office of Government Commerce, *ITIL Service Lifecycle - Introduction*. The Stationary Office, UK, 2007.

[4] ISOIEC20000, *IT Service Management, Part 1: Specification for service management*. ISO/IEC JTC1/SC7 Secretariat, 2005.

[5] COBIT 4.1, *Control Objectives for Information and related Technology: COBIT 4.1*. IT Governance Institute, 2007.

[6] F. Niessinka, V. Clerca, T. Tijdinka, and H. van Vliet, "The it service capability maturity model version 1.0," CIBIT Consultants&Vrije Universiteit, 2005.

[7] Microsoft, "Microsoft operations framework," http://technet.microsoft.com/en-us/library/cc506049.aspx, July 2010.

[8] J. Caldeira and F. B. e Abreu, "Influential factors on incident management: Lessons learned from a large sample of products in operation," in *Product-Focused Software Process Improvement*, A. Jedlitschka and O. Salo, Eds., vol. 5089. Springer Verlag, 6 2008, pp. 330–344.

[9] M. Kajko-Mattsson, C. Ahnlund, and E. Lundberg, "Cm3: Service level agreement," in *ICSM '04: Proceedings of the 20th IEEE International Conference on Software Maintenance*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 432–436.

[10] OGC, *ITIL Service Design*. The Stationary Office, UK, 2007.

[11] Office of Government Commerce, *ITIL Continual Service Improvement*. The Stationary Office, UK, 2007.

[12] R. Morgan and L. Ho, "Six sigma for it service management," Six Sigma Zone Report, 2004.

[13] M. Jäntti, A. Miettinen, and K. Vähäkainu, "A checklist for evaluating the software problem management model: a case study," in *SE'07: Proceedings of the 25th conference on IASTED International Multi-Conference*. Anaheim, CA, USA: ACTA Press, 2007, pp. 7–12.

[14] OGC, *ITIL Service Support*. The Stationary Office, UK, 2002.

[15] M. Kajko-Mattsson, "Maturity status within front-end support organisations," in *ICSE '07: Proceedings of the 29th international conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 652–663.

[16] Office of Government Commerce, *ITIL Service Operation*. The Stationary Office, UK, 2007.

[17] K. H. Bennett and V. T. Rajlich, "Software maintenance and evolution: a roadmap," in *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: ACM Press, 2000, pp. 73–87.

[18] B. P. Lientz and E. B. Swanson, *Software Maintenance Management*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1980.

[19] F. Niessink and H. van Vliet, "Software maintenance from a service perspective," *Journal of Software Maintenance*, vol. 12, no. 2, pp. 103–120, March/April 2000.

[20] M. Jäntti, "Difficulties in managing software problems and defects," Ph.D. dissertation, University of Kuopio, 2008.

[21] ISOIEC20000b, *IT Service Management, Part 2: Code of practice for service management*. ISO/IEC JTC1/SC7 Secretariat, 2005.

[22] E. Wallmueller, *Software quality assurance: A practical approach*. Prentice Hall International, 1994.

[23] IEEE Standard 982.1-1988, *IEEE Standard Dictionary of Measures to Produce Reliable Software*. IEEE, 1988.

[24] V. Marik, L. Kral, and R. Marik, "Software testing & diagnostics: Theory & practice," in *SOFSEM '00: Proceedings of the 27th Conference on Current Trends in Theory and Practice of Informatics*. London, UK: Springer-Verlag, 2000, pp. 88–114.

[25] Office of Government Commerce, *ITIL Service Transition*. The Stationary Office, UK, 2007.

[26] C. Bartolini and M. Salle, "Business driven prioritization of service incidents," in *Utility Computing*. Springer Berlin / Heidelberg, 2004, vol. 3278, pp. 64–75.

[27] P. Jalote, *CMM in Practice, Processes for Executing Software Projects at Infosys*. Addison-Wesley, 2000.

[28] IT Service Management Forum, "Itil service management self assessment," http://www.itsmfi.org/content/self-assessment-itil-v2, 7 2010.

[29] A. April, J. H. Hayes, A. Abran, and R. Dumke, "Software maintenance maturity model (smmm): the software maintenance process model: Research articles," *J. Softw. Maint. Evol.*, vol. 17, no. 3, pp. 197–223, 2005.

[30] M. Kajko-Mattsson, "Corrective maintenance maturity model: Problem management," in *ICSM '02: Proceedings of the International Conference on Software Maintenance (ICSM'02)*. Washington, DC, USA: IEEE Computer Society, 2002, p. 486.

[31] Quality Assurance Institute, "A software defect management process," Research Report number 8, 1995.

[32] W. Florac, "Software quality measurement a framework for counting problems and defects," Technical Report CMU/SEI-92-TR-22, 1992.

[33] I. Hirmanpour and J. Schofield, "Defect management through the personal software process," *Crosstalk, The Journal of Defense Software Engineering*, 2003.

[34] KISMET, "Kismet website," http://www.uef.fi/KISMET/, access: 18.1.2011.

[35] MaISSI, "Maissi website," http://www.uef.fi/maissi/, access: 18.1.2011.

[36] R. Yin, *Case Study Research: Design and Methods*. Beverly Hills, CA: Sage Publishing, 1994.

[37] K. Eisenhardt, "Building theories from case study research," *Academy of Management Review*, vol. 14, pp. 532–550, 1989.

[38] I. Benbasat, D. K. Goldstein, and M. Mead, "The case research strategy in studies of information systems," *MIS Q.*, vol. 11, no. 3, pp. 369–386, 1987.

[39] IT Service Management Forum, *Metrics for IT Service Management*. Van Haren Publishing, 2006.

# Statistical Convergence Investigation of Routing Protocols



Frank Bohdanowicz, Marcel Jakobs, Christoph Steigner
*University of Koblenz*
*Germany*
{bohdan,zimon,steigner}@uni-koblenz.de

***Abstract*—The performance of routing algorithms can only be compared if they undergo a comprehensive convergence analysis. In this paper, we present a new approach for convergence analysis in order to evaluate the benefit of a new distance vector routing algorithm, which is no longer affected by the well-known Counting-to-Infinity (CTI) problem which still occurs in topology loops. The newly developed *Routing with Metric-based Topology Investigation (RMTI)* protocol only uses event-triggered updates. Thus, the convergence time and the update traffic can be reduced. Convergence properties of RMTI are compared with the *Routing Information Protocol (RIPv2)* under the impeded condition of provoked CTIs. In this way, the performance benefit of RMTI can be shown in comparison to RIPv2. The major focus of this paper are the approaches to measure the convergence time of routing protocols in a newly developed test environment. Special effort is directed at minimizing measurement perturbations by the separation of the online data capturing task from the offline data evaluation task. The results show that this convergence measurement method is of universal quality. RMTI is a newly competitive intra-domain routing protocol which can perform filtering policies in contrast to other intra-domain routing protocols.**

***Keywords*- distance vector routing; metric-based topology investigation; routing loops; counting to infinity problem; routing evaluation**

## I. Introduction

Routing is the exchange of routing information between routers to provide them with reachability information about destination nodes in the computer network. Decentrally running routing protocols can repair interrupted links if the network topology contains enough loops. In case of a link failure, loops provide alternative links to the same destination node in a network.

However, loops in the topology are also the main reason for the occurrence of routing loops. A routing loop is a circular trace of a routing update message which returns to the same router, either directly from the neighbor router or via a loop topology. It is crucial to detect and to prevent routing loops since they can consume a large amount of network bandwidth and impact the end-to-end performance of the network.

All common routing protocols cope more or less efficiently with the routing loop problem as a routing system is a distributed system. Therefore, new routing protocols should deal with topology loops as well as routing loops in an efficient way in order to provide convergence and stability for the network.

The newly developed *Routing with Metric-based Topology Investigation (RMTI)* protocol [2] can detect routing loops and omit the Counting-to-Infinity (CTI) problem. The well-known CTI-problem stands for the characteristic routing-loop situation in RIP-networks. In a CTI situation the routing metric is constantly increased due to circulating routing update messages in a routing loop. RMTI accelerates the convergence time and reduces the update traffic by only using event-triggered updates together with neighbor-alive-notifications. In contrast to this new approach the common RIP protocol is mainly based on periodic update messages.

To compare the convergence time of routing protocols, a new test environment has been developed. This test environment consists of the following components (see Figure 1):

- A computer network based on virtual linux machines connected by software bridges.
- An online data capturing facility to collect characteristic data during a test run.
- A script-based event generator capable of triggering failure events.
- A topology generator able to generate many different but regularly structured topologies (see Figure 4) and random topologies with pre-defined constraints.
- An offline statistical analysis tool to visualize the results of a test run using plots and graphs.

Network failures which cause CTIs can be triggered, recorded and evaluated with the test environment. The update traffic together with a time stamp is recorded during the online data capturing phase. In the offline data analysis phase, the events, starting with the first network failure up to the end of the convergence process, are analyzed.

A major concern of this research is to include and compare the CTI-problem of common RIP algorithms into the convergence analysis. A convergent state in a distributed routing system is given when all forwarding tables in the routers contain the optimal next hop entries for the given network topology. The convergent state is not given if the forwarding tables do not offer the optimal next hop entries due to a certain network failure.

This paper is structured as follows: In Section 2, other approaches of convergence tests and routing protcols are reflected. In Section 3, the newly developed test environment is introduced. In Section 4, the tested routing protocol is presented. In Section 5, an overview about the routing algorithms tested so far is shown. In Section 6, the tested topologies and the convergence measurement technique is described. Furthermore the results of the convergence analysis are discussed in this section. The paper ends with an outlook of further research items.

## II. Related Work

Newly developed routing protocols or enhancements to existing routing protocols are usually tested in simulation test environments. They only analyze a certain part of the respective protocol in order to reveal its features. The authors of [11] and [26] analyze their protocol enhancements by simulation techniques. The test environment described in this paper is not based on simulation, but on emulation technique. It can analyze routing software which is fully implemented and ready to be used, e.g., in company networks.

The test environment is based on the *Virtual Network User Mode Linux (VNUML)* system [20]. A system especially developed for protocol testing in a virtual machine infrastructure. Virtual machines are generated with *User Mode Linux (UML)*[6][7] and are connected together by software bridges [25]. In order to generate many different network topologies and link failure situations, the VNUML system has to be extended by an automatic script software. This script software can also collect a great deal of data within log files.

Currently, there is hardly any research on convergence analysis in deployed networks. Therefore this newly approach is presented in the next section. Several enhanced distance vector protocols have been proposed in order to avoid routing loops as well as the CTI problem. In contrast to RMTI, many of these approaches increase the amount of information exchanged among the nodes of a network.

The Ad hoc On-demand Distance Vector (AODV) protocol [16] by Perkins extends the distance vector information originally based on subnet N, next hop NH and distance D, to a 4-tuple (N,NH,D,SEQ), where SEQ denotes the sequence number. This approach is loop free [15] but it is not compatible with RIP due to the required protocol changes.

The Enhanced Interior Gateway Routing Protocol (EIGRP) used by Cisco is based on the DUAL algorithm proposed by Garcia-Luna-Aceves. DUAL provides loop-free paths at every moment, which was proven in [10]. EIGRP is a Cisco proprietary routing protocol and not compatible with RIP because of a completely different protocol design.

A solution called Source Tracing has been introduced by Cheng *et al* [5] and Faimann [9]. This approach provides additional information in updates and routing tables by adding a first-hop indication to the path. In this way loops can be recognized recursively.

These protocols avoid the CTI problem because they provide loop freedom. Likewise, they are not compatible with the RIPv2 standard or are proprietary approaches.

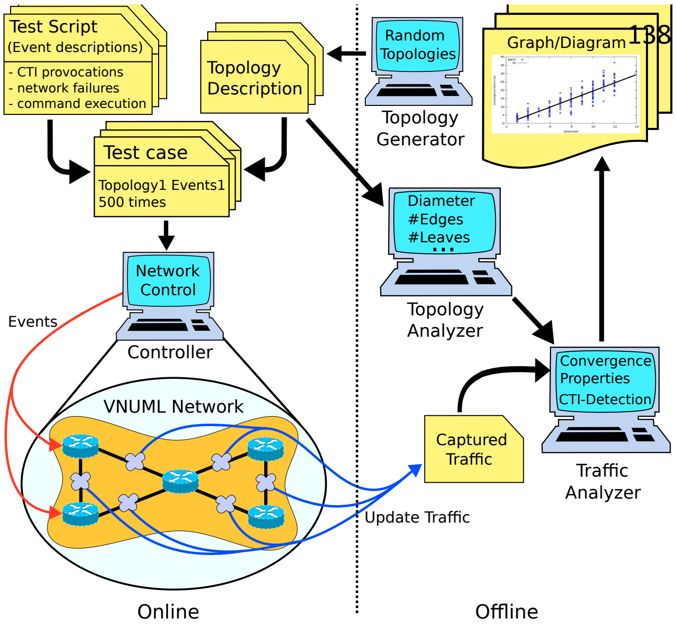



Fig. 1. Test Environment

In contrast to these approaches, we aim to provide a solution that has a complete and sound backward compatibility with every existing implementation of RIPv2 [12]. The enhanced knowledge is based on information already provided by the RIP protocol. Therefore, a new RMTI router can be deployed at selected nodes in an existing RIPv2-domain.

## III. Test Environment

Various tests with different topologies have been carried out in order to investigate the convergence properties of different routing protocols. For an automatic execution of these test runs a script software was written to perform the following operations which

- starts many different VNUML network scenarios in succession
- uses a test script of certain network failure situations (e.g., link corruption, router collapse)
- controls each VNUML scenario during runtime
- causes runtime failures in a predicted manner
- collects data needed for the offline evaluation
- evaluates the captured data during an offline phase with respect to convergence time and network traffic.

The measurements are carried out by the analysis of automatically generated tcpdump files [21] after the test run. Specific network data are logged and analyzed during each test run. This approach prevents measurement pertubations as only specific data are captured during runtime.

The separation of the test script file and the topology discription file (see Figure 1) allows for flexible design of test runs. The test script files define the topologies and the frequency of test runs. Each test run is divided in an *online* and *offline* phase (see Figure 1) in order to distort the runs as less as possible by the data capturing. This test environment is needed to provide measurement results that are as close to reality as possible in order to analyze convergence properties of routing algorithms.

The tested routing protocol implementations are part of the widely used Quagga Routing Software Suite [17]. The RMTI algorithm is implemented as a re-engineered version of the Quagga RIPv2 router. The test environment can carry out up to 1000 and more test runs where each run can be fully controlled and analyzed.

During a test run all characteristic actions and changes are saved with a corresponding timestamp for later synchronization. When the online test phase is finished the captured data are evaluated by a statistic analysis in the offline phase.

The test environment is developed for evaluating complex and comprehensive test runs without any human interaction. The environment can be configurated for many different test runs which cover various network failure situations. Each step of a test run can be pre-determined precisely by the test script. Many test cases including random or previously defined link and router failures can be performed.

To be able to test various failure scenarios that could occur in networks, the test environment allows for the opportunity to cause packet losses or transmission delays for single or multiple devices of one or more routers. In order to expose the distance vector routing algorithms to Counting-to-Infinity situations, the test environment can cause these situations by retaining certain update messages on certain router ports.

The high degree of automation of the test environment enables to gather a great deal of result data that can be statistically analyzed. Together with the network environment for the test cases, very detailed insight into the behavior of common routing algorithms can be obtained. The results of the test runs can be summarized and then visualized as diagrams. With this approach, the relationship of the various test parameters as well as a performance comparison between the different routing algorithms can be shown. The evaluation of these results enables the user to depict these relationships as mathematical functions. These functions are approximated using regression analysis. Additionally, the representation of frequency distributions such as in Figure 13 is supported. The test environment can average the results before analyzing them with statistical methods. It can create a network graph of the network topology even if the topology is randomly generated.

Before the test starts, topology properties of the network like diameter, articulations, leaves and number of nodes, edges and circles [8] are analyzed offline and saved in the topology description (see Figure 1). During the runtime of the test, all characteristic events are saved as routing update packets with a timestamp for the offline analysis after the run. After the runtime the convergence properties, such as convergence time or traffic amount, are calculated with the saved update traffic and information about the topology tested.

As the test environment is modularly designed it can be used for the comparison and testing of other intra domain routing algorithms as well. For this purpose, only the specifications for the offline analysis module have to be modified. This can be done by identifying the update packet which leads to the convergence of the network.

## A. Detailed description of test and measurement procedure

To measure the convergence time of a certain network, the network has to be changed from a convergent state to an in-convergent state by changing the status of certain links or routers in order to cause a link or router failure.

In order to calculate convergence properties of the network after a test run, the protocol analyzer *tcpdump* [21] is used to capture the existing network traffic. To get proper results the routing daemons in the routers have to reach a convergent state before the status of certain links or routers is changed to cause a failure. Then the sequence of state changes is recorded starting with the failure state and ending with the next convergent state. When the network has reached its convergent state, the run is automatically stopped and the next run follows.

After the online capturing of the router states is completed, the offline evaluation of the states can be executed. To evaluate the measurement, the captured update traffic has to be treated in the same way as the routing algorithms do in order to calculate the forwarding table entries. That means that the offline evaluation module repeats the processing steps of the routing algorithm.

The timestamped update packets of all subnets are sorted chronologically for evaluation. To identify the starting point of the convergence time $t_{conv}$, the first packet with the information about an unreachable subnet is selected and its timestamp saved for later use (as first timestamp $t_f$). As well, the timestamp of the first update packet can be taken as first timestamp to analyze the coldstart[1] behaviors of a routing algorithm.

In order to find the timestamp of the update packet leading to convergence (the last timestamp $t_l$), all update packets are re-processed from the start.

Whenever a routing table is updated, the timestamp of the appropriate update packet is saved. As the network must be in a convergent state when no routing table is updated any more, the last timestamp $t_l$ must belong to the update packet that led to this state. This timestamp is chosen for convergence time calculation.

The convergence time $t_{conv}$ is now defined as:

$$t_{conv} = t_l - t_f \text{ with:} \quad (1)$$

$t_{conv}$ = convergence time,
$t_l$ = last timestamp,
$t_f$ = first timestamp

Adding up the packet sizes of the whole traffic sent between the first timestamp $t_f$ and the last timestamp $t_l$ shows the traffic volume produced by routing updates during the convergence procedure.

[1] all routers in the considered network were rebooted and start with empty routing tables

## IV. IP NETWORKS AND ROUTING PROTOCOLS

Within computer networks, routers are special nodes which provide the whole network with information about the location of subnets. The IP protocol uses packet forwarding to deliver data packets from the source to the destination. Packet forwarding means that a node only knows the adjacent node, which is closer to the destination node. Therefore, every router has to know which adjacent router is closer to the subnet of the destination node. A router maintains a forwarding table listing the subnets in connection with the corresponding next hop router and the metric which represents a distance to the destination node. Routing is the process of completing and maintaining an accurate forwarding table. It is a distributed network-wide process which offers some properties such as dynamic adaptation to network changes, scalability and system stability. At any time, a forwarding table of a router must correctly describe the location of subnets and how to get there in order to prevent misguided data packets. There is always a small time gap beginning with a network failure and ending with the detection of the failure by the routers. Keeping the network free of invalid routing information during this time is a challenge every routing protocol has to cope with. It may occur that routing updates between routers become invalid on their way through the network and forwarding loops can take place. In practice, all common routing protocols are affected by forwarding loops [24]. A forwarding loop corresponds to a loop in the forwarding process of data packets. Thus, a routing loop corresponds to a loop in the routing process caused by circulating routing update packets. A routing loop is in close connection with a forwarding loop because the routing process has a direct impact on the forwarding process.

### A. Relevant Items of the RIP Protocol

The Routing Information Protocol (RIP) is considered as the classical representative of the distance vector protocol family. It employs the hop count as a metric to a destination node in the network (see Figure 2).

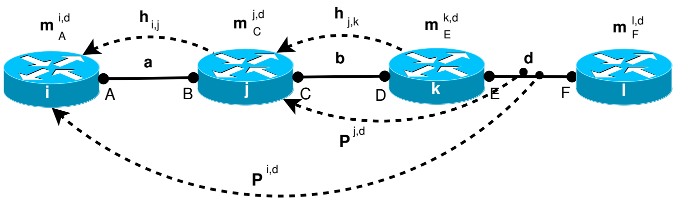



Fig. 2. Network model, $h_{i,j}$ = hop from router i to router j, $P_{i,d}$ = path from router i to subnet d with metric $m_{i,d}$.

RIP works like this: If a RIP router $i$ receives a routing update from an adjacent RIP router $j$ to a subnet $d$ with the metric $m^{j,d}$ via interface A of router $i$, this indicates the existence of a corresponding path $P^{j,d}$ from RIP router $j$ to subnet $d$. Router $i$ does not know the complete path $P^{j,d}$, but knows the number of hops of this path. Then, from the view of router $i$, there will be a path $P_A^{i,d}$ with metric $m_A^{i,d} = m^{j,d} + 1$ by appending a hop $h_{i,j}$ from $i$ to $j$ to the path $P^{j,d}$. Therefore, router $i$ knows the metric $m_A^{i,d}$ and the next hop towards subnet d (see Figure 2). If a RIP router has a valid path to subnet $d$, it will reject all equivalent or inferior paths to the same subnet. RIP prevents the proliferation of routing loops by implementing a limit on the number of hops allowed in a path from the source to a destination node. The maximum number of hops allowed is 15. This hop limit also limits the size of the diameter of a RIP network. A hop count of 16 is called infinity. It is considered an infinite distance and used to deprecate inaccessible routes. RIP mainly detects suddenly inaccessible routes by the expiration of timers due to missing confirmations. RIP uses three timer intervalls:

- an update timer (default 30 sec) for periodically sending out routing updates,
- a timeout timer (default 180 sec) to detect invalid entries in the routing table and
- a garbage collection timer (default 120 sec) to delete invalid entries from the routing table.

The timeout timer of an entry is re-initialized when the entry is confirmed by an incoming routing update. Should the timeout timer of the entry expire, the corresponding route is marked as unreachable with metric infinity (16), and the garbage collection timer is started. The garbage collection timer is stopped when the entry is confirmed by a valid routing update. If the garbage collection timer expires, the entry will be deleted from the routing table. RIP periodically sends routing updates every 30 seconds containing the whole routing table to all adjacent routers. As it is not useful to claim reachability for a subnet to the neighbor from which the route was learned, the *split horizon scheme* is defined in the RIP specification [12]. Split horizon omits routes learned from one neighbor in updates sent to that neighbor in order to avoid loops between two neighbor routers on one link. Another enhancement, Split horizon with poisoned reverse, includes such routes in updates, but sets their metrics to infinity. Routing updates are also sent with every change in the routing table only containing the affected entries. These event-driven routing updates are called triggered updates. RIP uses two kinds of message types, RIP updates and RIP requests which have the same message structure (see Figure 3) but differ in their processing. RIP updates are used to announce routing information to the adjacent routers. RIP requests are used to ask adjacent routers for routing information. The messages can be recognized by the value in the command field of the message structure. As RIP only uses one message structure type, it is very easy to implement and to deploy RIP in contrast to complex routing protocols like Open Shortest Path First (OSPF)[14]. The simple deployment of RIP is one reason why it is still widely used.

### B. RIP Request Messages

A RIP request is used to ask for a response containing all or part of a router's routing table. Requests and their treatments are part of the standard RIP specification [12]. Usually, RIP requests are sent as multicast messages to neighbors by routers which have just booted and try to fill in their routing tables.

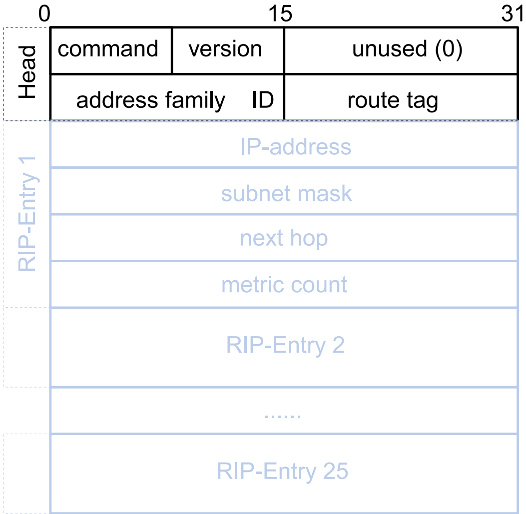



Fig. 3. The RIP message structure of the update and the request message as specified in [12].

Therefore, the most common request messages causes all neighbor routers to transmit all route entries in their routing table. Also single entries can be requested. An option which is mainly used for monitoring, e.g., by network monitoring tools.

The RIP request message is processed entry by entry (see Figure 3) and can have 25 entries at its maximum before a second message is needed. If there are no entries in a RIP request message, the requested router does not response. If there is exactly one entry in the request, and it has an address family identifier of zero and a metric of infinity (IP-Address 0.0.0.0 and metric 16), then this is a request to send the entire routing table. Except for this special case, the processing of the RIP request is quite simple. For each entry, the destination is looked up in the router's routing database and, if the route is listed, the route's metric is put in the metric field of the request message (see Figure 3). If there is no explicit route to the specified destination, the infinity value is put in the metric field. Once all the entries are filled in, the value in the command field is changed from request to response and the datagram is sent back to the requestor.

## V. The RMTI Protocol

The RMTI router evaluates redundant routing updates from different directions to the same destination. The RMTI protocol is implemented on top of the Routing Information Protocol (RIP). Thus, our RMTI-protocol is downward-compatible to RIP. As the RMTI extension does not change the message structure but only the processing algorithm, the RMTI technique can also enhance other distance vector routing protocols as well.

### A. *RMTI optimizations*

In distance vector routing the Counting-to-Infinity problem (CTI) reflects the routing loop problem. A group of RIP-Routers can get into an unstable state as soon as a CTI occurs. This problem cannot be solved by the well-known *split-horizon scheme*, because this approach can prevent CTIs in virtual loops but not in real loops. A real loop can be found in networks where at least three routers are connected to each other via separate links. Therefore the RIP protocol is restricted to a small network diameter in order to limit the impact of the CTI update sequence. This impact has two aspects:

1. wrong routing updates are proliferated and
2. payload packets are misguided.

The RMTI protocol avoids CTIs as well as routing loops and forwarding loops. If all routers in a network are RMTI-routers the infinity metric is no longer necessary. The RMTI protocol and its enhancements to RIP are described in [2][3][19]. The RMTI algorithm is an optimization that affects convergence time and update traffic. The update traffic is reduced by sending small neighbor-alive-notifications instead of periodic routing updates. An incoming neighbor-alive-notification confirms all routes received from this neighbor. Changes in the network will be advertised by event-triggered updates which are already part of the standard RIPv2 specification [12]. The neighbor-alive-notification is small in contrast to the ordinary periodic routing update whose size depends on the amount of subnets in the network. Moreover, due to the smaller neighbor-alive-notifications either

- the entire routing update traffic can be reduced or
- the sending interval and the convergence time of the network can be reduced.

There are two options to optimize the RMTI protocol either to have less update traffic or to reduce the convergence time.

### B. *RMTI with neighbor-alive-notifications*

The periodic routing updates are replaced by small neighbor-alive-notifications which accelerates the convergence time and the reliability of RMTI. In order to keep RMTI downward compatible to RIP, the neighbor-alive-notification is performed by empty RIP Request messages.

In correspondence to the RIP specification, a RIP router does not have to reply to a Request with no entries. An RMTI router can identify its neighbor as RMTI router, if it receives empty requests. The neighbor-alive-notification is very small and, is periodically sent via multicast to all neighbor RMTI routers. There is a time and a traffic optimization of the RMTI. The time optimized RMTI sends the neighbor-alive-notification every five seconds in order to detect link failures immediately. This optimization speeds up the convergence time. The traffic optimized RMTI's sending period is larger than 5 seconds in order to reduce the update traffic between the routers. Should an RMTI router detect an empty request from a neighbor router, it accepts this router as an RMTI router and does the following:

- Firstly, the timeout timers of all route entries in the routing table from this neighbor are reduced to ,e.g., 10 seconds. So, if the entries are not confirmed within 10 seconds they will be marked as unreachable.

- Secondly, every time a neighbor-alive-notification is received from the neighbor router, the corresponding route entries in the routing table will be confirmed at once and their timeout timer will be re-initialized.

The burden of periodical routing updates which contain the whole routing table could be omitted. The tedious routing update which transmits the whole routing table every 30 seconds is replaced by a small neighbor-alive-notification sent every 5 seconds. If a route is not confirmed by a neighbor in RIP for 180 seconds, it will be marked as unreachable. If a neighbor is not confirmed in RMTI within 10 seconds, all route entries from this neighbor will be marked as unreachable. Individual entry changes in the routing table are transmitted by the event-driven triggered updates further on.

The neighbor-alive-notification allows for *speeding up* the convergence time of the network in case of a link failure due to lower timeout and short-intervalled alive notifications. Since the triggered updates become most significant for the RMTI routing process, the sender needs an acknowledgment of the triggered updates. Our acknowledgment approach is based on *split horizon with poison reverse*, which is part of the RIPv2 specification [12]. All routes received by a triggered update are immediately acknowledged with metric infinity.

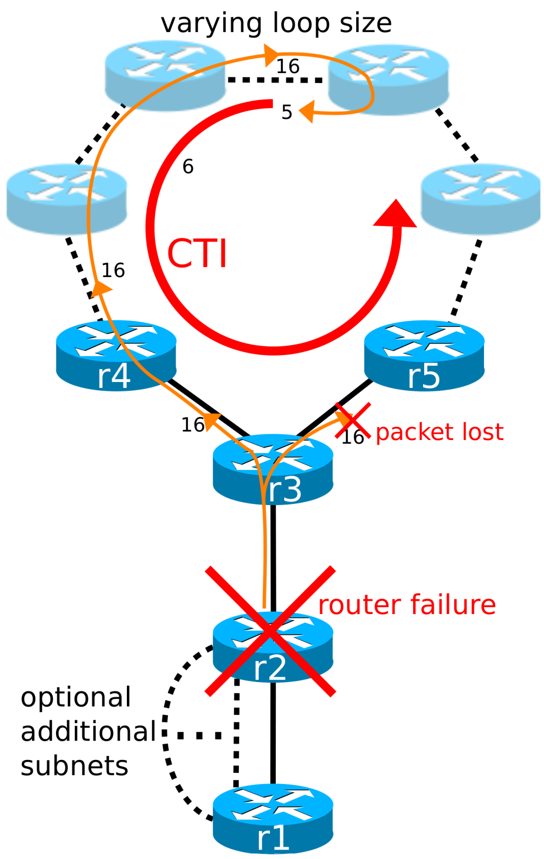



Fig. 5. Y-Topology class

## VI. Convergence Measurement

In this section, the tested topologies, the measurement methods, and the results of the convergence properties of the RIP and RMTI algorithms are presented.

### A. Tested Topologies

A simple network topology in which the CTI problem can occur is shown in Figure 5. This topology consists of a topology loop and a row of routers $r1$, $r2$ and $r3$. Router $r3$ is the node connecting the topology loop with the row. The respective subnets are peripherally located outside the topology loop connected to router $r1$ and $r2$. Routing information about these subnets can flow into a CTI sequence within the topology loop after a link failure between router $r2$ and $r3$ .

As depicted in Figure 5, a CTI can occur if the information of the unreachable subnet will be lost or just simply delayed in one path of the topology loop. If the link between router $r2$ and $r3$ fails, the subnets located between $r2$ and $r1$ are unreachable for $r3$. Router $r3$ does not receive any more updates and the timeout for these subnets expires. The routes in $r3$'s forwarding table are marked as unreachable with metric infinity. Then, router $r3$ sends a triggered update with metric infinity to router $r4$ and $r5$ and advertises this new routing information. Router $r4$ forwards the information about the unreachable subnets to the next router. In Figure 5 router $r5$ does not receive the routing information from $r3$ due to a transmission failure and the routers behind $r5$ in the topology loop do not receive the information about the unreachable subnets. At this point, the old invalid routing information prevails over the current routing one. The routers which have lost their route to the subnets accept the old routing information as new alternative routes. The well-known *split horizon scheme* cannot avoid the CTI situation. *Split horizon* is an extension to RIP which prevents a router from advertising routing information back to the router from which it was received. Router $r4$ gets the invalid routing information indirectly from router $r5$ and sends the information to $r3$. Router $r3$ accepts the invalid routing information from router $r4$ as new and valid alternative route to the subnets via $r4$. Then the CTI sequence starts. As long as the CTI occurs, a malicious routing loop is established.

For the test runs, several topologies were used which allow conclusions about expected correlations of a particular topology property and convergence properties. These topologies are presented in Figure 4. For the evaluation, over 300 topologies with a total of over 5000 test runs are analyzed. Due to the high computational complexity, the test topologies have been limited to a maximum size of 50 routers and 100 subnets. In order to obtain meaningful results it is important that any correlations between individual topologies are considered.

The following topologies are used in the test environment in order to analyse routing protocols (see Figure 4 and 5):

- the y-topology,
- the crown topology,
- the square topology,
- the Arpanet, described in [27], and
- the Internet 2 topology, used in [28].

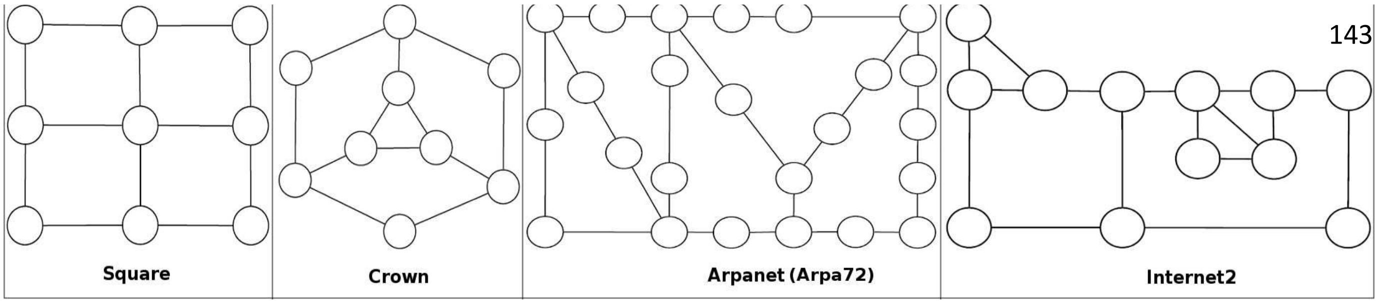



Fig. 4. The tested topologies: square, crown, arpanet, and internet2

### B. General Convergence Properties

In order to compare different routing algorithms, it is crucial to know how they behave in general. Coldstart tests are performed to measure convergence time and update traffic volume from the point in time where all router daemons are started until the network reaches the convergent state for the first time. For this test case, no CTI is provoked as the CTI situation is a special case. This test scenario was designed to test the propagation speed of the routing updates being the worst case scenario for convergence time and traffic.

These measurements covered over 300 (mostly randomly generated) topologies showing that the convergence time for both algorithms mainly depends on the network topologies diameter $d$. As shown in Figure 6 the convergence time increases in this test environment by a factor of 2.5 along with an increasing diameter $d$. However, the update traffic volume of both algorithms increases with polynomial growth with the number of subnets in the network (see Figure 7).

Test cases show that RMTI and RIP reache their convergent state at the same time (see Figure 6). If the time-optimized RMTI (Section V-A) is used, it does not affect the convergence time because the time benefits correspond to the detection of a topology change. With traffic improvement the RMTI needs 10 percent less traffic than RIP[2] (see Figure 7). The traffic volume for the time-optimized RMTI is the same as for RIP.

[2]To achieve this result, the first 60 seconds from coldstart are measured instead of stopping the measurement at the convergent state.

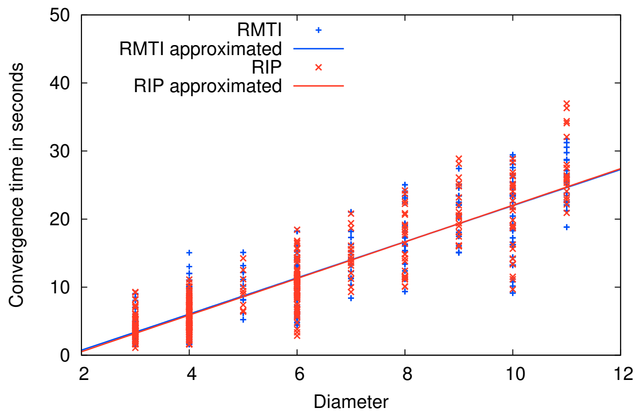



Fig. 6. Convergence time for RIP and RMTI at coldstart

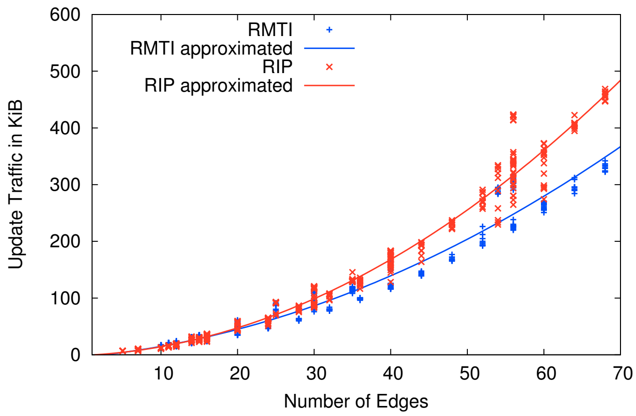



Fig. 7. Traffic comparison of optimized RMTI and RIP at coldstart

The analysis of the measurements shows that an even better equation for the approximation of convergence time $t_{approx}$ is achievable. Besides the network diameter $d$, the number of nodes $n$ in the network topology require to add the product $0.16 \cdot n$ to the convergence time. The variance is halved from 7.28 to 3.56. Now the approximated convergence time $t_{approx}$ from coldstart can be calculated for every network using the following Equation (2):[3]

$$t_{approx} = 2.5 \cdot d + 0.16 \cdot n - 5 \pm 1.89 \text{ [seconds] with:} \quad (2)$$

$t_{approx}$ = convergence time,
$d > 1$ = diameter of network topology ,
$n$ = Number of nodes in the network topology

In Figure 8 and 9 the dependencies between the number of edges (subnets) are shown for the square and the crown topology.

In Figure 10 and 11 the dependencies between the network diameter and the convergent time in seconds are shown for the square and the crown topology.

Fig 12 depicts the amount of traffic as function of the number of verticies (router nodes) and the number of edges (subnets) within a topology in relation to the traffic.

[3]The values may differ a bit depending on the test environment

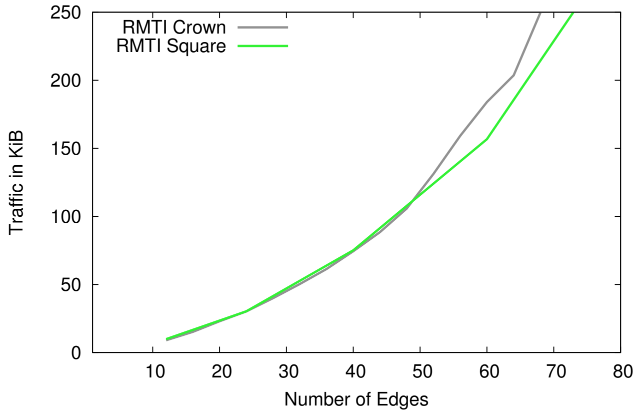



Fig. 8. Traffic comparison between the topologies square and crown with RMTI at coldstart.

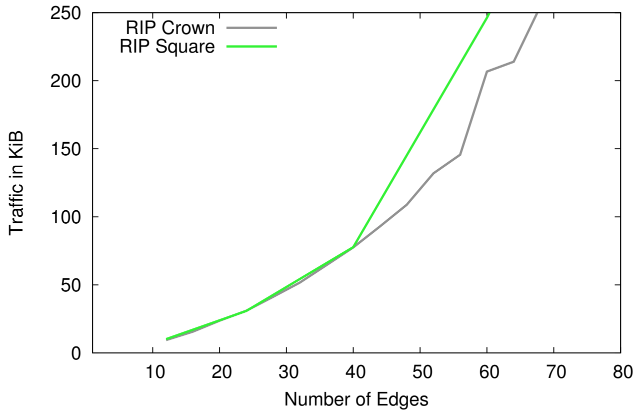



Fig. 9. Traffic comparison between the topologies square and crown with RIP at coldstart.

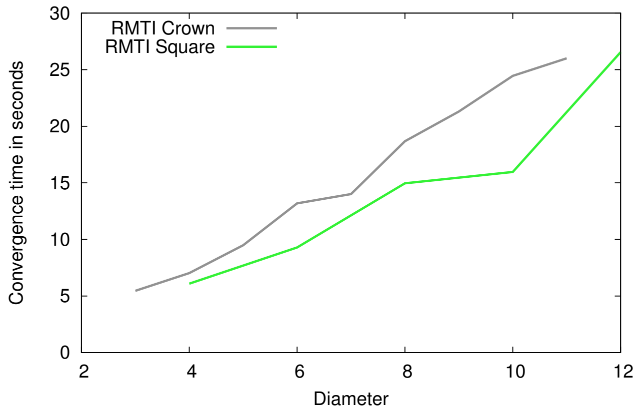



Fig. 10. Convergence time comparison between the topologies square and crwon with RMTI at coldstart.

## C. Importance of CTI avoidance

Despite the *Split Horizon* algorithm, CTIs can occur under certain circumstances when the network contains loops and links are flapping or routing updates are lost. To show the importance of CTI avoidance for distance vector algorithms,

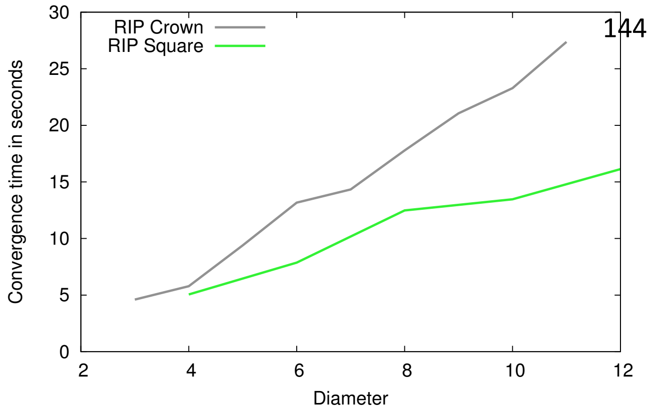



Fig. 11. Convergence time comparison between the topologies square and crwon with RIP at coldstart.

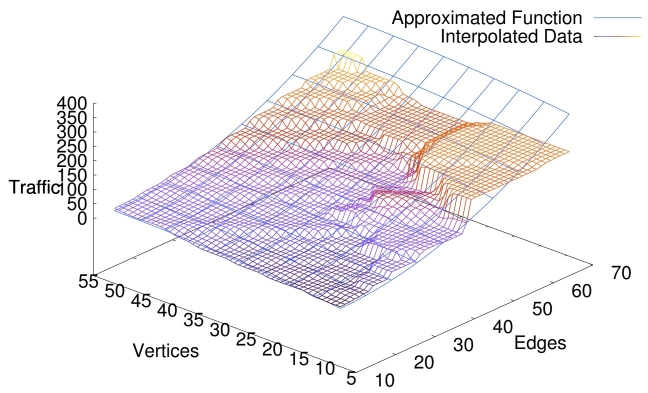



Fig. 12. Traffic comparison in relation to the growing number of edges (subnets) and verticies (router nodes).

some test runs have been performed that show the CTI frequency for links with different packet losses.

As CTIs can occur in Y-Topologies (Section 4) they have been tested with a loop size of 3 where a failure was provoked on one subnet. If the failure update of this subnet is announced to either upper router $r4$ or $r5$ (Figure 5), a CTI occurs. The bigger the loop size, the more probable is the occurrence of a CTI. This test case is a best case scenario because there are more subnets to pass and more possibilities to lose the information about the failed subnet in bigger loops.

As the results illustrate in Figure 13, the CTI frequency $F_{CTI}$ increases with the routing update packet loss $L_p$ (in percent) in a linear way.

The general rule is:

$$F_{CTI} = 0.55 \cdot L_p \pm 0.62 \text{ [\% (absolute)] with:} \qquad (3)$$

$F_{CTI}$ = CTI frequency in percent,
$L_p$ = packet loss in percent

With an update packet loss $L_p$ of 0.01%, a CTI is provoked with a probability of 0.0055% or averaged at every 180st

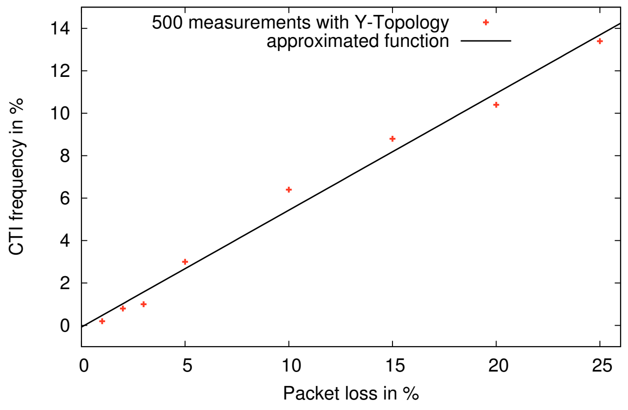



Fig. 13. CTI frequency depending on Packet loss

failing subnet.

### D. CTI Duration

The impact of CTI situations become apparent by examining their duration. As long as the CTI situation lasts, all payload traffic for the corrupted subnet is cycling in the loop. This is a burden for all routers and subnets located in the loop.

Since RMTI can detect routing loops and avoid CTI occurrences it eliminates this burden. To quantify these benefits the convergence time of RIP and RMTI have been measured in case of a CTI situation. As CTIs only occur in loops several Y-Topologies with different loop sizes (Figure 5) have been tested.

To provoke a CTI router, $r2$ (Figure 5) is shut down. This leads to an unreachability of a certain subnet. Router $r3$ is forced to propagate the new infinity metric of the unreachable subnet only to one of its neighbors ($r4$ or $r5$) so that this kind of update transfer must result in a CTI.

For each topology tested the time was measured starting when a router detected a failing subnet and ending when the network was convergent again. Hence, these results are independent of the RMTI optimization (time or traffic - see Section V-A), because time optimization only speeds up the topology change detection.

The result graph (Figure 14) shows that the benefit of RMTI is between 30 and 120 seconds. The average CTI duration with increasing loop size increases with a factor of 18.6 for RMTI. That is nearly one fourth of RIP's growth factor which is 74. The average CTI duration of all tested topologies with RMTI only needs 30% of the average RIP CTI duration. The irregularities (see Figure 14) of the RIP results are generated when the update timer of one router expires and affects the CTI in reaching infinity (metric 16). This only happens for certain loopsizes together with specific timer settings. For these tests the standard RIP update timer of 30 seconds was used. With other timer settings this behavior occurs with other loop sizes. RMTI can converge so much faster because RMTI stops the CTI at router $r3$ (Figure 5).

Starting the measurement with the topology change, RIP

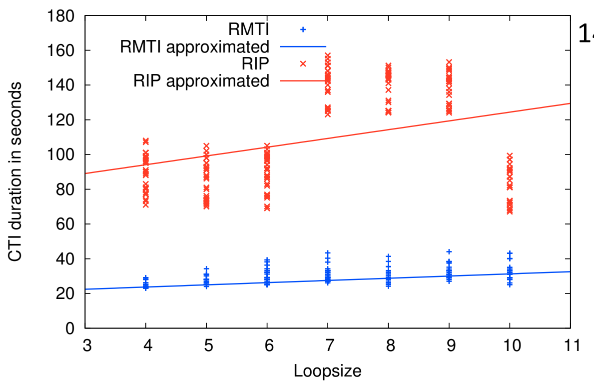



Fig. 14. CTI duration

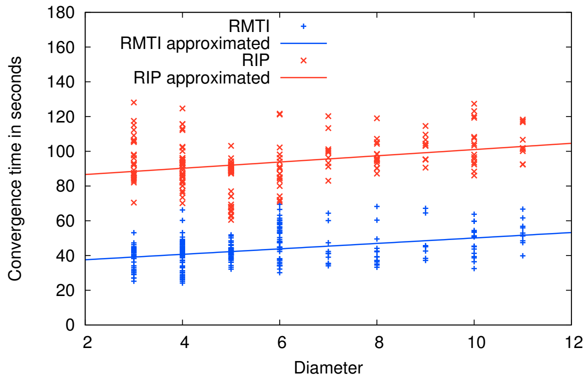



Fig. 15. Convergence time for random router failures

converges slower because of its long timeout timer of 180 seconds.

### E. Common Convergence Behavior

As the most common topology changes do not provoke CTIs (Figure 13), the convergence properties were also measured in case a router fails without CTI provocation. Therefore, a random router has been shut down in each test run. As shown in Figure 15 the RMTI algorithm with time optimization (Section V-A) converges 50 seconds faster on average than the normal RIP from the time point where the topology change happens. With a traffic optimized RMTI the convergence time is the same as with RIP, but there is a noticeble update traffic reduction.

## VII. Conclusion and Future Work

To demonstrate the advantages of new routing algorithms in contrast to common intra domain routing algorithms, comprehensive tests have to be carried out. In practice, a variety of network topologies and a succession of critical events have to be automatically generated by random generators. Test situations concerning the capability of a routing algorithm to re-organize itself after a topology change have to be covered.

With the newly developed test environment, one can show the benefits of the new RMTI algorithm in comparison to RIP. The flexible design of the test environment allows us to test RIP, RMTI and other intra domain routing protocols. The RMTI algorithm shows two important advantages: the possibility to avoid CTI situations and to converge much faster than other distance vector algorithms in case of a topology change. The ability of RMTI to choose whether to optimize a fast topology change detection or a traffic reduction (or a mixture of both) makes the test environment adaptable to the specific needs of many different networks. With the VNUML script, it is also possible to run larger networks on computer clusters which will support the results for those networks. The high degree of automating the test environment and the ability to gather large amounts of measurement data is a big advantage for analyzing all kinds of routing protocols. Another issue that we will deal with in our future research is the performance analysis of the RMTI algorithm with payload traffic and the comparison with other intra domain routing protocols.

We have shown that we can obtain a variety of characteristic routing parameters and various relationships between them.

## References


[1] F. Bohdanowicz, M. Jakobs, and Ch. Steigner, *Statistical Convergence Analysis of Routing Protocols*, in Proceedings of the Nineth International Conference on Networking (ICN 2010), Les Menuires, France, 2010.

[2] F. Bohdanowicz, H. Dickel, and Ch. Steigner, *Routing with metric-based Topology Investigation*, International Journal on Advances in Internet Technology, 2009, vol 2 no 1.

[3] F. Bohdanowicz, H. Dickel, and Ch. Steigner, *Metric-based Topology Investigation*, in Proceedings of the Eighth International Conference on Networking (ICN 2009), Cancun, Mexico, 2009.

[4] Ch. Steigner, H. Dickel, and T. Keupen, *RIP-MTI: A New Way to Cope with Routing Loops*, in Proceedings of the Seventh International Conference on Networking (ICN 2008), Cancun, Mexico, 2008.

[5] C. Cheng, R. Riley, S.P.R. Kumar, and J. J. Garcia-Luna-Aceves, *A loop-free extended bellman-ford routing protocol without bouncing effect*, ACM Sigc. Symp. Commun. Arch. and Prot., pp. 224-236, 1989.

[6] J. Dike, *User Mode Linux*, Prentice Hall, 2006.

[7] The user-mode-linux project page, URL: http://user-mode-linux.sourceforge.net, 18.01.2011.

[8] C. Berge, *Graphs and Hypergraphs*, North-Holland Publishing Company, 1973.

[9] B. Rajagopalan and M. Faiman, *A new responsive distributed shortest-path routing algorithm*, ACM Sigcomm Symposium Commun. Arch. and Protocols, pp. 237-246, 1989.

[10] J.J. Garcia-Luna-Aceves, *Loop Free Routing Using Diffusing Computations*, IEEE Transactions on Networking, 1993.

[11] K. Levchenko, G. M. Voelker, R. Paturi, and S. Savage, *XL: An Efficient Network Routing Algorithm*, Proc. Sigcomm 2008, August, 2008.

[12] G. Malkin, *RIP Version 2*, RFC 2453, 1998.

[13] V. Fuller, T. Li, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*, RFC 4632, 2006.

[14] J. Moy, *OSPF Version 2*, RFC 2328, 1998.

[15] C. E. Perkins, *Ad hoc networking*, Addison-Wesley, Amsterdam 2001.

[16] C. E. Perkins, E. Belding-Royer, S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, RFC 3561, 2003.

[17] The quagga project page, URL: http://www.quagga.net, 18.01.2011.

[18] Y. Rekhter, T. Li, S. Hares, *A Border Gateway Protocol 4*, RFC 4271, 2006.

[19] A. Schmid and Ch. Steigner, *Avoiding Counting to Infinity in Distance Vector Routing*, Telecommunication Systems 19 (3-4): 497-514, March - April, 2002, Kluwer Academic Publishers.

[20] VNUML project home page, URL: http://www.dit.upm.es/vnuml, Technical University of Madrid (UPM), 18.01.2011.

[21] tcpdump project home page, http://www.tcpdump.org, 18.01.2011.

[22] OpenWRT Project home page, http://www.openwrt.org, 18.01.2011.

[23] Andrew S. Tanenbaum, *Computer Networks*, 3rd ed., Prentice Hall PTR, 1996, pp. 358-359

[24] Z. Zhong, R. Keralapura, S. Nelakuditi, Y. Yu, J. Wang, C. Chuah and S. Lee, *Avoiding Transient Loops Through Interface-Specific Forwarding*, Transactions on Networking, IEEE/ACM, Volume: 15, Issue: 6, Dec. 2007.

[25] F. Galan, D. Fernandez, W. Fuertes, M. Gmez and J. Lpez de Vergara, *Scenario-based Virtual Network Infrastructure Management in Research and Educational Testbeds with VNUML: Application Cases and Current Challenges*, Annals of Telecommunications, special issue on Virtualization: a path for the future Internet, vol. 64(5), pp. 305-323, May 2009.

[26] B. Premore, *An Experimental Analysis of BGP Convergence Time*, ICNP '01: Proceedings of the Ninth International Conference on Network Protocols, IEEE Computer Society, 2001.

[27] Frank, Howard, Robert E. Kahn und Leonard Kleinrock, *Computer communication network design: Experience with theory and practice*, in Proceedings of the spring joint computer conference (AFIPS '72 Spring), May 1972.

[28] Gunes, Mehmet H. und Kamil Sarac, *Inferring Subnets in Router-level Topology Collection Studies*, in Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC '07), Oktober 2007.

[29] *Internet 2 Fact Sheet*, URL: http://www.internet2.edu/resources/AboutInternet2.pdf,18.01.2011.

# Media Streaming Observations: Trends in UDP to TCP Ratio

DongJin Lee, Brian E. Carpenter, Nevil Brownlee
{dongjin, brian}@cs.auckland.ac.nz, nevil@auckland.ac.nz
Department of Computer Science
The University of Auckland, New Zealand

***Abstract*—Widely used protocols (UDP and TCP) are observed for variations of the UDP to TCP ratio, elastic (and inelastic) flow behaviors, and of port number distribution, both over time and between different networks. The purpose of the study was to understand the impact of application trends, especially the growth in media streaming, on traffic characteristics. The results showed substantial variability but little sign of a systematic trend over time, and only wide spreads of port number usage. Despite the large network traces, the ratios appear to be rather dependent on application popularity (and their diversities), and so one cannot extrapolate from usage patterns on one network to those on another without allowing for at least as much variability as we have observed in this work.**

***Index Terms*—network traffic statistics; observation; UDP to TCP ratio; flow; volume; port number; streaming**

## I. Introduction

Along with annual bandwidth growth rates reported to be 50% to 60% per year both in the U.S. and worldwide [7], Internet traffic types, characteristics and their distributions are always changing. For example, recent Internet observations [15] [21] find that the majority of traffic and infrastructures have migrated to a small number of very large providers, such as those supporting cloud computing. Also, it has been widely predicted that within a few years, a large majority of network traffic will be audio and video streaming. Cisco's Visual Networking Index [4] has been actively involved in traffic forecasting, e.g., *Hyperconnectivity and the Approaching Zettabyte Era* [5]. Those reports assert that in 2010 video will exceed p2p in volume, becoming the main source of future IP traffic growth, and over 60% of all consumer Internet traffic will be video by 2013. They also state that video traffic can change the economic equation for service providers, given that video traffic is many times less valuable per bit than other content such as SMS service. Additional to the increased computational resources, increases in monitor screen size and its resolution give rise to larger document sizes (such as more pixels in images and videos), thus generating more traffic than before.

A common expectation in the technical community has been that streaming traffic would naturally be transmitted over UDP, probably using RTP, or perhaps in future over DCCP. Another view is that UDP and TCP might replace IP as the lowest common denominator [28] to achieve transparency through NATs and firewalls. Then, if non-TCP congestion control, signaling or other features are needed, a protocol must be layered on top of UDP instead of developing a better transport layer. This, if accompanied by a vast increase in streaming, would change the historic pattern whereby most traffic benefits from TCP's congestion management. Indeed, if the predicted increase in streaming traffic were to remove most flows from any form of congestion control, the consequences would be serious. Therefore, the evolution of the observed UDP to TCP ratio in actual Internet traffic is a subject of interest. Also, observing for trends in network statistics such as distributions of port numbers and flow characteristics are beneficial in network management. This paper is an expanded version of our earlier work on these topics [23].

We note that audio/video 'streaming' is not really a well-defined term, and it covers a variety of technologies. For example, video-on-demand packets are usually transmitted over TCP; streams are downloaded fully, then played from the local copy. This is suitable when the timeliness and bandwidth variability are not crucial. In others such as voice-over-IP solutions, with timeliness a high priority, streams are transmitted over UDP. Also, recent application advances allow streaming concepts to be much more diverse, such as p2p-based streaming and practical use of progressive download on a faster-than-real-time basis [25]. Furthermore, some streaming applications choose dynamically whether to use UDP, raw TCP or HTTP over TCP.

The UDP to TCP ratio has been briefly observed in [1], where UDP flows are often responsible for the largest fraction of traffic. Their summary indeed suggests that the current ratio can change with increasing demand for IPTV and UDP-based real-time applications. We note that both TCP and UDP traffic are useful in distinctive ways. UDP can be advantageous due low overheads especially in an organization's high performing storage systems, such as in SANs and NFS. In other words, traffic statistics can be largely different by the environments and our measurement scope is at Internet scale.

Our expectation was that the growth in streaming traffic would be reflected in a steady growth in the UDP to TCP ratio, or in a systematic change in the relative usage of various port numbers, or both. We conducted a preliminary survey on the basis of readily available data from a variety of measurements, in both commercial and academic networks, between 1998 and 2008. It showed that the UDP to TCP ratio, measured by number of packets, varied between 5% and 20%, but with no consistent pattern over the ten years. A report in [22]

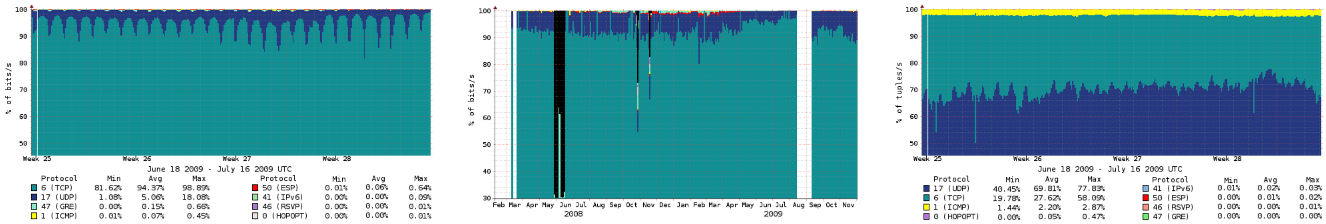


Fig. 1. CAIDA (2008–2009), Left: DirA – 4 weeks (bits), Center: Dir DirA – 20 months (bits), Right: DirB – 4 weeks (flows)

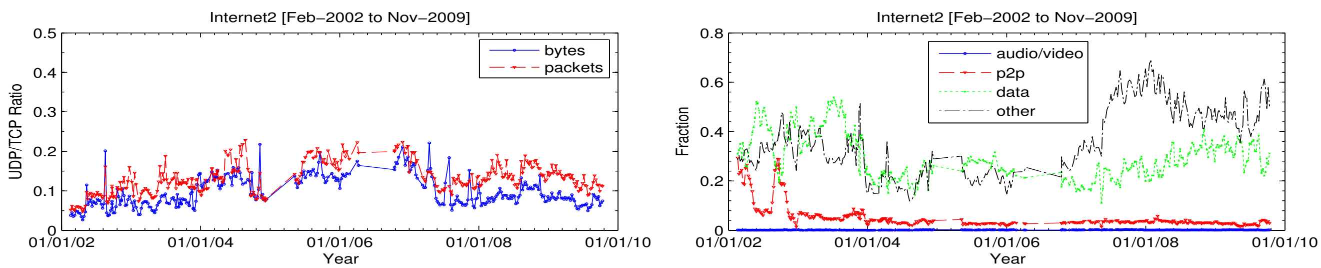



Fig. 2. Internet2 (2002-2009), Left: UDP to TCP ratio, Right: volume fractions of four application types (assigned from the Internet2)

observed an average ratio of 0.07 over the 78 ISPs globally. For Internet2, it was 0.05 in 2002, 0.22 in 2006, and 0.15 in 2008. Similar inconsistencies showed up in partial data from observations in Norway, Sweden [18], Japan, Germany, the UK, and elsewhere. These inconsistencies were surprising, and did not suggest a steady growth in UDP streaming. To better understand these issues, we observe how TCP and UDP traffic have varied over the years, either by number of flows, by their volume/duration, or by their traffic kinds.

We consider this study to be valuable to service providers and network administrators managing their traffic. This includes outlining observed statistical datasets to derive strategies, such as classifying application types, prioritizing specific flow types and provisioning based on usage scenarios. Also, a definite trend in the fraction of non-flow-controlled UDP traffic might affect router design as far as congestion and queue management is concerned. In this paper, we particularly observe two behaviors, 1) variation of UDP to TCP ratio over time, and 2) port numbers and elastic/inelastic flow distributions. As far as is possible from the data, we also observe application trends. We use the term "flow ratio" and "volume ratio" to represent the ratio of $\frac{UDP}{TCP}$ for their flow counts and data volumes respectively.

## II. Longitudinal Data

Long term protocol usage is observed from two locations: the CAIDA [2] and Internet2 [6] traffic[1]. CAIDA traffic data is from the OC192 backbone link of a Tier 1 ISP between Chicago and Seattle (direction A and B), reflecting various end-user aggregates. The Internet2 traffic reflects usage patterns by the US research and education community. Both datasets have HTTP and DNS traffic as the most widely used protocols for TCP and UDP respectively, but no particular specific application protocol was used predominantly.

Figure 1 shows plots for the CAIDA data. Although protocols such as ICMP, ESP and GRE are observed as well, TCP and UDP are in general most widely observed. We did not see a noticeable amount of SCTP or DCCP traffic. We observe that both DirA and DirB traffic contained about 95% TCP and 4% UDP bytes, measured daily and monthly (left and right). The volume ratio varied around an average of 0.05; the diurnal variation shows that during the peak time TCP volume (mainly HTTP) contributed as high as 98%, and during the offpeak time UDP volume can increase to 18%. Flow proportions (DirB, right plot) varied greatly as UDP flows are a lot more observed than TCP flows, e.g., on average 70% and as high as 77% of all flows are UDP. ICMP flows are observed stably, contributing about 2%.

The dataset from Internet2 (Figure 2) covers a longer period of measurement, from February 2002 to November 2009. On the left, we observe that the volume ratio has increased from early 2002 to mid 2004, then decreased from late 2006 to mid 2007, and again slight variations are observed from mid 2007 on. The UDP decrease observed in 2006 to 2007 may be due to the University of Oregon switching off a continuous video streaming service [17]. Generally the volume ratio varied between 5% and 20%, showing a higher variation than that of the CAIDA data. Comparing between 2002 and 2009, we find that the ratio of both bytes and packets has increased slightly by about 5%.

In this, there seems to be little evidence of change in

[1]Note that the datasets contained some irregular anomalies throughout the period which have been removed from the plots. For example, short but very high peak usage of unidentified protocol, missing-data and inconsistent data values were observed and discussed with the corresponding authors at CAIDA and Internet2. They are presumed to be due to occasional instrumentation errors or, in some cases, to overwhelming bursts of malicious traffic. If included in the analysis, they would dominate the traffic averages and invalidate overall protocol trends. The original data including these anomalous peaks are available at the cited web sites.

TABLE I
SUMMARY OF NETWORK TRACES

| Trace Name | Network Type | Date, [Starting time], Duration (hours) | Average Rate (Mb/s) | Volume: Bytes (GB) | Volume: TCP (%) | Volume: UDP (%) | Volume: ICMP (%) | Volume: Other (%) | Volume: $\frac{UDP}{TCP}$ Ratio | Number of Flows: Flows (M) | Number of Flows: TCP (%) | Number of Flows: UDP (%) | Number of Flows: ICMP (%) | Number of Flows: $\frac{UDP}{TCP}$ Ratio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AUCK-99 | UNIV | 1999-Nov-29, [13:42], 24.00 | 1.39 | 14.96 | 94.26 | 5.51 | 0.19 | 0.04 | 0.06 | 2.63 | 82.52 | 15.32 | 2.17 | 0.19 |
| AUCK-03 | UNIV | 2003-Dec-04, [00:00], 24.00 | 6.32 | 68.23 | 93.25 | 6.14 | 0.24 | 0.34 | 0.07 | 19.49 | 75.53 | 21.85 | 2.63 | 0.29 |
| AUCK-07 | UNIV | 2007-Nov-01, [16:00], 24.00 | 60.41 | 652.41 | 94.70 | 4.72 | 0.43 | 0.15 | 0.05 | 73.62 | 44.44 | 52.73 | 2.82 | 1.19 |
| AUCK-09 | UNIV | 2009-Aug-03, [09:00], 11.00 | 375.93 | 1860.85 | 93.77 | 6.12 | 0.02 | 0.08 | 0.07 | 93.84 | 59.65 | 39.45 | 0.90 | 0.66 |
| BELL-I-02 | ENT | 2002-May-20, [00:00], 96.00 | 1.78 | 76.79 | 90.70 | 8.58 | 0.05 | 0.66 | 0.09 | 6.42 | 94.39 | 3.68 | 1.98 | 0.04 |
| CAIDA-DirA-02 | BB | 2002-Aug-14, [09:00], 3.00 | 363.14 | 490.24 | 94.91 | 3.83 | 0.09 | 1.17 | 0.04 | 45.95 | 84.86 | 12.73 | 2.4 | 0.15 |
| CAIDA-DirB-03 | BB | 2003-Apr-24, [00:00], 1.00 | 117.93 | 53.07 | 94.86 | 4.66 | 0.10 | 0.38 | 0.05 | 11.49 | 78.59 | 19.28 | 2.13 | 0.24 |
| CAIDA-DirA-09 | BB | 2009-Mar-31, [05:59], 1.03 | 1250.83 | 579.76 | 96.69 | 2.74 | 0.48 | 0.09 | 0.03 | 46.96 | 43.16 | 54.46 | 2.38 | 1.26 |
| CAIDA-DirB-09 | BB | 2009-Mar-31, [05:59], 1.03 | 3687.70 | 1709.25 | 91.17 | 8.11 | 0.06 | 0.66 | 0.09 | 61.03 | 32.50 | 65.06 | 2.44 | 2.00 |
| ISP-A-99 | COMML | 1999-Nov-02, [14:04], 28.28 | 0.36 | 4.60 | 98.16 | 1.75 | 0.08 | 0.01 | 0.02 | 0.78 | 61.63 | 37.03 | 1.34 | 0.60 |
| ISP-A-00 | COMML | 2000-Jan-04, [09:47], 32.80 | 0.37 | 5.44 | 94.37 | 5.44 | 0.08 | 0.12 | 0.06 | 0.94 | 57.86 | 40.68 | 1.46 | 0.70 |
| ISP-B-05 | COMML | 2005-Jun-09, [07:00], 24.00 | 275.16 | 2971.74 | 92.26 | 6.93 | 0.22 | 0.59 | 0.08 | 513.76 | 62.88 | 33.79 | 3.32 | 0.54 |
| ISP-B-07 | COMML | 2007-Feb-08, [00:00], 24.00 | 341.66 | 3689.90 | 94.43 | 5.05 | 0.12 | 0.40 | 0.05 | 500.56 | 49.61 | 46.35 | 4.05 | 0.93 |
| LEIP-II-03 | UNIV | 2003-Mar-21, [21:00], 24.00 | 25.30 | 273.26 | 88.75 | 9.40 | 0.15 | 1.70 | 0.11 | 54.99 | 60.15 | 35.58 | 4.28 | 0.59 |
| NZIX-II-00 | IX | 2000-Jul-06, [00:00], 96.00 | 3.50 | 151.38 | 87.35 | 9.23 | 3.39 | 0.03 | 0.11 | 55.28 | 47.18 | 29.88 | 22.94 | 0.63 |
| SITE-I-03 | ENT | 2003-Aug-20, [04:20], 24.00 | 24.86 | 268.44 | 98.50 | 0.61 | 0.81 | 0.08 | 0.01 | 30.72 | 36.41 | 5.46 | 58.13 | 0.15 |
| SITE-II-06 | ENT | 2006-May-11, [15:30], 33.90 | 76.52 | 1167.32 | 98.96 | 0.76 | 0.01 | 0.26 | 0.01 | 21.76 | 79.37 | 19.32 | 1.62 | 0.24 |
| SITE-III-04 | COMML | 2004-Jan-21, [06:00], 24.30 | 110.15 | 1204.52 | 94.26 | 5.24 | 0.21 | 0.25 | 0.06 | 156.69 | 67.80 | 24.11 | 8.10 | 0.36 |
| WITS-04 | UNIV | 2004-Mar-01, [00:00], 24.00 | 3.45 | 37.29 | 93.29 | 5.45 | 0.42 | 0.83 | 0.06 | 15.68 | 41.76 | 54.77 | 3.50 | 1.31 |
| WITS-05 | UNIV | 2005-May-12, [00:00], 24.00 | 5.41 | 58.40 | 97.22 | 2.19 | 0.14 | 0.45 | 0.02 | 18.33 | 56.76 | 42.12 | 1.12 | 0.74 |
| WITS-06 | UNIV | 2006-Oct-30, [00:00], 24.00 | 7.34 | 79.25 | 95.83 | 3.42 | 0.29 | 0.45 | 0.04 | 27.75 | 33.43 | 65.03 | 1.54 | 1.95 |

protocol ratio, as most are diurnal variations with no particular increasing or decreasing patterns. On the right, both "audio/video" and "p2p" traffic are little utilized over the period, whereas "data" (consisting mainly of HTTP traffic) and "other" (using ephemeral port numbers) traffic have increased. For example, audio/video traffic contributes to about 0.3% and p2p traffic decreased from about 20% to only about 2%. This could indicate that audio/video streaming and file sharing have genuinely decreased as compared to typical HTTP traffic, or that there are emerging applications using arbitrary port numbers or 'hiding' such traffic inside HTTP (e.g., [19]). Indeed, since about beginning of 2007, both the data and other traffic have increased substantially, from about 20% to more than 50%.

## III. NETWORK STATISTICS

We next report observations from various networks[2] covering different network types in different years. Table I shows a summary of measured traces. In total, our traffic meter measured 21 traces ranging from the university, backbone, commercial, exchange and enterprise. Average rate varied from 0.4Mb/s to 3.7Gb/s, contributing from 5GB to 3.7TB (counting IP payloads only). A flow is identified by a series of packets with the same 5-tuple fields (source/destination IP address, source/destination port number, and protocol) and terminated by the fixed-timeout of 30 seconds. Since a flow is unidirectional, flow's source port number is used for observations.

Volume ratio varied between 0.02 and 0.11, showing that the TCP volume contributed the most traffic (avg: 0.06 and std: 0.03 across the networks). The UDP volume contributed about 1% to 9%, marginally small compared to TCP. In particular, the NZIX-II-00 and LEIP-II-03 networks had the highest ratio (about 9% UDP percentages), but they showed quite different port number usages. For example, NZIX-II-00 had the most UDP volume on port 53 (DNS) and 123 (NTP) while LEIP-II-03 had the most p2p UDP volume – port 4672 (eD2k) and 6257 (WinMX).

Considering the number of flows, the flow ratio varied between 0.04 and 2.00 (avg: 0.7 and std: 0.56 across the networks showing higher variation than the volume). For example, AUCK networks have the ratio increased from 0.19 (1999) to 1.19 (2007), then decreased to 0.66 (2009). Over time the WITS and CAIDA networks also have the ratio increased up to 1.95 (2006) and 2.00 (2009) respectively. Other networks are similar, though not systematic. Compared with volume, it shows that UDP flows in general are more frequently observed than TCP, but are mainly smaller in bytes. As far as total contributions are concerned, there is no observed trend to longer, fatter UDP flows as we might expect from streaming.

One reason why the flow ratios might fluctuate a lot, even for the same network, is that UDP seems to be used a lot for malicious transmission. A port scan, for example, generates many flows containing only a single packet by enumerating a large range of port numbers. Another reason might likely be due to small-sized signaling flows, which are often used by emerging applications. The flow ratio has been observed to be as high as 3.00 in other networks [1].

### A. *Elastic and Inelastic traffic flows*

Although the UDP to TCP ratios observed previously do not appear to indicate a potential growth in media streaming, protocols such as RTSP and NMSP can be carried by TCP and UDP. If elastic traffic (e.g., FTP) has similar flow characteristics to inelastic traffic (e.g., RTSP) such as durations, volume or packet delays, then we might be seeing behaviors whereby the traffic kinds may no longer stand out, making it much harder to even distinguish between elastic and inelastic traffic. In this, examining for the behavior differences between elastic and inelastic traffic kinds in actual Internet traffic is also a subject of interest.

Table II shows traffic volumes by three categories (elastic1,

[2] CAIDA [2], NLANR PMA [8] and WAND [12]

TABLE II
STATISTICS OF ELASTIC AND INELASTIC FLOW KINDS

| Trace Name | *elastic1* (%) ≥1 | ≥10 | ≥50 | ≥100 | *elastic2* (%) ≥1 | ≥10 | ≥50 | ≥100 | *inelastic* (%) ≥1 | ≥10 | ≥50 | ≥100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AUCK-99 | 62.94 | 53.62 | 28.64 | 21.09 | 13.13 | 12.99 | 11.34 | 10.89 | 3.29 | 3.28 | 3.25 | 3.24 |
| AUCK-03 | 73.44 | 65.81 | 42.96 | 31.17 | 9.77 | 9.60 | 8.01 | 7.59 | 1.88 | 1.86 | 1.85 | 1.85 |
| AUCK-07 | 55.56 | 52.32 | 43.76 | 38.50 | 5.2 | 5.03 | 4.55 | 4.33 | 1.71 | 1.71 | 1.71 | 1.71 |
| AUCK-09 | 75.02 | 73.72 | 66.06 | 61.38 | 1.69 | 1.69 | 1.55 | 1.48 | 3.34 | 3.34 | 3.34 | 3.33 |
| BELL-I-02 | 29.80 | 26.17 | 16.39 | 12.75 | 5.71 | 5.69 | 4.80 | 4.55 | 2.85 | 2.84 | 2.83 | 2.83 |
| CAIDA-DirA-02 | 58.90 | 52.87 | 39.83 | 33.89 | 2.17 | 2.12 | 1.73 | 1.66 | 3.41 | 3.40 | 3.39 | 3.38 |
| CAIDA-DirB-03 | 65.44 | 56.63 | 40.37 | 33.38 | 1.29 | 1.21 | 0.84 | 0.79 | 0.89 | 0.86 | 0.85 | 0.85 |
| CAIDA-DirA-09 | 38.51 | 36.45 | 32.64 | 30.66 | 38.05 | 37.71 | 37.48 | 37.45 | 0.35 | 0.34 | 0.34 | 0.34 |
| CAIDA-DirB-09 | 54.29 | 51.91 | 47.25 | 43.05 | 0.90 | 0.86 | 0.69 | 0.65 | 1.24 | 1.22 | 1.22 | 1.22 |
| ISP-A-99 | 37.79 | 32.79 | 19.22 | 13.33 | 21.60 | 21.37 | 18.91 | 17.94 | 0.13 | 0.13 | 0.12 | 0.12 |
| ISP-A-00 | 46.38 | 41.07 | 25.93 | 20.12 | 10.80 | 10.59 | 8.72 | 8.25 | 0.76 | 0.75 | 0.74 | 0.72 |
| ISP-B-05 | 17.71 | 16.07 | 11.96 | 9.82 | 3.63 | 3.58 | 3.16 | 2.92 | 1.58 | 1.55 | 1.52 | 1.49 |
| ISP-B-07 | 33.67 | 32.10 | 28.47 | 26.47 | 1.34 | 1.31 | 1.14 | 1.07 | 1.69 | 1.65 | 1.62 | 1.59 |
| LEIP-II-03 | 22.74 | 20.55 | 15.37 | 12.98 | 1.77 | 1.74 | 1.58 | 1.52 | 2.37 | 2.35 | 2.34 | 2.33 |
| NZIX-II-00 | 53.64 | 45.76 | 23.92 | 16.95 | 19.38 | 19.13 | 17.65 | 16.98 | 1.13 | 1.13 | 1.11 | 1.11 |
| SITE-I-03 | 16.38 | 15.06 | 11.70 | 10.15 | 47.12 | 47.09 | 46.88 | 46.54 | 0.74 | 0.74 | 0.73 | 0.73 |
| SITE-II-06 | 15.35 | 14.51 | 12.01 | 10.72 | 25.24 | 25.23 | 25.16 | 25.09 | 0.57 | 0.56 | 0.56 | 0.56 |
| SITE-III-04 | 39.74 | 36.57 | 26.71 | 22.88 | 4.92 | 4.90 | 4.5 | 4.41 | 4.85 | 4.84 | 4.83 | 4.82 |
| WITS-04 | 70.73 | 61.85 | 37.16 | 26.90 | 7.38 | 7.23 | 5.38 | 4.97 | 2.06 | 2.05 | 2.05 | 2.05 |
| WITS-05 | 77.51 | 65.80 | 42.57 | 32.72 | 10.23 | 9.97 | 7.53 | 6.76 | 0.94 | 0.94 | 0.93 | 0.93 |
| WITS-06 | 75.58 | 68.97 | 49.64 | 40.56 | 9.44 | 8.96 | 5.98 | 5.40 | 0.44 | 0.43 | 0.42 | 0.42 |

elastic2 and inelastic flows)[3]. Each category of the flow kinds is then observed by packet counts; ≥1, ≥10, ≥50 and ≥100. Generally, elastic traffic contributed most of the traffic volume and varied more than the inelastic traffic. The elastic1 (HTTP/S) traffic flows particularly contributed the highest volumes (as high as 77.5% `WITS-05`). When the number of packets per flow are ranked by percentile, we see a noticeable decrease in the values, e.g., 75% with ≥1 packet down to 61.4% with ≥100 packets (`AUCK-09`), 33.7% to 26.5% (`ISP-B-07`), and 70.7% to 26.9% (`WITS-04`). This shows that elastic flows tend to carry a small number of packets. The elastic2 traffic is similiar but the percentiles decrease much less.

Conversely, it seems particularly noticeable that inelastic traffic shows almost no percentile decrease, even considering their relatively small proportions. For example, we observe 1.24% with ≥1 packet down to just 1.22% with ≥100 packets (`CAIDA-DirB-09`). With such small percentile decreases, the inelastic flows by themselves generally seem to carry larger volumes, and are perhaps longer lived. Inelastic traffic *volume* varied between 0.13% and 4.85%, however it shows no particular sign of increasing or decreasing. This also applied to the same network in different years.

We measured two flow characteristics in detail; flow lifetime and flow inter-packet variance. We particularly observed flows with at least 50 packets, since the flows carrying small numbers of packets do not represent appropriate user data transmission. Two flow characteristic distributions are shown in Figure 3 and Figure 4 in the Appendix. Both Figures show the flows with at least 50 packets; flows with at least 100 packets are observed to be similar (and generally log-normal distributed).

For lifetime plots, we observe that in nearly all of the networks, the distributions of the inelastic flows clearly stand out by lasting a lot longer compared to the elastic ones. For example, `CAIDA-DirB-09` has about 40% of the inelastic flows lasting up to one minute while 90% of the elastic ones last up to one minute. Similarly, `SITE-I-03` has about 50% of the inelastic flows lasting for more than ten minutes while virtually no elastic flows lasts for more than ten minutes.

To further observe streaming-like behaviors, we measure inter-packet arrival times in individual flows so as to observe their variations. In general, inelastic flows should have approximately constant inter-packet arrival times (low variance). To compare between the two traffic kinds, the coefficient of variation (CoV) for each flow's packet variances are computed (Figure 4). We observe that the inelastic flows have a significantly lower CoV than the elastic flows – representing the constant packet rates – as would expected from their streaming behaviors. For example, `WITS-04` has about 40% of inelastic flows having up to CoV one, while only about 4% of elastic flows having up to CoV one. Similiarly, `SITE-III-04` has about 60% of inelastic flows having up to CoV two, but only about 20% of elastic flows having up to CoV two.

Some networks mainly have those inelastic flows with low CoVs (e.g., `AUCK-03`, `SITE-III-04`). However, in more recent networks, the elastic2 flows have CoVs as low as the inelastic ones (e.g., `AUCK-09`, `ISP-B-07`, `CAIDA-DirA-09`). We also observe that the elastic1 flows have the largest CoV ranges.

These observations show some flow behaviors that seem to resemble streaming, and such traffic can be distinguished by our two measurements. However as far as the overall proportions are concerned, no trend of longer-lived or packet variances are observed between the different networks or between the years.

[3] We use port number assignments from [9], [11] to group the flows by three categories. We also exclude control/signaling port numbers, e.g., port 21 (FTP), 2979 (H.263), 5005 (RTP).

- *elastic1*: port 80 (HTTP) and 443 (HTTPS)
- *elastic2*: port 20 (FTP), 22 (SSH), 25 (SMTP), 110 (POP3) and 143 (IMAP)
- *inelastic*: port 322 (RTSPS), 537 (NMSP), 554 (RTSP), 1257 (Shockwave2), 1755 (MMS), 1790 (NMSP), 1935 (RTMP), 5004 (RTP), 6801 (Net2Phone), 6970-7170 (RealAudio), 7070 (RTSP), 8554 (RTSP-ALT) and 16384-16403 (iChat)

### B. Port Numbers

The rest of the plots in the Appendix show our observed port numbers. For example, each page shows three networks; Table III shows top10 most used port numbers, ranked according to their proportions for flows, volume and duration. It also shows a cumulated percentage of these top10 and top20 ports. In the middle (Figure 5), the port rank distributions are displayed as log-log plots. The left plots are the AUCK-99, center plots are the AUCK-03, and right plots are the AUCK-07 networks. The bottom (Figure 6) shows the cumulative distribution function (CDF) plot – the top two plots are for TCP, showing port numbers on a linear and a log scale respectively, and the bottom two plots are for UDP. The rest of the plots follow the same arrangement for other networks.

Overall, the top10 flows together contributed about 18% (ISP-B-05) to 60% (CAIDA-DirA-09) for TCP, and 9% (CAIDA-DirB-09) to 76% (SITE-I-03) for UDP. The ranges for the top10 volumes were greater, i.e., 33% (ISP-B-05) to 88% (AUCK-09) for TCP, and 11% (CAIDA-DirB-09) to 86% (BELL-I-02) for UDP. Using CoV metric across the networks, we find that TCP volume/flows and UDP volume/flows varied in ratio 0.22/0.25 and 0.45/0.42 respectively; UDP traffic is clearly more fluctuating. Again, we find little systematic trend for both TCP and UDP; those variabilities show that the traffic can either be heavily dominated by a few port numbers, or diversely dispersed. Various other well-known port numbers (up to 1023) also contributed to the top10. The individual port usages are less significantly contributed for higher ranks, e.g., top20 increases total percentages only slightly.

For TCP, we observe that HTTP/S traffic contributed the most and often appeared in the top rank. We also observe that generally recent networks have more high-end port numbers compared to the older networks. For UDP, DNS traffic were the most common, although rank distributions appear similar between the networks, we observe that the distributions are less skewed over the years, given that their volumes are already marginally small. Volumes on the port numbers are more diversely spread over the years, e.g., top10 volumes have reduced from 77% to 53% (WITS-04 to WITS-06), and only less than 17% of UDP volumes (CAIDA-DirA-09, CAIDA-DirB-09, ISP-B-07) are observed. These changes show that there are more applications using different port numbers in recent years. None of these ports however indicate any plausible evidence of incremental streaming traffic.

We observe how the port numbers are distributed by their attributes – number of flows and volume/duration. Measuring the volume for a particular port number is the same as measuring an aggregated flow size on that port number. Similarly, duration measures the total aggregated flow lifetimes of a given port number.

Here, we find that often up to 70% to 90% of port numbers used are below 10,000. The rest of the port usage appears quite uniformly distributed, although not strictly linear. A step in the CDF for one particular port number shows that this port is heavily used in the network being studied, e.g., FTP/SMTP and HTTP/S traffic, which is to be expected for well-known ports or registered ports. The registered ports are those from 1024 to 49151, so steps in the CDF are to be expected throughout this range. We do see this in several plots, for both UDP and TCP. We also see a roughly linear CDF for ports in the dynamic range above 49151, which is to be expected if they are chosen pseudo-randomly, as good security practice requires. The situation between 1024 and 49151 is somewhat confused, because many TCP/IP implementations appear to use arbitrary ranges between 1024 and 65535 for dynamic ports (often referred to as "ephemeral" ports, which is not a term defined in the TCP or UDP standards or in the IANA port allocations). It appears different Operating Systems, as well as their different versions, use a different range by default [10].

Both volume and duration distributions appear similar to the flow distribution, i.e., increase in the number of flows also increases total volume and durations. Some port numbers do not correlate equally with flows, volume and duration. For example, BELL-I-02 contained almost no flows on port 7331, but those flows carried more than 70% of volume and duration. Similarly, SITE-I-03 contained 0.4% of FTP data flows, but those contributed more than 43% of volume.

For older traces, a majority of protocols are low numbered, e.g., ISP-A-99 have more than 90% of traffic flows and volumes contributed to port number below 10,000, for both TCP and UDP. Conversely, recent traces have only up to about 50% (ISP-B-07). UDP traffic is a lot more linearly distributed across the port range, e.g., both CAIDA-DirB-09 and ISP-B-07. Also, DNS traffic volumes are no longer significant, e.g., contributing from 42% (ISP-A-99) to less than 2% (ISP-B-07). These changes appear to be the major differences between the older and newer traces, given that the volume ratios hardly changed.

## IV. Discussion

The UDP to TCP ratio does not seem to show any systematic trend; there are variations over time and between networks, but nothing we can identify as characteristic. In particular, there is nothing in the data to suggest a sustained growth in the share of UDP traffic caused by growth in audio and video streaming. Within TCP, we have seen some indication of streaming by well-known ports, e.g., those flows generally last much longer and have distinctly low variance of inter-packet arrival times.

Although we have observed a diversity of port numbers increasing over time, recent (2009) traffic volume appears to be aggregated on HTTP/S, and thus a prediction of increasing web traffic could be reasonable (e.g., [5]). It appears that a large number of application developers are taking advantage of and utilizing web traffic to increase interoperability through NATs and firewalls, mitigating deployment and operation issues [21], and in some cases to benefit from HTTP caching. From this, we may again observe the top port ranks contributing a lot more HTTP/S traffic, making the volume distributions similar to older network traffic.

It also appears that DNS traffic that was once a main contributor of UDP volume no longer stands out; instead UDP port numbers are more spread, presumably due to application diversities, possibly including streaming traffic. In fact, superficial evidence suggests that popular streaming solutions are at least as likely to use TCP (with or without HTTP) as they are to use UDP (with or without RTP). Our observations cannot directly detect this, but it is certain that we are not seeing a significant volume shift from TCP to UDP. Since streaming traffic is believed to be increasing, we must have an increase in the amount of TCP traffic for which TCP's response to congestion and loss (slowing down and retransmitting) is counter-productive.

In many cases, there are correlations of our three attributes, e.g., port 80 with a high proportion of flows is also likely to have a high proportion of both volume and duration. Similarly, an unpopular port number is likely to have low values for flows, volume and duration. However, certain ports with a low number of flows could contribute a high volume of traffic. Port usage trends are obviously dependent on application trends. As we have seen, these vary between networks, so local observations are the only valid guide. This could be significant if a service provider is planning to use any kind of address sharing by restricting the port range per subscriber [26]. There seems to be no general rule about which ports are popular, except for the few very well-known service ports.

Our observations of port usage also shows considerable but not systematic variation between networks. This is somewhat surprising; all the networks are large enough that we would expect usage patterns to average out and be similar in all cases. We can speculate that the demographics of the various user populations (e.g., students and academics versus general population) cause them to use rather different sets of operating systems and applications. However, the main lesson is that one cannot extrapolate from usage patterns on one network to those on another without allowing for at least as much variability as we have observed in this study.

From this, our observations also suggest several guidelines for potential measurements on operational networks. First, variation in the number of flows may indicate network instabilities and abnormal behaviors. The observed variability implies that one needs to be flexible when configuring the measurement parameters, e.g., the traffic meter's flow table size, perhaps adjusting the flow timeout differently for each port number. Second, the volume and duration of flows indicate potential network improvements based on port usages; in the port and rank distribution, the slopes indicate how the port numbers are concentrated in small or large ranges. This information can be considered for purposes such as prioritizing specific applications of interest, or new strategy in load balancing and accounting/billing. Flow-based routing (for example, [27]) has the ability to resolve integrity of inelastic (including VoIP and p2p) traffic by keeping track of flows for faster routing, though little evidence of applications has been reported.

## V. Related Work

Our observations share a similar view with measurement done in [1], i.e., a high UDP flow count and potential signaling flows. However, we detailed each network's traffic statistics, observed for per-flow behaviors of wide variabilities of the port number ranges, covering wider network traces. Our study extends the work in [23]; we included elastic and inelastic flow behaviors to observe potential streaming traffic, and all network summaries are detailed in this work. We note that port-based observations can give inaccurate protocol identification; however studies have shown (e.g., [20], [21]) that port numbers still give reasonable insights into applications and trends. Faber [14] suggested that IP hosts producing UDP flows could be characterized by weight functions, e.g., between p2p and scans. Also, McNutt and De Shon [24] have computed correlations in the usage of ephemeral ports to identify potential malicious traffic patterns. Wang *et al.* [29] reported on a short term study of the distribution of ephemeral port usage; they consider any port above 1024 to be ephemeral, not distinguishing between the registered and dynamic ports. Ephemeral port number cycling can be visualized so as to detect hidden services [16]. Allman [13] suggested different ways to select ephemeral ports that are more diverse and robust against security threats. Much interest in the choice of ephemeral port numbers was aroused by the DNS vulnerability publicized in 2008 [3]. It is to be expected that as developers learn the lesson of this vulnerability, randomization of port numbers may become more prevalent.

## VI. Conclusion

In this paper, we have have observed the two widely used protocols (UDP and TCP) to measure how their UDP to TCP ratio varied. Particularly we observed that there is no clear evidence that the ratio is increasing or decreasing. The ratio is rather dependent on application popularity and, consequently, on user choices. The volume ratio had subtle variations – the majority of volume is dominated by TCP, with a diurnal pattern. The flow ratio had larger variations – many flows are UDP but with very small volume.

Although the ratio does not vary systematically among the networks, each had quite different port number distributions. For example, data from recent years of ISP networks contained a large amount of p2p traffic, while enterprise networks contained a large amount of FTP traffic. Again, user choices are at work. Well-known streaming flows such as RTP, NMSP, and RTMP are visible especially in recent years, however there are no particular signs of incremental use of them.

As we note that emerging applications use arbitrary port numbers, identifying applications solely based on port numbers alone could lead to inaccurate assumptions; deep packet inspection may be the only approach in practice to determine the streaming traffic, provided that the packets are not encrypted. It could continue to be, on the other hand, that the streaming methods may simply further be evolved or integrated into elastic data traffic, provided that over-provisioning is widely practiced. Nevertheless, the trend towards more streaming traffic seems undeniable. However, contrary to what might naively be expected, there is no evidence of a resulting trend to relatively more use of UDP to carry it. In fact, the evidence is of widespread variability in the fraction of UDP traffic. Similarly, there is no clear trend in port usage, only evidence of widespread variability.

We had hoped to derive some general guidelines about the likely trend in traffic patterns, particularly concerning the fraction of non-congestion-controlled flows and the distribution of port usage. There appear to be no such guidelines in the available data. We consider that router and switch designers, as well as network operators, should be well aware of high variability in these basic characteristics, and design and provision their systems accordingly. In particular, one cannot extrapolate from measurements of one user population to the likely traffic patterns of another. It seems that all network operators need to measure their own protocol and port usage profiles.

## ACKNOWLEDGMENTS

Preliminary data on UDP to TCP ratios was kindly supplied by Arnold Nipper, Toshinori Ishii, Kjetil Olsen, Mike Hughes and Arne Oslebo. We are grateful to Ryan Koga of CAIDA and to Stanislav Shalunov, formerly of Internet2, for information about their respective datasets.

## REFERENCES

[1] "Analyzing UDP usage in Internet traffic," http://www.caida.org/research/traffic-analysis/tcpudpratio/, [Online; accessed 01-Aug-2009].

[2] "CAIDA Internet Data – Realtime Monitors," http://www.caida.org/data/realtime/index.xml, [Online; accessed 30-Oct-2009].

[3] "CERT Vulnerability Note VU#800113," http://www.kb.cert.org/vuls/id/800113/, [Online; accessed 10-Jun-2009].

[4] "Cisco Visual Networking Index: Usage Study," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/Cisco_VNI_Usage_WP.pdf, [Online; accessed 20-August-2009].

[5] "Hyperconnectivity and the Approaching Zettabyte Era," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf, [Online; accessed 20-August-2009].

[6] "Internet2 NetFlow: Weekly Reports," http://netflow.internet2.edu/weekly/, [Online; accessed 30-Oct-2009].

[7] "Minnesota Internet Traffic Studies (MINTS)," http://www.dtc.umn.edu/mints/home.php, [Online; accessed 20-August-2009].

[8] "Passive Measurement and Analysis (PMA)," http://pma.nlanr.net/, [Online; accessed 01-Nov-2008].

[9] "Port Numbers," http://www.iana.org/assignments/port-numbers, [Online; accessed 30-October-2009].

[10] "The Ephemeral Port Range," http://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html, [Online; accessed 30-October-2009].

[11] ""Well Known" TCP and UDP ports used by Apple software products," http://support.apple.com/kb/ts1629.

[12] "WITS: Waikato Internet Traffic Storage," http://www.wand.net.nz/wits/, [Online; accessed 01-Nov-2008].

[13] M. Allman, "Comments on selecting ephemeral ports," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 2, pp. 13–19, 2009.

[14] S. Faber, "Is there any value in bulk network traces?" *FloCon*, 2009.

[15] P. Gill, M. F. Arlitt, Z. Li, and A. Mahanti, "The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse?" in *PAM*, 2008, pp. 1–10.

[16] J. Janies, "Existence plots: A low-resolution time series for port behavior analysis," in *VizSec '08: Proceedings of the 5th international workshop on Visualization for Computer Security*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 161–168.

[17] Joe St Sauver, University of Oregon, "Personal communication," 2008.

[18] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2007, pp. 111–116.

[19] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos, "Is p2p dying or just hiding?" in *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 2004, pp. 1532–1538.

[20] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," in *CONEXT '08: Proceedings of the 2008 ACM CoNEXT Conference*. New York, NY, USA: ACM, 2008, pp. 1–12.

[21] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, F. Jahanian, and M. Karir, "2009 Internet Observatory Report," http://www.nanog.org/meetings/nanog47/presentations/Monday/Labovitz_ObserveReport_N47_Mon.pdf, 2009.

[22] C. Labovitz, D. McPherson, S. Iekel-Johnson, and M. Hollyman, "Internet Traffic Trends," http://www.nanog.org/meetings/nanog43/presentations/Labovitz_internetstats_N43.pdf, 2008.

[23] D. Lee, B. E. Carpenter, and N. Brownlee, "Observations of UDP to TCP Ratio and Port Numbers," in *Fifth International Conference on Internet Monitoring and Protection (ICIMP)*. IEEE, 2010, pp. 99–104.

[24] J. McNutt and M. D. Shon, "Correlations between quiescent ports in network flows," *FloCon*, 2005.

[25] A. Odlyzko, "The Delusions of Net Neutrality," in *Telecommunications Policy Research Conference*, 2008.

[26] R. Bush (ed.), "The A+P Approach to the IPv4 Address Shortage (work in progress)," http://tools.ietf.org/id/draft-ymbk-aplusp, 2009.

[27] L. Roberts, "A radical new router," *Spectrum, IEEE*, vol. 46, no. 7, pp. 34–39, July 2009.

[28] J. Rosenberg, "UDP and TCP as the New Waist of the Internet Hourglass," http://tools.ietf.org/id/draft-rosenberg-internet-waist-hourglass-00.txt.

[29] H. Wang, R. Zhou, and Y. He, "An Information Acquisition Method Based on NetFlow for Network Situation Awareness," *Advanced Software Engineering and Its Applications*, pp. 23–26, 2008.

## APPENDIX

### PLOTS

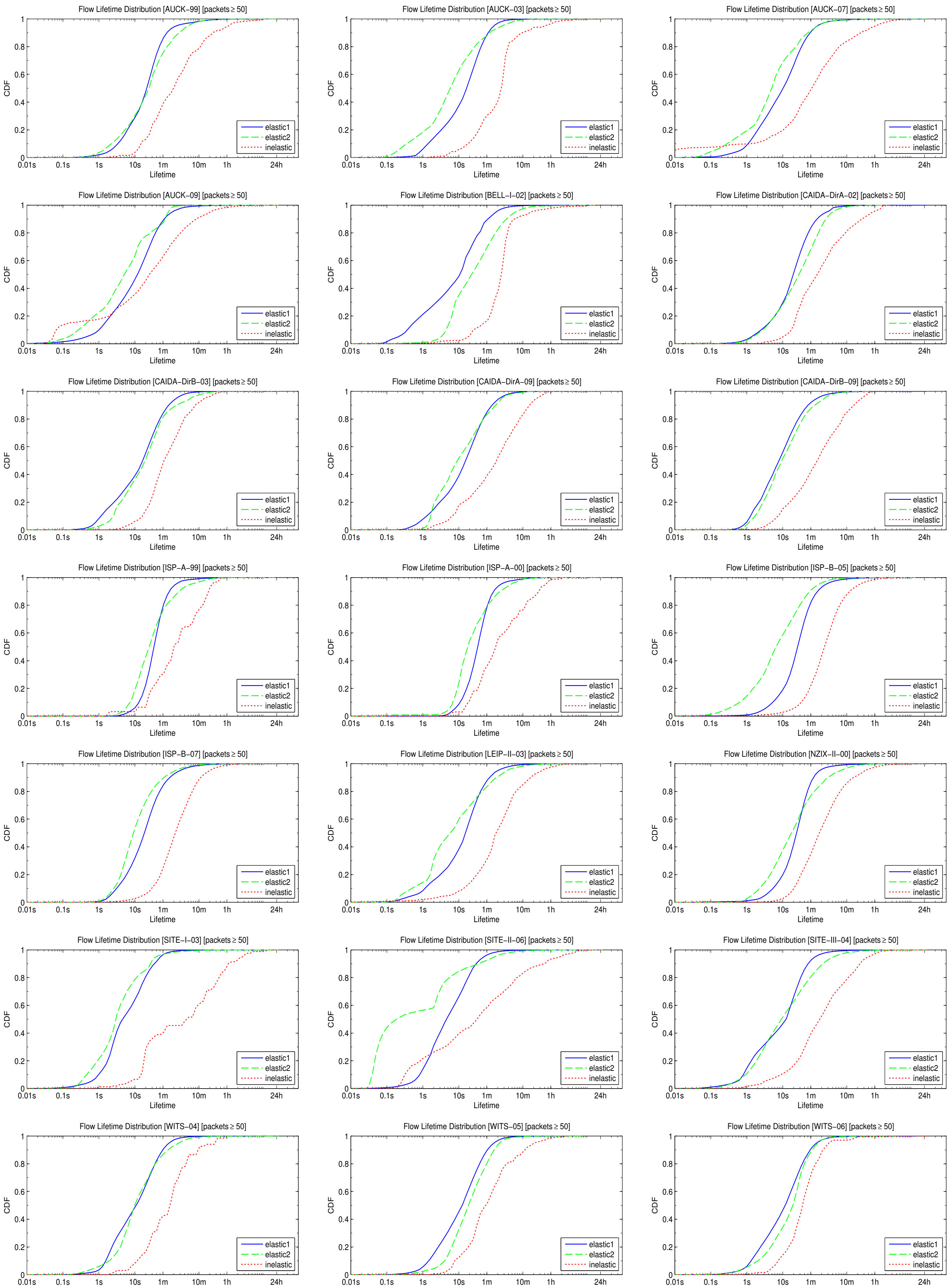



Fig. 3. Flow Lifetime Distribution – showing three flow kinds carrying at least 50 packets

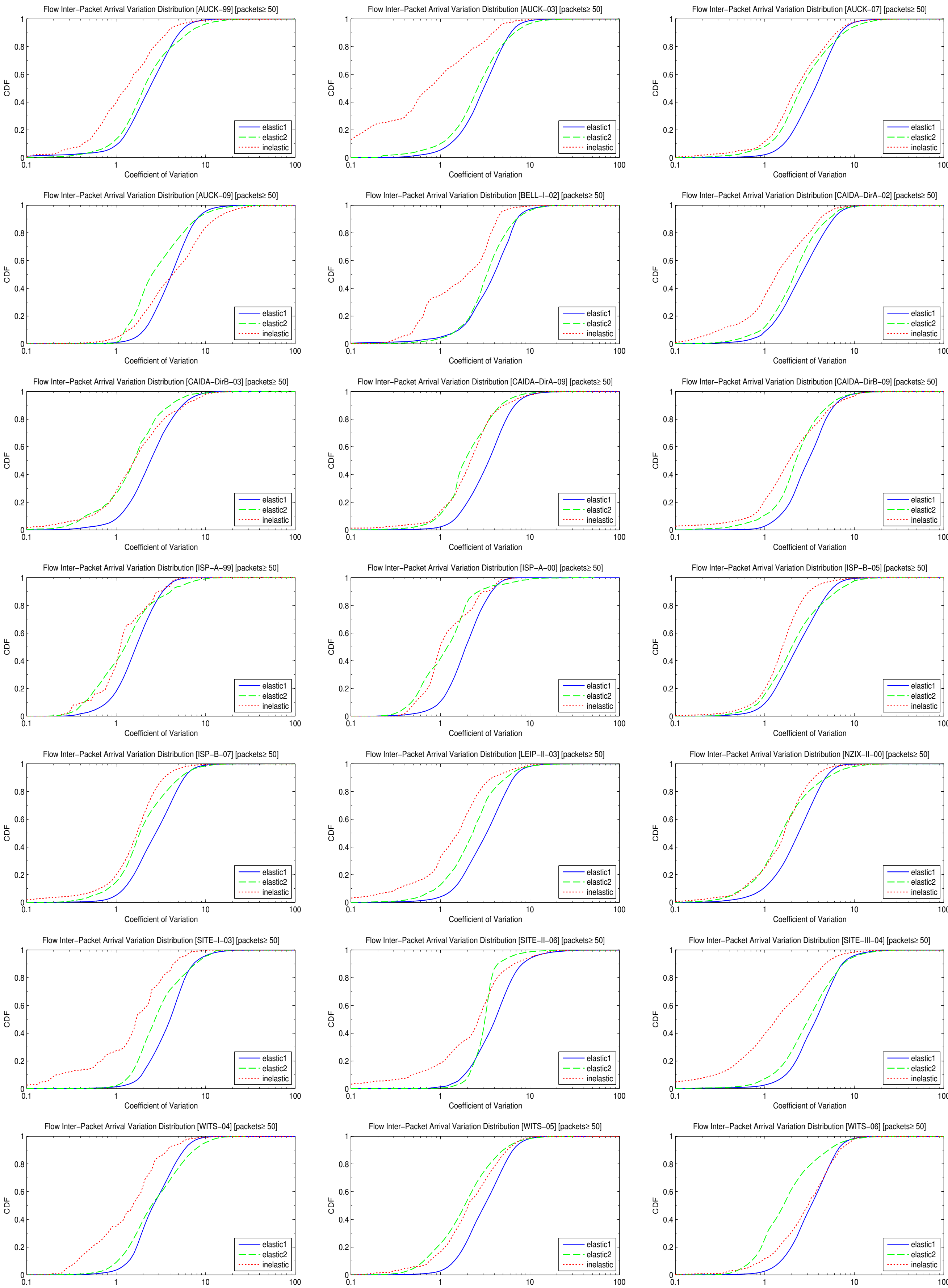



Fig. 4. Flow Inter Packet Arrival Variation Distribution – showing three flow kinds carrying at least 50 packets

TABLE III
TOP10 PORT USAGE – LEFT:AUCK−99, CENTER:AUCK−03, RIGHT:AUCK−07

| AUCK−99−TCP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Lifetime | |
| Port# | % | Port# | % | Port# | % |
| 80 | 38.57 | 80 | 60.06 | 80 | 30.53 |
| 113 | 2.16 | 83 | 2.52 | 25 | 3.21 |
| 25 | 2.10 | 20 | 1.03 | 83 | 2.89 |
| 83 | 1.14 | 40221 | 0.88 | 119 | 1.46 |
| 443 | 0.67 | 40220 | 0.87 | 22 | 1.07 |
| 8080 | 0.62 | 40219 | 0.86 | 6665 | 0.62 |
| 110 | 0.40 | 52179 | 0.71 | 443 | 0.56 |
| 22 | 0.27 | 52180 | 0.71 | 21 | 0.48 |
| 21 | 0.19 | 52178 | 0.70 | 20 | 0.48 |
| 8001 | 0.18 | 2013 | 0.68 | 23 | 0.47 |
| Top10 | 46.29 | Top10 | 69.03 | Top10 | 41.78 |
| Top20 | 47.35 | Top20 | 72.63 | Top20 | 44.74 |

| AUCK−03−TCP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Lifetime | |
| Port# | % | Port# | % | Port# | % |
| 80 | 18.83 | 80 | 59.26 | 80 | 21.50 |
| 25 | 4.18 | 443 | 10.26 | 443 | 6.41 |
| 443 | 3.77 | 119 | 3.21 | 9050 | 3.09 |
| 2703 | 0.88 | 20 | 0.62 | 25 | 2.90 |
| 1863 | 0.87 | 1755 | 0.59 | 7000 | 1.02 |
| 9050 | 0.37 | 25 | 0.45 | 1863 | 0.89 |
| 1080 | 0.32 | 873 | 0.38 | 5190 | 0.67 |
| 7000 | 0.27 | 993 | 0.34 | 13130 | 0.49 |
| 20349 | 0.26 | 8000 | 0.30 | 119 | 0.43 |
| 1025 | 0.23 | 22 | 0.27 | 2703 | 0.26 |
| Top10 | 29.98 | Top10 | 75.69 | Top10 | 37.65 |
| Top20 | 31.21 | Top20 | 77.35 | Top20 | 39.09 |

| AUCK−07−TCP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Lifetime | |
| Port# | % | Port# | % | Port# | % |
| 80 | 29.80 | 80 | 54.02 | 80 | 26.76 |
| 443 | 7.49 | 443 | 4.35 | 25 | 6.92 |
| 25 | 6.33 | 554 | 1.21 | 443 | 4.11 |
| 2703 | 1.21 | 873 | 1.21 | 1863 | 0.86 |
| 1863 | 0.69 | 20 | 0.51 | 5222 | 0.39 |
| 6000 | 0.49 | 3355 | 0.46 | 5190 | 0.33 |
| 993 | 0.35 | 3389 | 0.38 | 993 | 0.21 |
| 1080 | 0.20 | 3202 | 0.35 | 61 | 0.20 |
| 21 | 0.12 | 25 | 0.33 | 554 | 0.20 |
| 143 | 0.08 | 1935 | 0.29 | 2848 | 0.17 |
| Top10 | 46.75 | Top10 | 63.11 | Top10 | 40.18 |
| Top20 | 47.30 | Top20 | 64.75 | Top20 | 41.23 |

| AUCK−99−UDP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 53 | 36.00 | 27532 | 16.96 | 443 | 16.70 |
| 1099 | 16.06 | 2926 | 15.69 | 53 | 14.77 |
| 123 | 7.96 | 3130 | 12.12 | 3130 | 13.10 |
| 4000 | 4.66 | 53 | 11.96 | 40657 | 4.08 |
| 1024 | 3.52 | 16232 | 3.99 | 2809 | 2.46 |
| 40657 | 1.26 | 5010 | 2.22 | 36497 | 1.66 |
| 3130 | 1.21 | 16187 | 2.00 | 4000 | 1.51 |
| 137 | 0.79 | 17106 | 1.81 | 1024 | 1.40 |
| 443 | 0.48 | 1363 | 1.81 | 6980 | 1.19 |
| 36497 | 0.40 | 14684 | 1.67 | 6978 | 1.16 |
| Top10 | 72.35 | Top10 | 70.24 | Top10 | 58.03 |
| Top20 | 75.24 | Top20 | 80.21 | Top20 | 66.79 |

| AUCK−03−UDP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 53 | 34.70 | 53 | 20.34 | 53 | 38.73 |
| 32769 | 10.95 | 49188 | 6.92 | 32769 | 30.05 |
| 6277 | 3.02 | 49212 | 5.57 | 50524 | 4.36 |
| 1026 | 2.71 | 5004 | 5.19 | 35546 | 4.03 |
| 1025 | 2.66 | 32769 | 3.88 | 32786 | 2.34 |
| 50524 | 2.43 | 49180 | 2.61 | 12345 | 1.79 |
| 35546 | 2.32 | 49210 | 2.33 | 12371 | 1.78 |
| 1027 | 2.17 | 49186 | 2.31 | 50342 | 0.96 |
| 1028 | 2.03 | 49204 | 2.28 | 51024 | 0.90 |
| 1029 | 1.54 | 10000 | 1.91 | 51835 | 0.88 |
| Top10 | 64.55 | Top10 | 53.35 | Top10 | 85.83 |
| Top20 | 73.98 | Top20 | 67.76 | Top20 | 90.63 |

| AUCK−07−UDP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 53 | 43.43 | 53 | 20.27 | 53 | 27.58 |
| 24051 | 1.96 | 35026 | 9.77 | 32776 | 11.76 |
| 32776 | 1.27 | 60264 | 6.24 | 32782 | 11.04 |
| 32782 | 1.23 | 60010 | 5.90 | 24051 | 4.77 |
| 24405 | 0.68 | 46015 | 5.25 | 46015 | 3.37 |
| 123 | 0.18 | 60018 | 4.72 | 123 | 2.28 |
| 2976 | 0.15 | 51452 | 2.66 | 6277 | 1.59 |
| 13326 | 0.12 | 59004 | 2.23 | 443 | 1.13 |
| 1096 | 0.12 | 1996 | 1.72 | 11113 | 1.04 |
| 17200 | 0.11 | 10000 | 1.62 | 24405 | 0.78 |
| Top10 | 49.24 | Top10 | 60.39 | Top10 | 65.34 |
| Top20 | 50.08 | Top20 | 67.38 | Top20 | 71.12 |

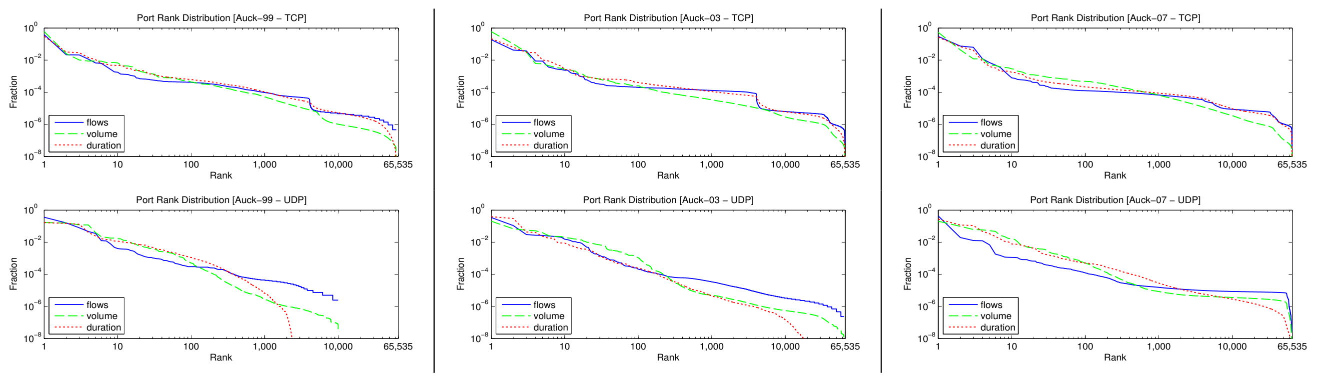



Fig. 5. Port Rank Distribution – Left:AUCK−99, Center:AUCK−03, Right:AUCK−07

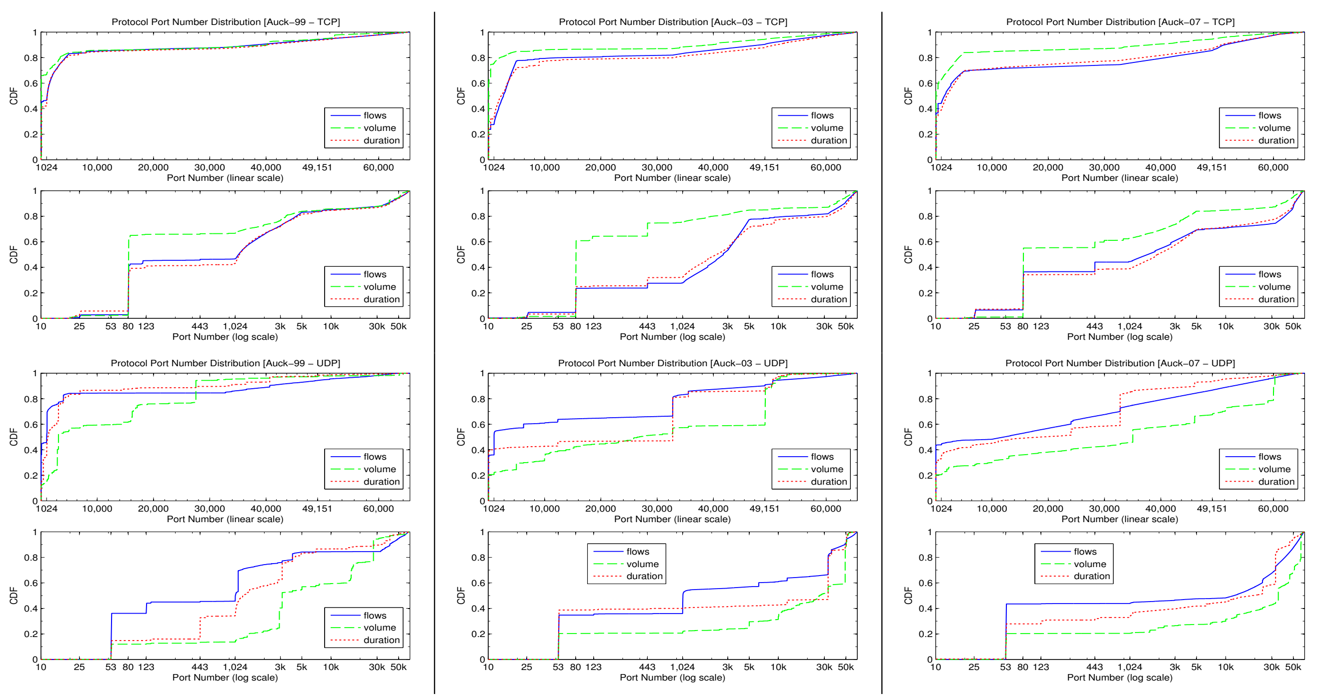



Fig. 6. Port Number Distribution – Left:AUCK−99, Center:AUCK−03, Right:AUCK−07

TABLE IV
TOP10 PORT USAGE – LEFT: AUCK-09, CENTER: BELL-I-02, RIGHT: CAIDA-DirA-02

| AUCK-09-TCP | | | | | | BELL-I-02-TCP | | | | | | CAIDA-DirA-02-TCP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 80 | 34.89 | 80 | 70.41 | 80 | 28.19 | 80 | 28.35 | 119 | 32.28 | 80 | 17.88 | 80 | 39.23 | 80 | 65.27 | 80 | 33.92 |
| 443 | 5.32 | 3131 | 5.99 | 443 | 7.43 | 2000 | 2.38 | 80 | 28.12 | 711 | 3.37 | 25 | 2.68 | 1755 | 3.02 | 25 | 2.40 |
| 3128 | 3.14 | 443 | 4.13 | 3128 | 6.34 | 443 | 2.04 | 6677 | 2.59 | 22 | 3.31 | 21 | 2.65 | 4662 | 2.37 | 4662 | 1.73 |
| 3131 | 1.38 | 3128 | 3.86 | 3131 | 1.95 | 25 | 1.57 | 564 | 2.45 | 25 | 1.77 | 8080 | 0.59 | 1214 | 1.90 | 8010 | 1.69 |
| 25 | 1.03 | 554 | 2.02 | 25 | 1.02 | 5190 | 1.34 | 10986 | 1.41 | 564 | 1.36 | 4662 | 0.42 | 6699 | 1.27 | 1214 | 1.62 |
| 1863 | 0.45 | 1935 | 1.08 | 1863 | 0.42 | 21 | 1.31 | 22 | 1.29 | 21 | 1.25 | 53 | 0.30 | 2189 | 0.63 | 6699 | 1.43 |
| 6000 | 0.37 | 993 | 0.31 | 10000 | 0.15 | 22 | 0.99 | 554 | 1.20 | 6346 | 1.20 | 1214 | 0.29 | 6346 | 0.60 | 6667 | 1.17 |
| 2703 | 0.20 | 873 | 0.30 | 554 | 0.15 | 711 | 0.89 | 443 | 1.20 | 11021 | 1.17 | 110 | 0.29 | 2401 | 0.47 | 1755 | 0.83 |
| 9050 | 0.20 | 22 | 0.17 | 5222 | 0.15 | 1863 | 0.32 | 1755 | 1.02 | 443 | 1.07 | 1863 | 0.27 | 8080 | 0.41 | 21 | 0.76 |
| 993 | 0.13 | 8002 | 0.11 | 993 | 0.11 | 5050 | 0.16 | 55418 | 0.98 | 5190 | 0.86 | 6667 | 0.21 | 119 | 0.33 | 8080 | 0.54 |
| Top10 | 47.11 | Top10 | 88.38 | Top10 | 45.91 | Top10 | 39.35 | Top10 | 72.55 | Top10 | 33.24 | Top10 | 46.93 | Top10 | 76.28 | Top10 | 46.09 |
| Top20 | 47.77 | Top20 | 89.19 | Top20 | 46.41 | Top20 | 40.23 | Top20 | 79.05 | Top20 | 38.52 | Top20 | 47.96 | Top20 | 78.73 | Top20 | 48.78 |

| AUCK-09-UDP | | | | | | BELL-I-02-UDP | | | | | | CAIDA-DirA-02-UDP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 53 | 43.76 | 33001 | 24.69 | 1513 | 11.84 | 137 | 21.41 | 7331 | 72.10 | 7331 | 70.43 | 53 | 37.76 | 1052 | 16.02 | 53 | 32.43 |
| 1513 | 0.92 | 33670 | 19.91 | 49153 | 7.13 | 53 | 3.87 | 33264 | 2.79 | 55 | 4.39 | 6257 | 18.13 | 1047 | 15.67 | 6257 | 5.83 |
| 123 | 0.63 | 38168 | 7.91 | 10002 | 4.25 | 123 | 3.33 | 161 | 2.57 | 53 | 3.86 | 1214 | 4.02 | 53 | 6.07 | 28800 | 5.32 |
| 14398 | 0.17 | 59002 | 5.34 | 10003 | 4.12 | 32532 | 2.37 | 24716 | 2.22 | 137 | 3.35 | 27243 | 2.12 | 6257 | 4.47 | 5555 | 4.22 |
| 17822 | 0.16 | 16402 | 4.58 | 53 | 3.35 | 500 | 1.35 | 53 | 1.59 | 8482 | 1.18 | 123 | 1.28 | 1716 | 2.64 | 27243 | 3.42 |
| 10306 | 0.15 | 53 | 3.55 | 49154 | 2.07 | 24503 | 1.31 | 24504 | 1.17 | 6899 | 1.11 | 5555 | 0.90 | 12203 | 2.01 | 2002 | 1.86 |
| 36589 | 0.10 | 59004 | 1.96 | 46015 | 1.97 | 27732 | 1.18 | 22888 | 1.06 | 24503 | 0.79 | 137 | 0.88 | 27015 | 1.43 | 137 | 1.59 |
| 51504 | 0.10 | 5442 | 1.89 | 443 | 1.76 | 6899 | 1.18 | 6899 | 1.01 | 14137 | 0.73 | 27005 | 0.86 | 6112 | 0.84 | 1214 | 1.55 |
| 2535 | 0.08 | 65321 | 1.58 | 1684 | 1.68 | 55 | 1.14 | 7170 | 0.85 | 24721 | 0.63 | 27015 | 0.86 | 4708 | 0.79 | 12345 | 1.41 |
| 41048 | 0.08 | 1044 | 1.00 | 3128 | 1.44 | 28753 | 1.02 | 137 | 0.81 | 27161 | 0.63 | 1717 | 0.64 | 49606 | 0.62 | 6112 | 1.24 |
| Top10 | 46.15 | Top10 | 72.42 | Top10 | 39.60 | Top10 | 38.15 | Top10 | 86.18 | Top10 | 87.10 | Top10 | 67.45 | Top10 | 50.55 | Top10 | 58.86 |
| Top20 | 46.74 | Top20 | 79.54 | Top20 | 48.14 | Top20 | 46.33 | Top20 | 91.18 | Top20 | 91.32 | Top20 | 71.10 | Top20 | 54.29 | Top20 | 64.25 |

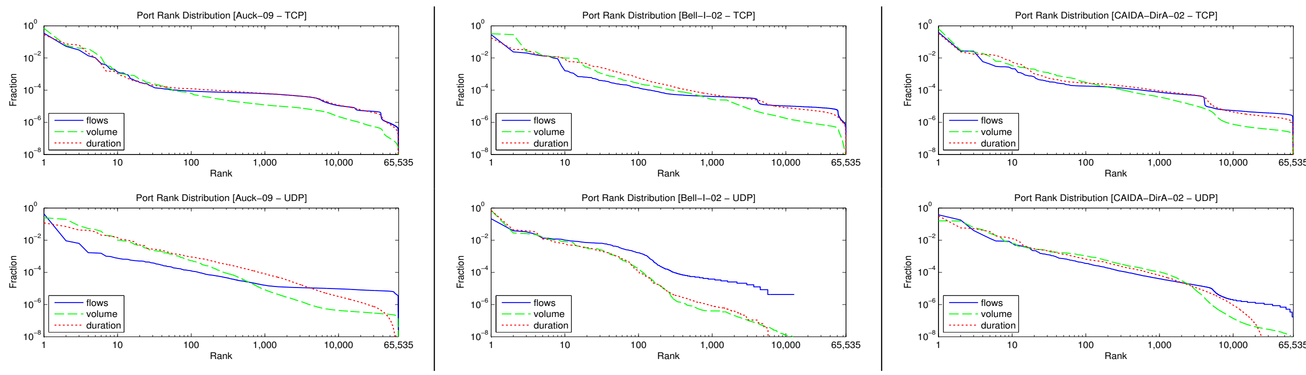



Fig. 7. Port Rank Distribution – Left: AUCK-09, Center: BELL-I-02, Right: CAIDA-DirA-02

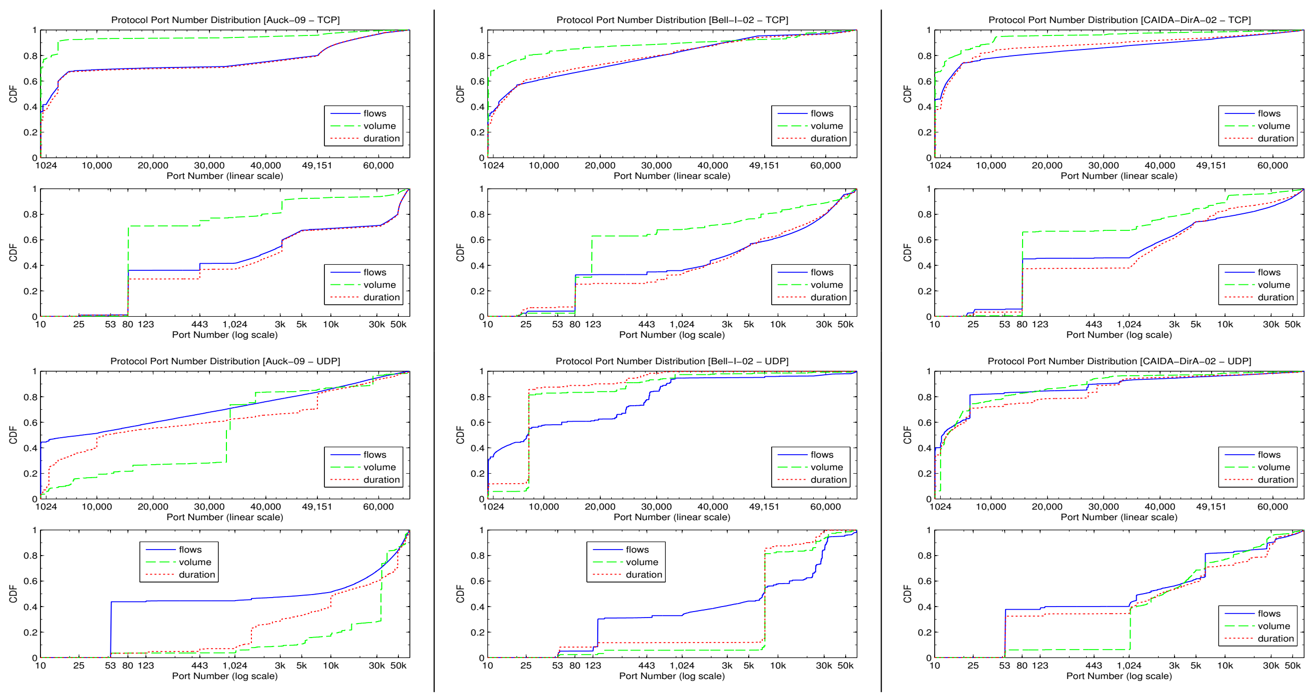



Fig. 8. Port Number Distribution – Left: AUCK-09, Center: BELL-I-02, Right: CAIDA-DirA-02

TABLE V
TOP10 PORT USAGE – LEFT:CAIDA-DirB-03, CENTER:CAIDA-DirA-09, RIGHT:CAIDA-DirB-09

| CAIDA-DirB-03-TCP | | | | | | CAIDA-DirA-09-TCP | | | | | | CAIDA-DirB-09-TCP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 80 | 28.02 | 80 | 72.69 | 80 | 22.84 | 80 | 35.58 | 20 | 42.07 | 80 | 25.33 | 80 | 24.41 | 80 | 65.58 | 80 | 15.61 |
| 1080 | 2.55 | 4662 | 1.39 | 4662 | 3.62 | 25 | 15.84 | 80 | 41.41 | 25 | 6.49 | 25 | 2.40 | 443 | 1.18 | 9050 | 5.56 |
| 4662 | 0.96 | 443 | 1.12 | 25 | 1.39 | 443 | 6.38 | 443 | 1.87 | 9050 | 5.57 | 9050 | 2.04 | 554 | 0.98 | 25 | 1.68 |
| 81 | 0.88 | 6699 | 1.01 | 1080 | 1.24 | 9050 | 1.43 | 9050 | 0.63 | 443 | 3.96 | 443 | 1.19 | 9050 | 0.84 | 443 | 1.17 |
| 25 | 0.77 | 81 | 0.84 | 6699 | 0.68 | 22 | 0.19 | 25 | 0.56 | 6881 | 0.32 | 2710 | 0.45 | 81 | 0.39 | 6881 | 0.35 |
| 889 | 0.60 | 88 | 0.83 | 139 | 0.67 | 23 | 0.14 | 1935 | 0.14 | 28805 | 0.27 | 445 | 0.34 | 1935 | 0.36 | 21 | 0.21 |
| 49555 | 0.37 | 8080 | 0.68 | 6667 | 0.60 | 21 | 0.11 | 110 | 0.10 | 51413 | 0.17 | 6667 | 0.32 | 35627 | 0.19 | 6346 | 0.20 |
| 10002 | 0.34 | 1214 | 0.63 | 1214 | 0.59 | 11762 | 0.11 | 6881 | 0.10 | 13130 | 0.16 | 22 | 0.22 | 51413 | 0.13 | 2710 | 0.20 |
| 6588 | 0.34 | 7675 | 0.47 | 81 | 0.55 | 445 | 0.11 | 554 | 0.07 | 45682 | 0.13 | 11762 | 0.19 | 5001 | 0.11 | 51413 | 0.19 |
| 179 | 0.29 | 1755 | 0.41 | 49555 | 0.47 | 1755 | 0.10 | 19101 | 0.06 | 6346 | 0.11 | 21 | 0.17 | 52815 | 0.11 | 17326 | 0.19 |
| Top10 | 35.13 | Top10 | 80.07 | Top10 | 32.64 | Top10 | 60.00 | Top10 | 86.99 | Top10 | 42.52 | Top10 | 31.72 | Top10 | 69.87 | Top10 | 25.37 |
| Top20 | 36.82 | Top20 | 81.71 | Top20 | 35.28 | Top20 | 60.51 | Top20 | 87.48 | Top20 | 43.18 | Top20 | 32.76 | Top20 | 70.78 | Top20 | 26.17 |
| CAIDA-DirB-03-UDP | | | | | | CAIDA-DirA-09-UDP | | | | | | CAIDA-DirB-09-UDP | | | | | |
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 22321 | 21.30 | 14567 | 11.76 | 53 | 17.62 | 53 | 11.61 | 53 | 6.70 | 53 | 7.40 | 53 | 6.88 | 57722 | 2.56 | 57722 | 11.20 |
| 53 | 11.73 | 27005 | 6.98 | 22321 | 8.42 | 123 | 0.74 | 25175 | 1.56 | 3074 | 0.71 | 6881 | 0.61 | 53 | 1.88 | 53 | 1.95 |
| 7674 | 11.15 | 554 | 6.05 | 6257 | 4.36 | 6881 | 0.39 | 161 | 1.47 | 6881 | 0.62 | 6257 | 0.30 | 60096 | 1.32 | 6881 | 0.72 |
| 6257 | 3.21 | 53 | 5.37 | 7674 | 3.45 | 50000 | 0.17 | 5150 | 1.15 | 500 | 0.48 | 6346 | 0.20 | 3074 | 1.25 | 6257 | 0.58 |
| 1026 | 1.55 | 27010 | 3.45 | 1024 | 1.41 | 49152 | 0.16 | 22209 | 1.10 | 10000 | 0.40 | 45682 | 0.17 | 15000 | 1.22 | 3074 | 0.38 |
| 1027 | 1.54 | 1247 | 2.15 | 6112 | 1.26 | 6346 | 0.15 | 3074 | 0.87 | 6348 | 0.36 | 60001 | 0.16 | 49262 | 0.98 | 10000 | 0.30 |
| 1025 | 1.53 | 6257 | 2.05 | 28800 | 1.25 | 65535 | 0.13 | 64065 | 0.84 | 6346 | 0.32 | 32768 | 0.09 | 5004 | 0.56 | 6346 | 0.27 |
| 1029 | 1.27 | 12203 | 1.49 | 27005 | 1.04 | 16001 | 0.13 | 15000 | 0.67 | 10001 | 0.24 | 50000 | 0.08 | 18350 | 0.47 | 60001 | 0.24 |
| 1028 | 1.04 | 27015 | 1.23 | 3601 | 0.95 | 10000 | 0.11 | 60023 | 0.65 | 32768 | 0.22 | 20129 | 0.08 | 4500 | 0.46 | 15000 | 0.22 |
| 137 | 0.87 | 6112 | 1.22 | 5325 | 0.95 | 6800 | 0.11 | 7566 | 0.54 | 123 | 0.18 | 60000 | 0.07 | 1044 | 0.46 | 500 | 0.16 |
| Top10 | 55.19 | Top10 | 41.75 | Top10 | 40.73 | Top10 | 13.71 | Top10 | 15.54 | Top10 | 10.91 | Top10 | 8.64 | Top10 | 11.16 | Top10 | 16.02 |
| Top20 | 60.46 | Top20 | 49.36 | Top20 | 46.65 | Top20 | 14.49 | Top20 | 19.92 | Top20 | 12.11 | Top20 | 9.16 | Top20 | 13.98 | Top20 | 17.02 |

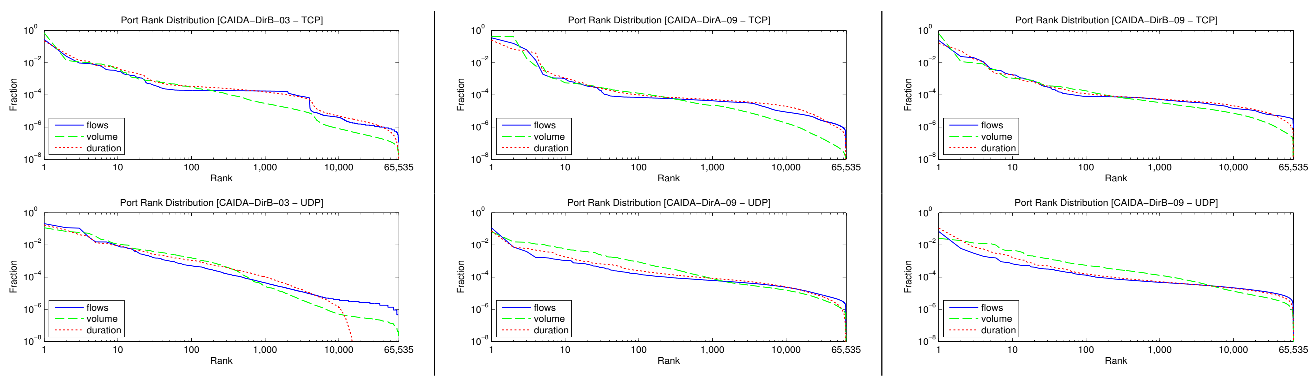



Fig. 9. Port Rank Distribution – Left:CAIDA-DirB-03, Center:CAIDA-DirA-09, Right:CAIDA-DirB-09

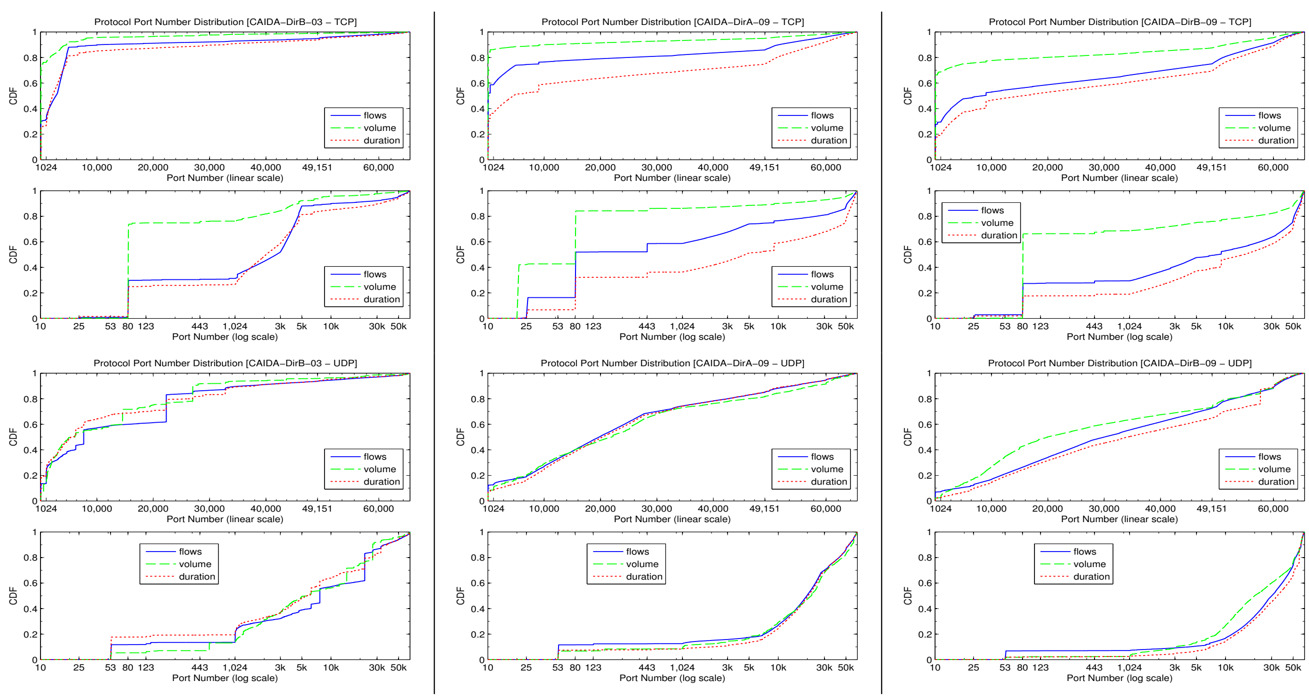



Fig. 10. Port Number Distribution – Left:CAIDA-DirB-03, Center:CAIDA-DirA-09, Right:CAIDA-DirB-09

TABLE VI
TOP10 PORT USAGE – LEFT:ISP-A-99, CENTER:ISP-A-00, RIGHT:ISP-B-05

| ISP-A-99-TCP | | | | | | ISP-A-00-TCP | | | | | | ISP-B-05-TCP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 80 | 33.48 | 80 | 38.12 | 80 | 25.96 | 80 | 36.21 | 80 | 44.30 | 80 | 24.99 | 80 | 6.90 | 80 | 16.17 | 6881 | 4.73 |
| 25 | 3.57 | 1040 | 15.96 | 25 | 3.21 | 110 | 2.79 | 1040 | 27.09 | 1040 | 3.52 | 4662 | 3.46 | 4662 | 4.98 | 80 | 4.11 |
| 110 | 2.97 | 110 | 10.69 | 6699 | 2.74 | 25 | 2.42 | 110 | 4.07 | 6699 | 2.62 | 6881 | 2.30 | 6881 | 3.22 | 4662 | 4.04 |
| 113 | 2.88 | 6699 | 7.10 | 6667 | 2.34 | 113 | 1.63 | 6699 | 2.68 | 6667 | 2.53 | 6346 | 1.43 | 6346 | 2.93 | 6346 | 3.31 |
| 6667 | 1.91 | 119 | 1.11 | 1040 | 1.99 | 6667 | 1.00 | 2117 | 1.25 | 25 | 1.86 | 25 | 1.18 | 8000 | 1.63 | 16881 | 1.00 |
| 443 | 0.53 | 20 | 1.09 | 110 | 1.63 | 443 | 0.45 | 119 | 0.86 | 4901 | 1.16 | 445 | 0.84 | 6699 | 1.15 | 6699 | 0.79 |
| 1863 | 0.28 | 25 | 0.64 | 4901 | 1.17 | 23 | 0.32 | 6700 | 0.66 | 6666 | 1.14 | 1863 | 0.76 | 119 | 0.88 | 6348 | 0.66 |
| 8888 | 0.27 | 53358 | 0.57 | 2222 | 0.58 | 20 | 0.29 | 20 | 0.52 | 1374 | 1.09 | 16881 | 0.57 | 110 | 0.77 | 6882 | 0.63 |
| 81 | 0.25 | 23 | 0.38 | 1533 | 0.53 | 24554 | 0.27 | 81 | 0.50 | 110 | 0.88 | 110 | 0.56 | 6348 | 0.74 | 25 | 0.50 |
| 1032 | 0.25 | 2660 | 0.38 | 1073 | 0.44 | 13628 | 0.27 | 23 | 0.36 | 6668 | 0.82 | 135 | 0.38 | 16881 | 0.56 | 1863 | 0.48 |
| Top10 | 46.40 | Top10 | 76.03 | Top10 | 40.57 | Top10 | 45.64 | Top10 | 82.29 | Top10 | 40.61 | Top10 | 18.37 | Top10 | 33.04 | Top10 | 20.25 |
| Top20 | 48.56 | Top20 | 78.27 | Top20 | 44.20 | Top20 | 47.51 | Top20 | 84.59 | Top20 | 45.10 | Top20 | 20.36 | Top20 | 36.13 | Top20 | 22.90 |

| ISP-A-99-UDP | | | | | | ISP-A-00-UDP | | | | | | ISP-B-05-UDP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 53 | 54.50 | 53 | 42.05 | 53 | 46.89 | 53 | 53.74 | 28001 | 14.24 | 53 | 28.32 | 4672 | 21.29 | 6346 | 8.59 | 6346 | 19.07 |
| 4000 | 3.30 | 1533 | 4.86 | 1646 | 12.24 | 4000 | 2.25 | 53 | 12.73 | 138 | 9.94 | 6881 | 8.14 | 6348 | 3.66 | 53 | 5.89 |
| 137 | 2.27 | 3328 | 4.65 | 4000 | 7.95 | 137 | 1.80 | 1080 | 7.95 | 1646 | 7.34 | 53 | 6.79 | 7000 | 2.51 | 6881 | 3.00 |
| 1646 | 1.16 | 3635 | 3.97 | 1645 | 5.08 | 138 | 1.79 | 7877 | 7.65 | 4000 | 6.04 | 6346 | 3.95 | 4672 | 2.48 | 4672 | 2.90 |
| 1645 | 1.01 | 3225 | 3.19 | 28800 | 1.91 | 1646 | 1.15 | 7777 | 5.72 | 6112 | 3.24 | 6257 | 1.46 | 53 | 2.37 | 32770 | 1.81 |
| 138 | 0.82 | 137 | 2.85 | 137 | 1.76 | 7778 | 0.94 | 1037 | 4.38 | 1645 | 2.53 | 123 | 0.98 | 16881 | 2.19 | 8000 | 1.68 |
| 1026 | 0.75 | 6112 | 2.70 | 6112 | 1.29 | 1645 | 0.91 | 27960 | 4.06 | 1080 | 2.03 | 1083 | 0.71 | 27005 | 1.87 | 6257 | 1.24 |
| 4936 | 0.52 | 1646 | 2.30 | 1026 | 0.93 | 1026 | 0.44 | 6112 | 3.48 | 4200 | 1.99 | 6190 | 0.70 | 27016 | 1.50 | 123 | 0.91 |
| 1025 | 0.49 | 3370 | 2.26 | 1533 | 0.92 | 6112 | 0.39 | 49608 | 2.58 | 28001 | 1.82 | 32770 | 0.68 | 6881 | 1.27 | 28800 | 0.82 |
| 123 | 0.43 | 4000 | 1.72 | 1025 | 0.68 | 1025 | 0.35 | 138 | 2.57 | 1037 | 1.78 | 1087 | 0.52 | 6257 | 1.13 | 4000 | 0.78 |
| Top10 | 65.25 | Top10 | 70.53 | Top10 | 79.66 | Top10 | 63.77 | Top10 | 65.35 | Top10 | 65.03 | Top10 | 45.22 | Top10 | 27.58 | Top10 | 38.09 |
| Top20 | 67.16 | Top20 | 78.76 | Top20 | 85.14 | Top20 | 65.94 | Top20 | 79.69 | Top20 | 75.01 | Top20 | 49.24 | Top20 | 33.06 | Top20 | 43.86 |

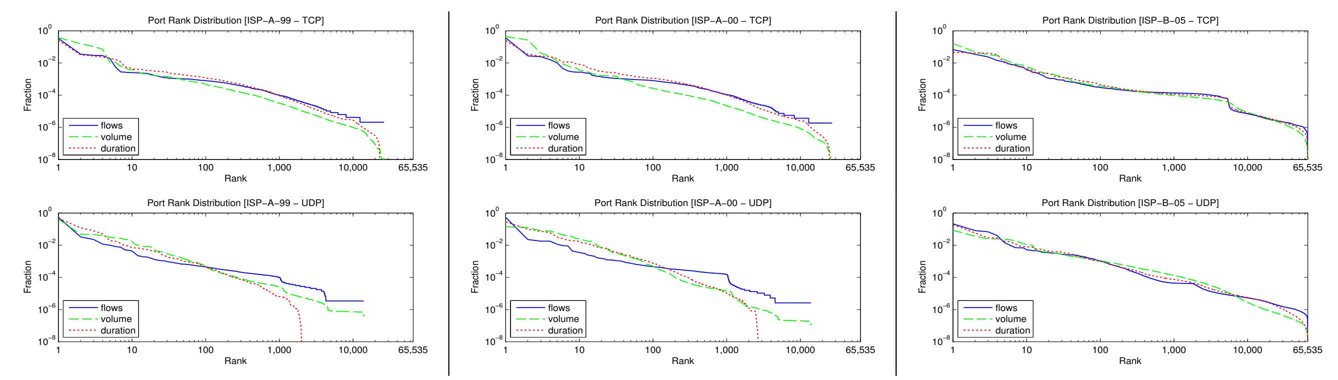



Fig. 11. Port Rank Distribution – Left:ISP-A-99, Center:ISP-A-00, Right:ISP-B-05

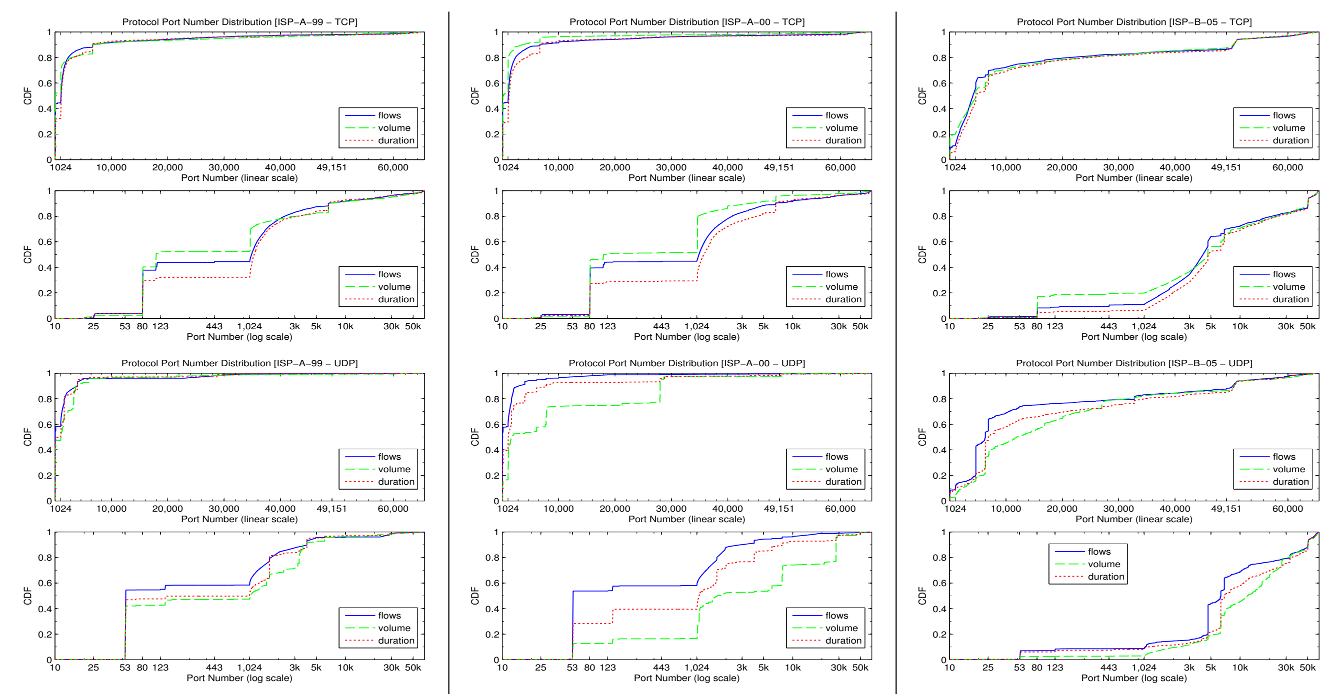



Fig. 12. Port Number Distribution – Left:ISP-A-99, Center:ISP-A-00, Right:ISP-B-05

TABLE VII
TOP10 PORT USAGE – LEFT:ISP-B-07, CENTER:LEIP-II-03, RIGHT:NZIX-II-00

| ISP-B-07-TCP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 80 | 11.78 | 80 | 32.40 | 80 | 4.66 |
| 6881 | 1.61 | 6881 | 1.20 | 6881 | 2.30 |
| 4662 | 1.42 | 119 | 1.02 | 4662 | 1.03 |
| 1863 | 1.06 | 4662 | 0.91 | 6346 | 0.95 |
| 443 | 0.82 | 443 | 0.71 | 443 | 0.63 |
| 110 | 0.62 | 3077 | 0.69 | 3077 | 0.48 |
| 6346 | 0.43 | 110 | 0.63 | 1863 | 0.45 |
| 25 | 0.39 | 6346 | 0.62 | 3724 | 0.33 |
| 20003 | 0.21 | 554 | 0.49 | 664 | 0.30 |
| 664 | 0.19 | 19101 | 0.38 | 32459 | 0.30 |
| Top10 | 18.52 | Top10 | 39.06 | Top10 | 11.43 |
| Top20 | 20.09 | Top20 | 41.06 | Top20 | 13.55 |

| LEIP-II-03-TCP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 4662 | 28.79 | 80 | 23.70 | 4662 | 18.37 |
| 80 | 9.79 | 4662 | 9.00 | 80 | 5.10 |
| 4661 | 0.81 | 6699 | 4.91 | 6346 | 4.26 |
| 443 | 0.46 | 1214 | 4.76 | 6435 | 2.32 |
| 1214 | 0.41 | 2634 | 0.94 | 1214 | 1.45 |
| 6346 | 0.39 | 1755 | 0.90 | 6699 | 0.91 |
| 21 | 0.31 | 554 | 0.88 | 1841 | 0.83 |
| 5190 | 0.30 | 20 | 0.58 | 6369 | 0.80 |
| 1841 | 0.26 | 22 | 0.56 | 6667 | 0.71 |
| 25 | 0.26 | 2959 | 0.45 | 5190 | 0.50 |
| Top10 | 41.77 | Top10 | 46.69 | Top10 | 35.25 |
| Top20 | 43.32 | Top20 | 50.20 | Top20 | 37.75 |

| NZIX-II-00-TCP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 80 | 24.21 | 80 | 44.96 | 80 | 17.51 |
| 443 | 2.09 | 20 | 2.96 | 25 | 2.64 |
| 25 | 1.57 | 443 | 2.19 | 6667 | 2.27 |
| 110 | 1.54 | 110 | 1.47 | 443 | 1.95 |
| 53 | 0.61 | 6699 | 1.30 | 119 | 0.82 |
| 3128 | 0.42 | 119 | 0.88 | 110 | 0.78 |
| 113 | 0.39 | 8080 | 0.87 | 2048 | 0.70 |
| 2048 | 0.26 | 53 | 0.87 | 6699 | 0.65 |
| 20 | 0.23 | 4044 | 0.81 | 179 | 0.52 |
| 37 | 0.23 | 2048 | 0.75 | 4044 | 0.48 |
| Top10 | 31.54 | Top10 | 57.07 | Top10 | 28.32 |
| Top20 | 32.63 | Top20 | 60.01 | Top20 | 30.86 |

| ISP-B-07-UDP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 53 | 3.15 | 3076 | 6.84 | 3076 | 18.02 |
| 6881 | 2.91 | 53 | 1.74 | 53 | 4.41 |
| 4672 | 2.69 | 3074 | 1.64 | 6346 | 3.97 |
| 3076 | 2.19 | 16567 | 1.12 | 6881 | 1.37 |
| 6346 | 0.83 | 6881 | 0.98 | 4672 | 1.14 |
| 49152 | 0.46 | 6348 | 0.97 | 8000 | 1.14 |
| 11773 | 0.35 | 6346 | 0.91 | 3072 | 0.88 |
| 18870 | 0.32 | 5004 | 0.87 | 41170 | 0.80 |
| 80 | 0.32 | 7000 | 0.75 | 10290 | 0.75 |
| 10986 | 0.31 | 13005 | 0.70 | 12288 | 0.74 |
| Top10 | 13.53 | Top10 | 16.53 | Top10 | 33.23 |
| Top20 | 16.12 | Top20 | 21.03 | Top20 | 38.31 |

| LEIP-II-03-UDP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 4672 | 13.63 | 27015 | 17.59 | 6257 | 9.64 |
| 6257 | 4.56 | 27005 | 8.59 | 1214 | 2.72 |
| 53 | 3.20 | 1701 | 3.71 | 1841 | 2.68 |
| 1214 | 2.38 | 6257 | 2.39 | 28800 | 2.40 |
| 1841 | 2.15 | 27010 | 2.21 | 53 | 2.20 |
| 2857 | 1.28 | 53 | 1.52 | 3600 | 1.86 |
| 3407 | 1.12 | 14758 | 1.18 | 2857 | 1.73 |
| 3847 | 1.10 | 7714 | 0.98 | 3772 | 1.51 |
| 4964 | 1.09 | 3281 | 0.91 | 3407 | 1.49 |
| 1027 | 1.08 | 7777 | 0.88 | 27015 | 1.38 |
| Top10 | 31.60 | Top10 | 39.96 | Top10 | 27.61 |
| Top20 | 39.90 | Top20 | 47.13 | Top20 | 36.42 |

| NZIX-II-00-UDP | | | | | |
|---|---|---|---|---|---|
| Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % |
| 53 | 32.41 | 27500 | 15.86 | 53 | 39.99 |
| 123 | 18.88 | 53 | 14.71 | 28800 | 7.22 |
| 1486 | 1.47 | 27005 | 9.46 | 1486 | 2.15 |
| 4978 | 1.04 | 27015 | 5.59 | 6112 | 2.11 |
| 1553 | 1.03 | 27910 | 4.71 | 123 | 2.03 |
| 4888 | 0.62 | 6112 | 4.18 | 443 | 1.83 |
| 137 | 0.57 | 123 | 1.85 | 137 | 1.25 |
| 1646 | 0.54 | 26005 | 1.44 | 1553 | 1.24 |
| 1024 | 0.54 | 28001 | 1.31 | 27005 | 1.20 |
| 1025 | 0.42 | 7777 | 1.27 | 520 | 1.14 |
| Top10 | 57.51 | Top10 | 60.39 | Top10 | 60.17 |
| Top20 | 59.09 | Top20 | 69.93 | Top20 | 66.97 |

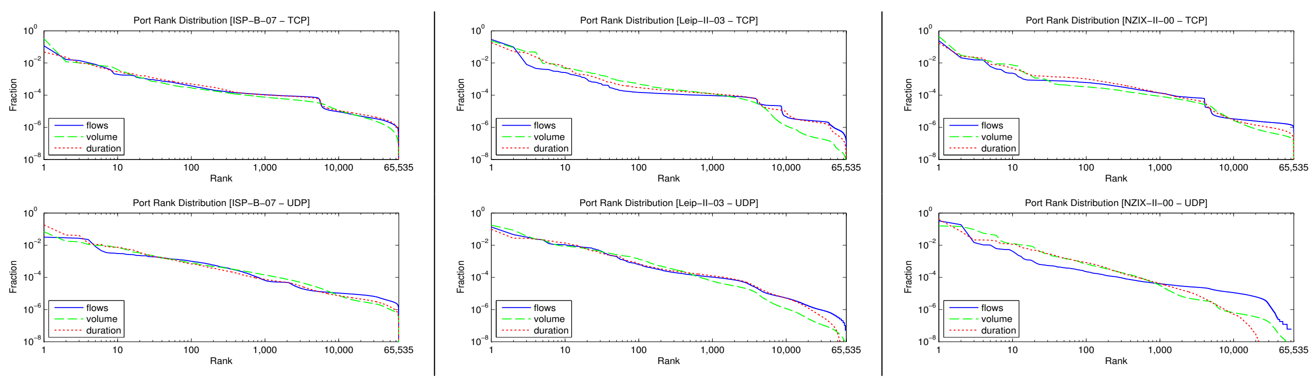



Fig. 13. Port Rank Distribution – Left:ISP-B-07, Center:LEIP-II-03, Right:NZIX-II-00

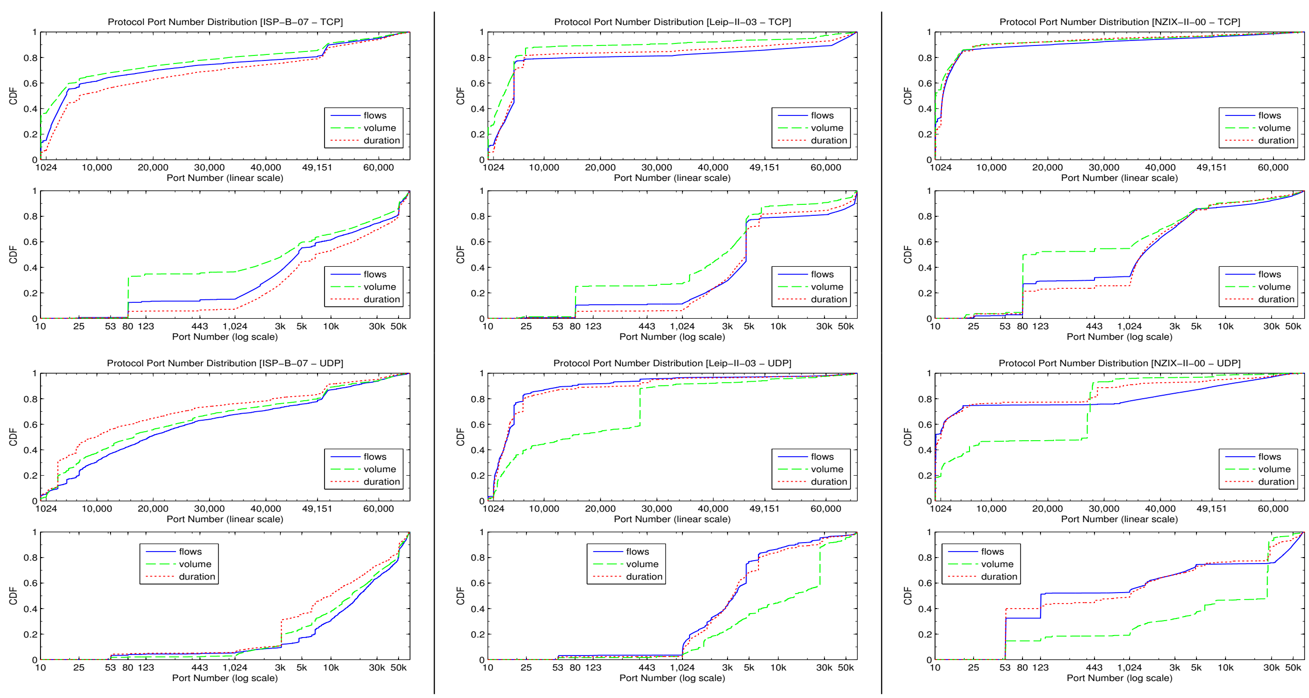



Fig. 14. Port Number Distribution – Left:SITE-I-03, Center:SITE-II-06, Right:SITE-III-04

TABLE VIII
TOP10 PORT USAGE – LEFT:SITE-I-03, CENTER:SITE-II-06, RIGHT:SITE-III-04

SITE-I-03-TCP

| Flows Port# | % | Volume Port# | % | Duration Port# | % |
|---|---|---|---|---|---|
| 80 | 22.72 | 20 | 43.73 | 80 | 14.61 |
| 6667 | 1.98 | 80 | 15.14 | 25 | 2.14 |
| 25 | 1.84 | 3306 | 1.03 | 20 | 1.98 |
| 135 | 0.58 | 119 | 0.72 | 21 | 1.02 |
| 20 | 0.43 | 1854 | 0.71 | 22 | 0.71 |
| 443 | 0.33 | 48611 | 0.71 | 6346 | 0.67 |
| 21 | 0.28 | 49200 | 0.63 | 119 | 0.62 |
| 113 | 0.14 | 50014 | 0.32 | 4662 | 0.53 |
| 2234 | 0.11 | 40458 | 0.30 | 3306 | 0.45 |
| 143 | 0.09 | 24961 | 0.29 | 6699 | 0.34 |
| Top10 | 28.51 | Top10 | 63.60 | Top10 | 23.08 |
| Top20 | 29.12 | Top20 | 66.25 | Top20 | 24.91 |

SITE-II-06-TCP

| Flows Port# | % | Volume Port# | % | Duration Port# | % |
|---|---|---|---|---|---|
| 80 | 34.37 | 20 | 24.89 | 80 | 23.96 |
| 6662 | 4.20 | 80 | 13.99 | 3306 | 5.46 |
| 3306 | 1.59 | 3306 | 9.55 | 20 | 2.43 |
| 443 | 1.02 | 443 | 0.98 | 25 | 1.95 |
| 21 | 0.90 | 2518 | 0.91 | 443 | 1.75 |
| 25 | 0.82 | 1642 | 0.91 | 22 | 1.50 |
| 20 | 0.48 | 1749 | 0.84 | 119 | 1.47 |
| 6944 | 0.43 | 1197 | 0.61 | 21 | 1.12 |
| 22 | 0.38 | 3371 | 0.33 | 6881 | 0.57 |
| 1863 | 0.36 | 4967 | 0.33 | 554 | 0.54 |
| Top10 | 44.55 | Top10 | 53.33 | Top10 | 40.75 |
| Top20 | 46.17 | Top20 | 55.56 | Top20 | 42.44 |

SITE-III-04-TCP

| Flows Port# | % | Volume Port# | % | Duration Port# | % |
|---|---|---|---|---|---|
| 80 | 20.97 | 80 | 38.71 | 80 | 10.69 |
| 3531 | 6.33 | 6881 | 3.50 | 3531 | 7.52 |
| 1863 | 3.34 | 6882 | 1.85 | 1863 | 3.00 |
| 220 | 3.26 | 20 | 1.53 | 6881 | 2.36 |
| 25 | 0.81 | 554 | 1.50 | 6346 | 1.36 |
| 443 | 0.72 | 22 | 1.38 | 6882 | 1.09 |
| 5190 | 0.44 | 1214 | 1.28 | 5190 | 0.96 |
| 4662 | 0.39 | 1755 | 1.02 | 5757 | 0.84 |
| 2703 | 0.36 | 6346 | 1.00 | 6667 | 0.79 |
| 6346 | 0.32 | 3155 | 0.84 | 4662 | 0.73 |
| Top10 | 36.93 | Top10 | 52.62 | Top10 | 29.34 |
| Top20 | 38.54 | Top20 | 56.50 | Top20 | 34.43 |

SITE-I-03-UDP

| Flows Port# | % | Volume Port# | % | Duration Port# | % |
|---|---|---|---|---|---|
| 53 | 44.71 | 53 | 41.57 | 53 | 40.61 |
| 123 | 11.09 | 36682 | 6.53 | 2568 | 9.00 |
| 33129 | 7.47 | 8164 | 5.94 | 4772 | 8.20 |
| 2568 | 3.55 | 33129 | 4.40 | 2131 | 7.52 |
| 4772 | 3.10 | 4772 | 3.07 | 33129 | 6.11 |
| 2131 | 2.14 | 36644 | 3.03 | 28784 | 5.48 |
| 29812 | 1.46 | 2568 | 2.74 | 36644 | 3.66 |
| 36644 | 0.96 | 2131 | 2.74 | 45566 | 2.13 |
| 1028 | 0.60 | 123 | 1.99 | 1029 | 1.66 |
| 1025 | 0.45 | 20020 | 1.84 | 3685 | 1.56 |
| Top10 | 75.53 | Top10 | 73.86 | Top10 | 85.95 |
| Top20 | 78.47 | Top20 | 86.07 | Top20 | 91.98 |

SITE-II-06-UDP

| Flows Port# | % | Volume Port# | % | Duration Port# | % |
|---|---|---|---|---|---|
| 53 | 37.56 | 5004 | 12.72 | 53 | 21.89 |
| 62375 | 5.81 | 53 | 5.99 | 63395 | 12.60 |
| 63395 | 4.99 | 49200 | 3.79 | 62375 | 7.31 |
| 0 | 1.59 | 1455 | 3.57 | 1027 | 2.57 |
| 4665 | 0.95 | 10000 | 3.53 | 34075 | 1.42 |
| 6881 | 0.77 | 54041 | 2.72 | 6970 | 1.41 |
| 34075 | 0.73 | 2746 | 2.52 | 1028 | 1.25 |
| 123 | 0.61 | 2328 | 2.30 | 5004 | 0.95 |
| 54811 | 0.57 | 31189 | 2.15 | 27014 | 0.88 |
| 54045 | 0.56 | 14634 | 1.82 | 54041 | 0.84 |
| Top10 | 54.13 | Top10 | 41.11 | Top10 | 51.13 |
| Top20 | 57.00 | Top20 | 52.95 | Top20 | 58.05 |

SITE-III-04-UDP

| Flows Port# | % | Volume Port# | % | Duration Port# | % |
|---|---|---|---|---|---|
| 53 | 12.67 | 53 | 4.44 | 53 | 7.89 |
| 1630 | 2.25 | 1028 | 3.38 | 6660 | 3.98 |
| 32769 | 2.08 | 17479 | 2.30 | 6346 | 3.24 |
| 32774 | 2.02 | 7000 | 0.94 | 3531 | 3.07 |
| 3531 | 1.94 | 6660 | 0.92 | 32774 | 2.92 |
| 3680 | 1.84 | 32774 | 0.90 | 4121 | 2.81 |
| 1721 | 1.69 | 32773 | 0.73 | 1630 | 2.11 |
| 1906 | 1.65 | 16384 | 0.65 | 32769 | 2.01 |
| 1272 | 1.59 | 13992 | 0.59 | 3680 | 1.89 |
| 37755 | 1.48 | 5004 | 0.57 | 1272 | 1.79 |
| Top10 | 29.20 | Top10 | 15.42 | Top10 | 31.71 |
| Top20 | 40.13 | Top20 | 19.93 | Top20 | 45.95 |

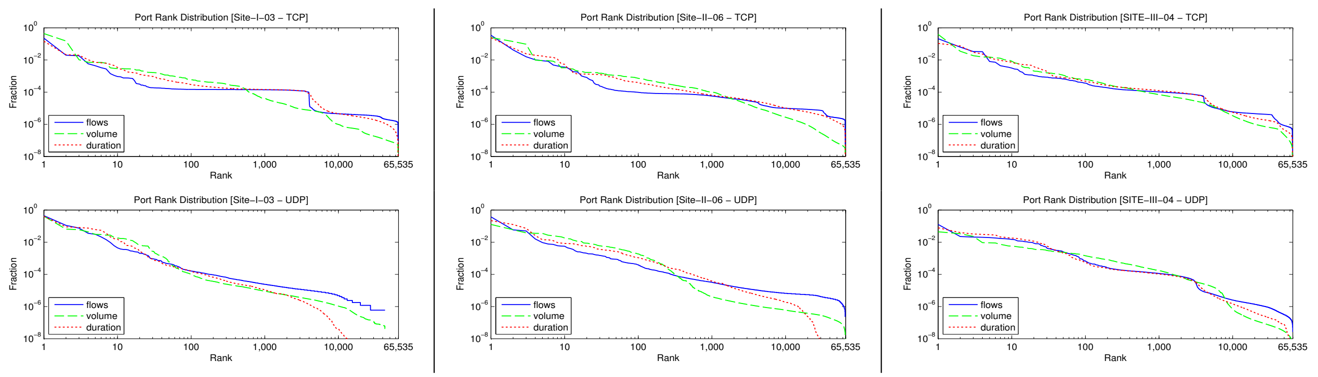



Fig. 15. Port Rank Distribution – Left:SITE-I-03, Center:SITE-II-06, Right:SITE-III-04

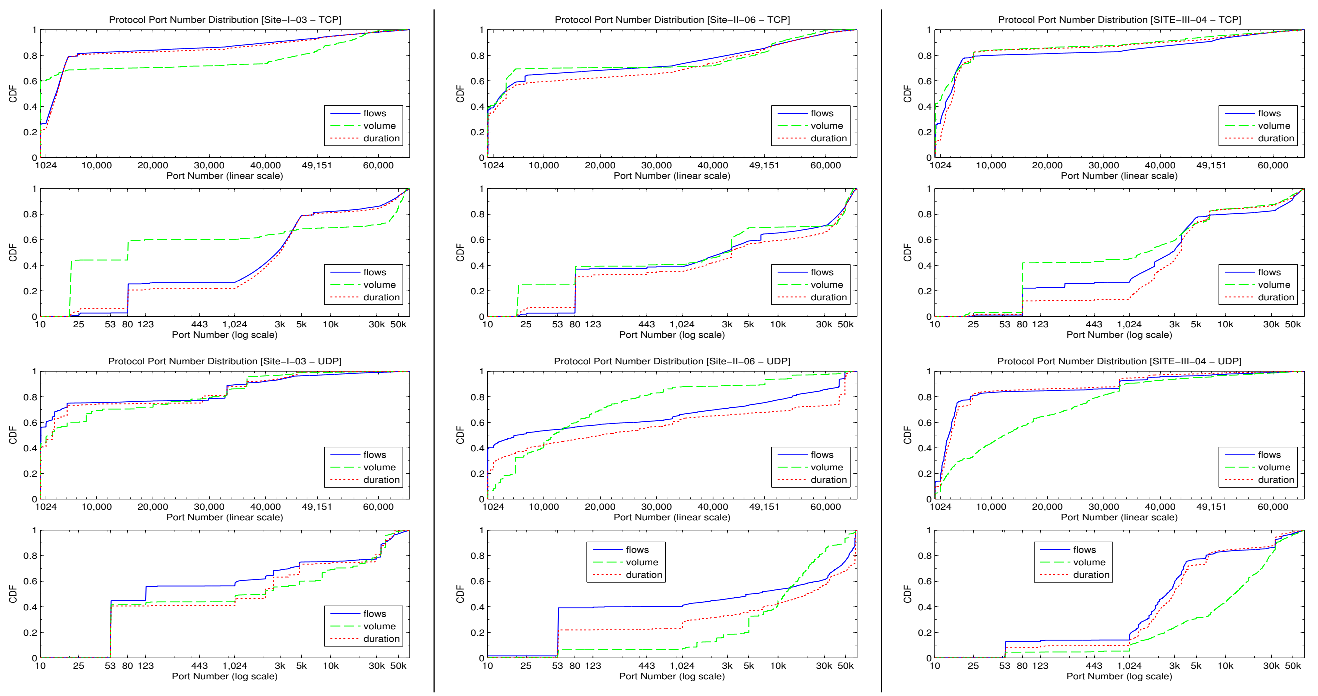



Fig. 16. Port Number Distribution – Left:SITE-I-03, Center:SITE-II-06, Right:SITE-III-04

TABLE IX
TOP10 PORT USAGE – LEFT:WITS-04, CENTER:WITS-05, RIGHT:WITS-06

| WITS-04-TCP | | | | | | WITS-05-TCP | | | | | | WITS-06-TCP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 80 | 26.75 | 80 | 56.38 | 80 | 19.44 | 80 | 25.84 | 80 | 61.12 | 80 | 23.84 | 80 | 28.56 | 80 | 61.05 | 80 | 22.80 |
| 443 | 4.98 | 443 | 9.63 | 443 | 8.00 | 443 | 10.12 | 443 | 6.21 | 25 | 9.53 | 25 | 7.42 | 443 | 9.01 | 25 | 11.69 |
| 25 | 2.25 | 10000 | 0.74 | 25 | 4.20 | 25 | 3.59 | 2048 | 1.87 | 443 | 4.08 | 443 | 5.30 | 2048 | 0.90 | 443 | 6.73 |
| 22002 | 0.96 | 44329 | 0.74 | 6667 | 1.35 | 2703 | 2.44 | 8080 | 1.08 | 1863 | 1.85 | 2703 | 2.69 | 25 | 0.90 | 1863 | 0.99 |
| 113 | 0.85 | 119 | 0.69 | 1863 | 1.20 | 2048 | 0.83 | 10000 | 0.92 | 2048 | 0.71 | 1863 | 0.57 | 8080 | 0.59 | 10000 | 0.54 |
| 220 | 0.78 | 2048 | 0.69 | 6881 | 0.80 | 1863 | 0.83 | 554 | 0.84 | 3389 | 0.67 | 2048 | 0.56 | 10000 | 0.50 | 8810 | 0.52 |
| 1863 | 0.71 | 6881 | 0.68 | 6882 | 0.54 | 113 | 0.62 | 25 | 0.71 | 2703 | 0.39 | 8810 | 0.17 | 22 | 0.37 | 2703 | 0.44 |
| 2048 | 0.36 | 2508 | 0.57 | 10000 | 0.47 | 3001 | 0.50 | 873 | 0.61 | 10000 | 0.35 | 26547 | 0.17 | 110 | 0.36 | 6667 | 0.38 |
| 1025 | 0.24 | 25 | 0.49 | 22 | 0.42 | 6000 | 0.23 | 3389 | 0.36 | 22 | 0.35 | 8080 | 0.15 | 1748 | 0.32 | 22 | 0.29 |
| 1438 | 0.18 | 6882 | 0.41 | 6883 | 0.41 | 8080 | 0.23 | 2034 | 0.30 | 8080 | 0.26 | 143 | 0.13 | 4556 | 0.24 | 5222 | 0.26 |
| Top10 | 37.89 | Top10 | 70.62 | Top10 | 36.42 | Top10 | 44.99 | Top10 | 73.71 | Top10 | 41.77 | Top10 | 45.60 | Top10 | 74.00 | Top10 | 44.37 |
| Top20 | 38.76 | Top20 | 73.51 | Top20 | 38.54 | Top20 | 46.13 | Top20 | 75.32 | Top20 | 43.10 | Top20 | 46.38 | Top20 | 75.77 | Top20 | 45.92 |

| WITS-04-UDP | | | | | | WITS-05-UDP | | | | | | WITS-06-UDP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Flows | | Volume | | Duration | | Flows | | Volume | | Duration | | Flows | | Volume | | Duration | |
| Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % | Port# | % |
| 53 | 27.23 | 53 | 33.63 | 53 | 27.73 | 53 | 36.21 | 53 | 45.84 | 53 | 13.82 | 53 | 35.43 | 53 | 43.24 | 53 | 15.52 |
| 123 | 6.22 | 16384 | 33.20 | 123 | 9.08 | 123 | 4.66 | 123 | 2.85 | 1194 | 9.78 | 17940 | 2.13 | 17940 | 3.07 | 123 | 9.47 |
| 1026 | 4.35 | 27960 | 2.38 | 10000 | 3.12 | 1038 | 0.48 | 12294 | 2.57 | 123 | 9.09 | 123 | 1.17 | 15607 | 2.50 | 17940 | 8.04 |
| 137 | 0.58 | 123 | 2.25 | 10003 | 3.08 | 32768 | 0.42 | 27960 | 1.36 | 1038 | 4.35 | 15282 | 1.16 | 123 | 0.83 | 15282 | 5.28 |
| 1025 | 0.25 | 1701 | 1.65 | 137 | 1.64 | 6277 | 0.22 | 24794 | 0.93 | 10023 | 2.68 | 6277 | 0.16 | 1406 | 0.78 | 6277 | 4.26 |
| 1027 | 0.23 | 1026 | 1.45 | 32774 | 1.20 | 1026 | 0.15 | 1194 | 0.79 | 10897 | 2.60 | 33625 | 0.12 | 10984 | 0.66 | 22361 | 3.16 |
| 32768 | 0.21 | 16386 | 1.20 | 32768 | 1.07 | 32769 | 0.14 | 6277 | 0.65 | 22391 | 2.59 | 13364 | 0.11 | 33522 | 0.63 | 14201 | 3.11 |
| 1028 | 0.20 | 137 | 0.62 | 49157 | 1.07 | 1025 | 0.14 | 32768 | 0.47 | 10008 | 2.13 | 4672 | 0.11 | 5002 | 0.58 | 33625 | 1.26 |
| 1029 | 0.14 | 1027 | 0.32 | 1030 | 1.06 | 1027 | 0.12 | 1038 | 0.46 | 32768 | 1.89 | 32768 | 0.10 | 15282 | 0.54 | 5011 | 1.06 |
| 1030 | 0.13 | 161 | 0.28 | 952 | 1.06 | 24441 | 0.11 | 161 | 0.37 | 6277 | 1.76 | 1036 | 0.07 | 54045 | 0.51 | 33089 | 1.05 |
| Top10 | 39.54 | Top10 | 76.97 | Top10 | 50.11 | Top10 | 42.65 | Top10 | 56.30 | Top10 | 50.70 | Top10 | 40.57 | Top10 | 53.35 | Top10 | 52.21 |
| Top20 | 40.46 | Top20 | 78.69 | Top20 | 60.19 | Top20 | 43.43 | Top20 | 58.75 | Top20 | 59.48 | Top20 | 41.00 | Top20 | 56.36 | Top20 | 60.73 |

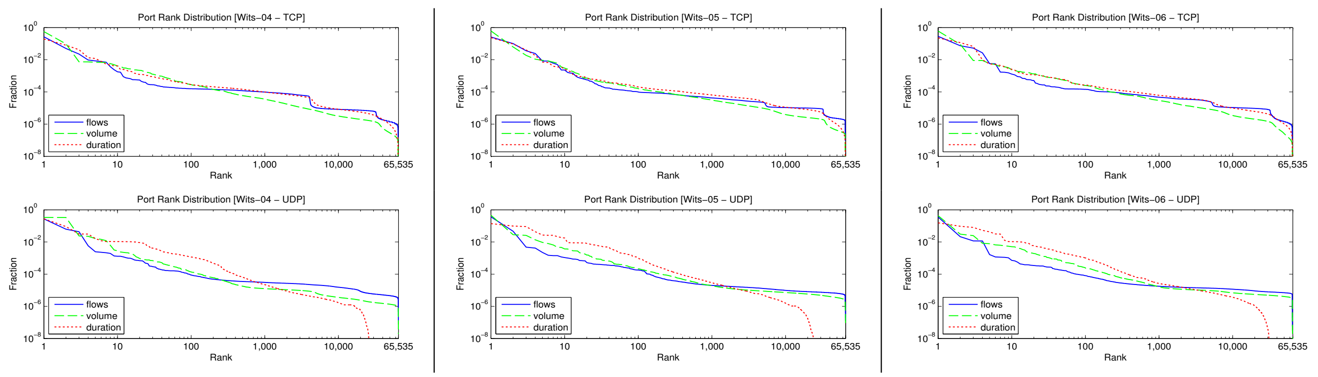



Fig. 17. Port Rank Distribution – Left:WITS-04, Center:WITS-05, Right:WITS-06

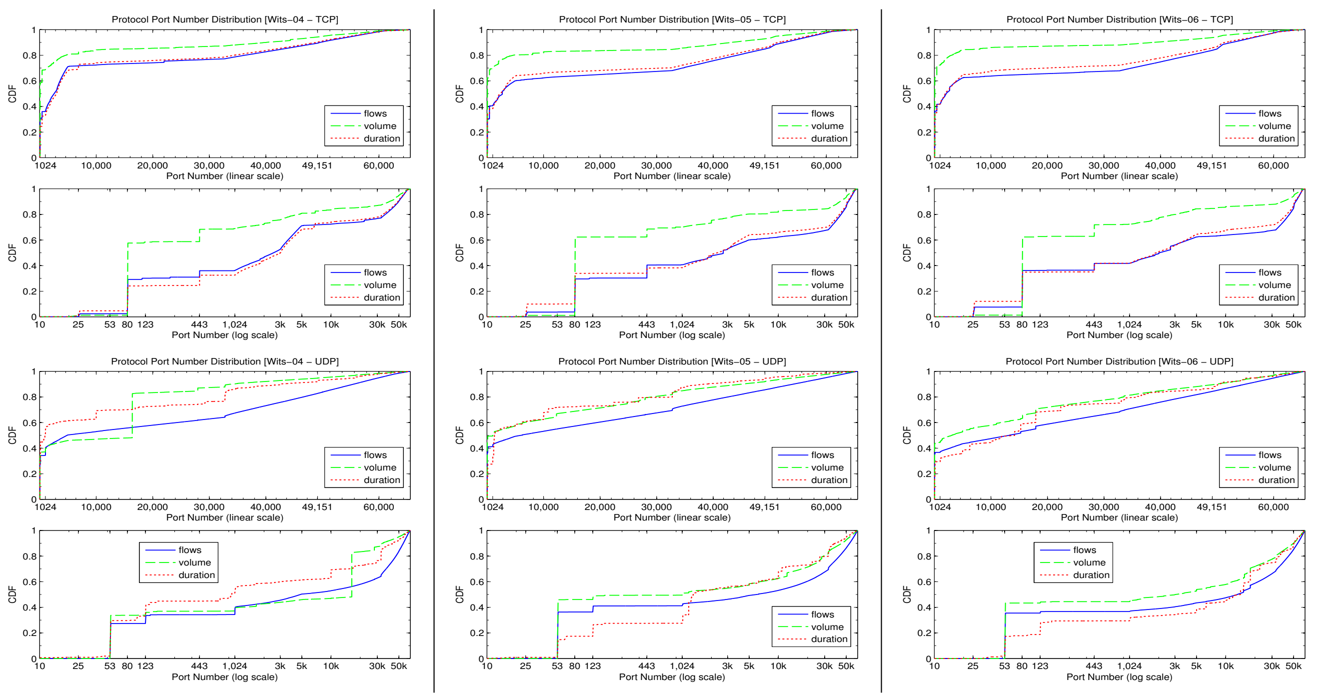



Fig. 18. Port Number Distribution – Left:WITS-04, Center:WITS-05, Right:WITS-06

# Prioritized Redundancy of Data Storage in Wireless Sensor Networks

Cosmin Dini

University of Haute Alsace
34 rue du Grillenbreit 68008 Colmar – France
France
cosmin.dini@uha.fr

Pascal Lorenz

University of Haute Alsace
34 rue du Grillenbreit 68008 Colmar – France
France
lorenz@ieee.org

***Abstract*— Wireless Sensor Networks have evolved into complex deployments, where their nodes can have a full network protocol stack, database systems, etc. The main rationed resource in such a deployment is energy. Having power usage tightly managed ensures a long operational life for the node in cases where replenishments (either via recharge or battery change) are difficult or impossible. The basic deployment of Wireless Sensor Networks consists of sensing nodes as well as a relay node (i.e., sink), which collects sensory data to be relayed via a reliable network. The sink node can become unreachable due to malfunction, scheduled uptime or, in the case of mobile sink nodes, due to being out of the sensor nodes' reach. In addition, the sensor nodes may decide against relaying data for some period. In these cases, optimal use of sensor node memory space also becomes critical. In this article, we classify data types and establish a set of node level approaches that can be taken to make the most of limited data storage via a prioritized data reduction. Wireless Sensor Networks often operate in locations with limited access while relying on a restricted set of resources. This calls for careful management of local assets such as energy and storage. Storage is used to host data of various interest levels for subsequent relay to a base station or for queries through the network. The size of the data needs to be managed in view of data's relevance. While there is a debate for complete or partial data extraction from sensors, having special data process functions and operation primitives proven useful for sensor OSs. To deal with robustness and reliability, data processing at the network/sensor level satisfies some of the reliability requirements, especially when communications are not operational. There are situations were data reduction is an alternative when storage is not longer available and data is aging, especially when some sensor links are not properly operational. Using predictions and optimized parameters to prioritize data reduction is a solution. While approaching the data reduction from the perspective of a single node is important, there are benefits from looking at Wireless Sensor Networks deployment as a whole. There is an opportunity to relocate some data to less active nodes and spare it from a reduction process. There are energy considerations to take into account in evaluating the potential benefit from spending some energy to relocate data. Three factors play a part in this: the source node, the receiver node, and the data itself. The source node needs to save enough energy to relay its data once the sink node becomes available. The receiver node needs to have space available, or at least have a lot of space occupied by low importance data. The data itself needs to be self-enclosed as far as parameters needed for importance computation. Another aspect that becomes feasible in a multi node environment is having redundant copies of data to protect against potential node failures. These copies have their own particularities, a notable one being that their importance function cannot depend on other data. This paper presents a methodology that enables the node to be useful by collecting data beyond the point where its data storage size would otherwise allow. We build upon primitive data reduction operations to construct a framework that can be used for a sensible data age-out scheme. The methodology enables various factors, both internal and external to the sensor, to influence the data aging process and the data reduction operations. Additionally, we define special heuristics for data reduction using a set of data processing primitives and special data parameters. We apply these heuristics via a methodology that enables various factors, both internal and external to the sensor, to influence the data aging process and the data reduction operations.**

***Keywords - sensors; networks; storage; operations; data management; data sensor storage; data priority; optimized parameters; prediction models; prioritized redundancy.***

## I. INTRODUCTION

Wireless Sensor Networks (WSN) have appeared as a special case of wireless networks specialized in data gathering. They find a niche in industrial monitoring, military sensing, locating assets, etc. [1]. An advantage comes from the fact that such a deployment makes no assumption about pre-existing infrastructure. Therefore, the WSN nodes have to bring with them all the resources that they expect to utilize during their lifetime: power, storage, processing unit, antenna, etc. Figure 1 presents the classical WSN deployment of sensor nodes $n_i$, and a sink S. Each sensor node n can wirelessly communicate with other sensor nodes that are within reach, shown in dashed lines.

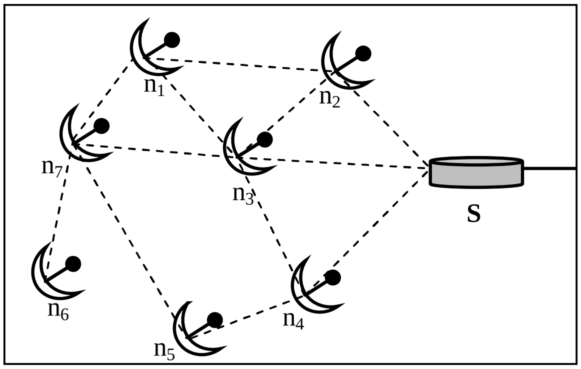



Figure 1. Classical WSN deployment.

Much research has been devoted to optimizing the usage of limited resources. In particular for WSNs, energy has been a main focus. Very resourceful power sources have been suggested, nuclear energy included. Both routing and dissemination protocols have been approached from the energy conservation standpoint. In addition, some proposals target replenishing energy from sources such as sun, wind, water flow, etc. [10].

Other resources related to WSN deployment have received less attention compared to energy source and usage. In this article, the focus is data storage, which just like energy, is limited. Unlike energy, which once used is gone, data storage space can be reclaimed by discarding existing stored data. Another difference between energy and data storage constraints is that one can propose ways of harnessing energy to prolong the life of a wireless node, but so far, there is no way of harnessing data storage space.

There are many examples outside of the WSN world where storage space is a factor. Many cases are outside of the technology world and include storage spaces, garages, warehouses, disk drives, kitchen drawers, video surveillance recording, etc. Unlike the WSN scenario, these cases allow for direct human intervention: additional storage space, although for a cost, can be achieved.

We consider a WSN node in cases where unloading the data is not possible at all times. The sink node, or the next hop routing node, may not be available at times. Referring to Figure 1, if the sink S is not available, nodes $n_2$, $n_3$, and $n_4$ cannot relay their data. Similarly, if $n_7$ is unavailable, $n_6$ cannot relay data as $n_7$ is the only possible next hop for routing. It is the responsibility of the node to use the storage space in order to hold the most relevant data until this data can be relayed at the expense of less relevant data. A set of business logic instructions that are deployed with the node provide the decision making. The same principles can apply to sensor nodes that are deployed and later physically retrieved for data extraction without ever having to wirelessly transmit any data.

Specialized operating systems, such as TinyOS, provide the primitive operation set for code execution. The data storage approach that we propose also needs to be supported by a set of primitive operations.

Unattended data collections in remote areas without ease of access pose a challenge for traditional network deployments. Usual assumptions no longer apply, specifically: available space is an issue, power is no longer reliably available, and tech support can not immediately intervene to fix unexpected issues [13].

Wireless Sensor Networks (WSNs) have evolved to answer to these constraints. In addition to having sensing capabilities, WSNs also perform the essential functions of traditional networks, such as routing. However, under the constraints imposed by isolation of the deployment, approaches to traditional algorithms need to be reconsidered [14].

There are two approaches to data handling in WSNs. On the first hand, a sensor network can be left to its bare functions of data collection. All data is then forwarded to a base station for data analysis. In such a case, there is little computation requirement from the processing unit on the sensor nodes. This can lead to large amounts of data circulating through the network, leading to excessive use of power, a critical resource in WSNs. A second approach is to leave the collected data on the sensor nodes, while retrieving specific needed data by running queries through the network. This limits the data transfer to the exact results of the data extraction. In this case, we deal with limitations caused by the storage space on the nodes. There are some approaches to alleviate somewhat the effects of a limited storage space, and they usually involve some sort of load sharing [21].

Whatever the mechanism to make better use of storage space, there is nothing that can produce more available storage space. At some point, data needs to be aged out. What data and how it is aged out has not been fully addressed in existing literature. In [12], a set of primitive operations to support a prioritized data reduction is presented. In its description, the approach was limited to presenting the simplest building blocks on a single node. The data reduction basically presented in [12] is extended to show how the primitives are used together in order to manage the storage size aspect of a WSN node operational state. Requirements posed by unattended data collections in remote areas become very challenging for traditional network deployments. The main problem is raised by the fact that users might look for full collected data, while effective business models take into consideration a small fraction of it.

Most of the WSNs (Wireless Sensor Networks) also perform the essential functions for data processing; one of the most important, in special cases of uncontrolled link availability, is data reduction under several the constraints driven by the nature of the data, the relevance of the data, the data dependency, and the business model using such data. A sensor on a node captures a time series representing the evolution of a sensed physical variable over space and time.

Reducing the amount of data sent throughout the network is a key target for long-term, unattended network sensors. A second target, equally relevant, is defined by unattended networks with unreliable links. In this case, gathered data might rapidly aging and exceed the storage capability at a given node. Data reduction mechanisms are used to partially handle these cases [22] [23] [36].

Unnecessary communication as well as appropriate data reduction techniques can be modeled in the case of physical phenomena with a pre-defined, application-dependent accuracy [36]. If an accepted error is bounded as [$-e$, $+e$], with $e$ in $R+$, only values exceeding the predicted one by +/- $e$ will be communicated. Similarly, if the errors of the gathered values are within the bonded interval, data reduction can be more simplified.

The difference between these two situations is the following; for communication, in the absence of notification from a given node, the receiving node assumes that the value obtained form the prediction model is in the required error bound. For unreliable links, despite if this assumption, the gathering node must reduce the amount of collected data that accumulates. In this paper, we present a series of heuristics on predictions used to summarize collected data.

The rest of the article first deals with primitive operations that are executed at node level independently of the rest of the WSN. We then present the state of the art in collaborative data storage as well as data reduction.. The next section reviews the primitives introduced in [12]. We then extend the single node model to present a complete framework. More heuristics for prediction on data processing are presented for multi-node complemented by prioritized data redundancy.

## II. Related Work

Current work in WSN associated storage management revolves around energy efficiency in manipulating and querying the data [3][4], improving the characteristics of stored data [5] [6][7], and making use of adjacent nodes in order to gain access to additional storage [8][9].

Siegmund et al. [5] propose FAME-DBMS to provide a robust data storage solution. This system ensures reliability and integrity of the data, and provides a customizable query engine. It answers to the requirements related to data retrieval more so than to storing data. In order to deal specifically with encryption, Joao Girão et al. [6] present edTinyPEDS, an encryption data storage engine.

The energy usage is still relevant when focusing on storage and querying. Ahn and Krishnamachari [3] evaluated the scalability of a WSN performance with respect to the distributed nature of data.

Park and Elmasri [4] evaluated several storage schemes in terms of where the data is stored, what types of routing protocols are most appropriate, and finally the impact that each storage approach has on energy usage.

Khan et al. [8] presented the problem of data persistence in a congested WSN scenario. The main point is that congested networks can drop packets, which in turn translates to a waste of energy equivalent to the cost of sending the dropped packets. The proposed approach involves clustering where cluster nodes can act as temporary buffers during congestion periods.

Current work on WSN related storage has been limited to data characteristics and management across several nodes. The case of a standalone node has not yet been considered.

In the following, we present current approaches to deal with a continuously accumulating amount of data. We cover both aspects geared towards individual nodes and aspects geared towards network wide approaches.

### *2.1 Data Reduction*

In their work on managing data in storage-centric WSNs, Diao et al. [20] touch on some of the issues surrounding the management of growing data amounts. There exists a common perception that, for historical data, every single bit of collected data must be stored and maintained. A distinction is made between “dumb data collection” sensor networks, and those that support queries throughout an entire WSN. There is clearly a serious potential for excessive data accumulation in the case of the latter. The potential for data aging to alleviate the problem is brought up, but no framework is established on how aging can be done in a sensible manner; vacuuming older data to secondary storage is proposed as a potential solution, but it assumes a rather reliable connection to a base station.

Khan et al. [15] address the point of data persistence within a network in the case of congestion along transmission paths. Similarities can be seen between this case and the case of excessively accumulating data. The proposed solution in the case of a data storage issue due to congestion is additional buffering along the path to a sink node. This cannot be a full answer to the problem of a growing amount of collected data. Whatever the about of storage on hand, there will always be applications and processes demanding even more data space.

Tilak et al. [16] only briefly touch upon the data aging aspect of storage, but concentrate more on the benefits of collaborative or global approaches.

The approaches that we have seen so far lack consideration for the exact data that is collected. The tendency to treat data uniformly, or as a function of time only, can leave out data of significant importance.

A more functional solution to a single WSN node storage size management is given in [12]. The underlying assumptions are that we are dealing with a "dumb data collection" sensor network. The deployment is therefore only used to collect data and do the best it can, at each individual node, to maintain the most relevant data until an opportunity presents itself to relay all this data to a sink node. Data reduction is performed according to pre-established business case rules and several methods to achieve data reduction are presented. On a case by case basis, some data reduction approaches are preferable to others. We consider this to address the problem, but added benefit can be found in allowing nodes that experience a spike in data collection to share some of the data with neighboring node(s).

### 2.2 Collaborative Storage

Several approaches have been proposed with regards to collaborative storage. By having a wider view of the deployment, such protocols can attempt to address aspects such as alleviating situations where certain nodes produce more data than others, deduplication of data that was collected unknowingly by more than one sensor, and ultimately, as a side effect of deduplication, less energy is needed to relay the data to a base station.

In [17], Tilak et al. proposed a Cluster Based Collaborative Storage (CBCS) as a specific solution to collaborative storage. Several algorithm improvements in WSNs have this common approach of clustering nodes, electing cluster heads, and establishing an overall tree structure through the network. CBCS does this in the context of storage management. Nodes are grouped in clusters based on geographical data. Cluster heads (CH) are elected, one per cluster; they have the task of aggregating all data from the cluster nodes. This improves power use efficiency as only the CH needs to further relay the data towards the sink node. Such an approach seems to place more emphasis on the energy saving rather than dealing with large amounts of data. Aggregating all the data from the cluster nodes on the CH storage space cannot be beneficial when we are trying to solve storage issues.

In Figure 2, a visual representation of clusters is shown. Most clusters are created with geographical proximity in mind in order to minimize energy expenditures on communication. The designated or elected cluster heads, shown circled, handle the communication with a sink node, S, which is linked to a stable reliable network.

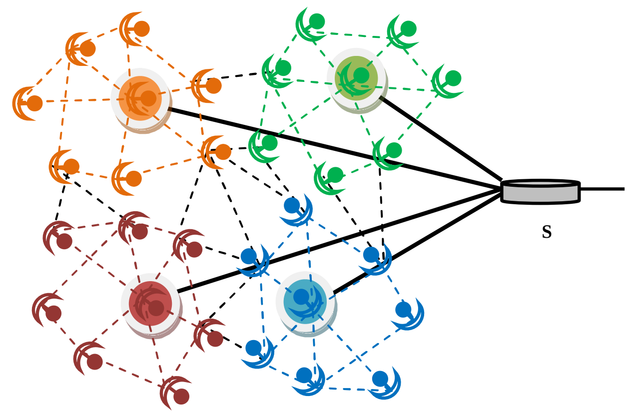



Figure 2. Clusters in a WSN.

In [18], Shenker et al. proposed a method, Data Centric Storage (DCS), to store data as identified by its name. Related data would in the end be stored either on the same or on neighboring sensor nodes. This would facilitate deduplication and also improve queries since data pertaining to a query would reside in the same proximity, therefore avoiding the need for queries to be run through large sections of the network. This approach is useful for WSNs that are designed to support searches, but does not apply to networks where data analysis is done offsite.

Siegmund et al. [19 addressed the issue of data integrity. Data redundancy is achieved in the network via the implementation of a new abstraction layer in the WSN. This layer can support the need for data redundancy. While robustness is of importance, there is very limited work in view of alleviating potential data overload in the network.

Collaborative storage presents challenges in terms of locating data during queries through the network. It also adds to the power requirements of the nodes. In the proposals published so far, there is no consideration for collaborative storage to mitigate memory limitations on some sensor nodes. There is also no effort to give a semantic to the stored data in order to assist with data age-out in the context of a node who is only storing the data without having produced it.

In the following, we summarize a data processing model introduced in [22] [23] and prediction approaches for data reduction [24].

In the past, the database community pushed different data-reduction operators, *e.g.*, aggregation and reduction, with no enough flexibility to handle extracting complete raw sensor readings (i.e., using "SELECT *" queries).

There are two specific needs to perform in-network data processing, *i.e.*, (i) to significantly reduce communications costs (energy), and (ii) to deal with link-down situations. In-network aggregation was proposed in [26][27], while data reduction via wavelets or distributed regression in [28][29].

All these techniques do not provide the desired data granularity, as requested by network users.

Managing data in a storage-centric approach was studied in different approaches, based on a reliable connection [32], additional buffering [30], or collaborative framework [31]. Details on collaborative storage are provided in [23].

OS primitives acting on recurring and non-recurring data collection have been proposed in [22]. Mainly, compression, thinning, sparsing, grain coursing, and range representation were used to deal with data aging in a pessimistic and optimistic approach. As a note, data deduplication was not considered in the above model. To optimize data reduction, concepts of data units, data importance, and compensation factors were introduced in [23]. Mainly, measurements are partitioned into contiguous intervals; data importance relates to the business semantics, while the compensation factor underlines the importance in business computation of a given data. The model considers that there is data that cannot be reduced in any circumstance. A mechanism for data dependency between different data units was presented in [23]. Further division of the data units (leading to more flexibility) was not considered at this point. Associated with the new concepts, the following functions were introduced: interval production function (only for recurring data), default compensation function, data importance function, and data reduction function. Solutions based on data redundancy (leading to more robust deployments and measurements) were not considered.

A data reduction specification use case considering data dependency across data units was presented in [23]. Consequently, appropriate values for data importance can be derived considering the constraints on the data importance computation as pertaining to two categories: (i) internal constraints and (ii) external constraints. External constraints are caused by factors over which input data has no effect. Such factors are data age and inherent interest in the data depending on the exact purpose of the data collection. Internal constraints represent inter- and intra-data dependencies.

A prediction model for approximate data collection is presented in [24]. The techniques are based on probabilistic models (BBQ system, [33]). We apply the prediction models to the framework proposed in [22][23] considering also the approximation scheme providing data compression and prediction [34] and predictive models from [35].

In this paper, we consider the PDR components introduced in [23] (Figure 1) to derive appropriate data reduction considering known correlations (spatial, temporal, etc.) and prediction models.

Several approaches and protocols have been proposed with regards to collaborative storage. By having a wider view of the deployment, such protocols attempt to address aspects such as alleviating situations, where certain nodes produce more data than others, deduplication of data that was collected unknowingly by more than one sensor, and ultimately, as a side effect of deduplication, less energy is needed to relay the data to a base station.

In [37], Tilak *et al.* proposed a Cluster Based Collaborative Storage (CBCS) as a specific solution to collaborative storage. Several algorithm improvements in WSNs have this common approach of clustering nodes, electing cluster heads, and establishing an overall tree structure through the network. CBCS does this in the context of storage management. Nodes are grouped in clusters, based on geographical data. Cluster heads (CH) are elected, one per cluster; they have the task of aggregating all data from the cluster nodes. This improves power use efficiency as only the CH needs to further relay the data towards the sink node. Such an approach seems to place more emphasis on the energy saving rather than dealing with large amounts of data. Aggregating all the data from the cluster nodes on the CH storage space cannot be beneficial when we are trying to solve storage issues.

In Figure 2, a visual representation of clusters is shown. Most clusters are created with geographical proximity in mind in order to minimize energy expenditures on communication. The designated or elected cluster heads, shown circled, handle the communication with a sink node, S, which is linked to a stable reliable network.

In [38], Shenker *et al.* proposed a method, Data Centric Storage (DCS), to store data as identified by its name. Related data would in the end be stored either on the same node, or on neighboring sensor nodes. This would facilitate deduplication and also improve queries since data pertaining to a query would reside in the same proximity, therefore avoiding the need for queries to be run through large sections of the network. This approach is useful for WSNs that are designed to support searches, but does not apply to networks where data analysis is done offsite.

Siegmund *et al.* [39] address the issue of data integrity. Data redundancy is achieved in the network via the implementation of a new abstraction layer in the WSN. This layer can support the need for data redundancy. While robustness is of importance, there is very limited work in view of alleviating potential data overload in the network.

Collaborative storage presents challenges in terms of locating data during queries through the network. It also adds to the power requirements of the nodes. In the proposals published so far, there is no consideration for collaborative storage to mitigate memory limitations on some sensor nodes. There is also no effort to give a semantic to the stored data in order to assist with data age-out in the context of a node who is only storing the data without having produced it.

Having multiple nodes interact in an effort to better manage storage space gives the opportunity for two considerations: storage space sharing and redundancy. In this chapter, we address these concepts from the perspective of prioritized data reduction.

*2.3 Storage space sharing*

*2.3.1 Load sharing*

Most research related to sharing across several nodes the impact created by data refer to balancing out the data access hot-spots. Occasionally, data queried in a network will be accessed in a skewed manner resulting in a small set of nodes receiving a disproportionate amount of queries and hence taxing the energy storage of those specific nodes. In order to deal with these cases, methods to replicate the data are presented as well as methods to divide the data across several nodes. In [40], the problem is approached from the perspective of k-d tree mapped data. In such a mapping, data storage is done independently of the node that produced the data. When one node becomes overwhelmed with respect to the amount of data that it needs to store, a tree rearrangement is done in order to balance the data load.

A k-d tree covers a two dimensional surface by recursively dividing the surface in two parts. At the point where we have a single node inside a parcel, that parcel no longer undergoes division and the node becomes the leaf of the path that created the parcel.

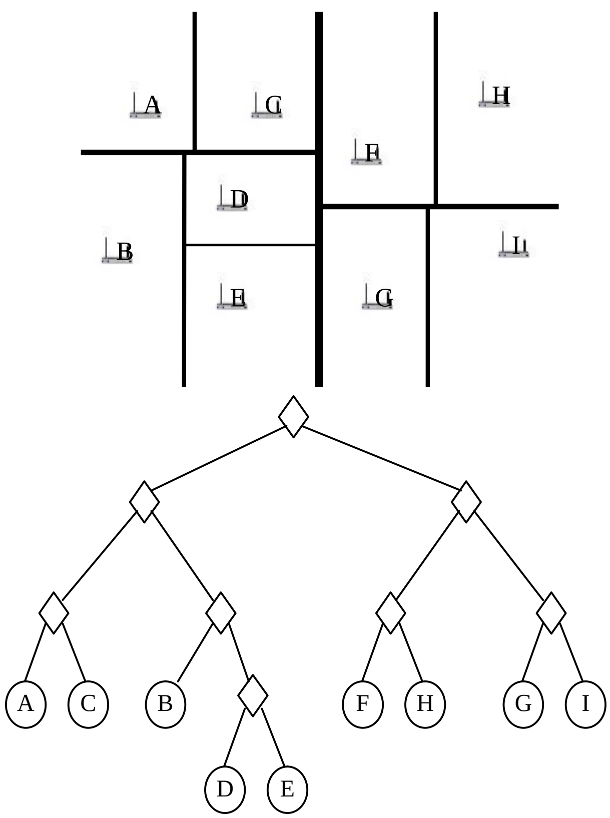



Figure 3. Division of a WSN in a k-d tree.

Another facet of load sharing is found in [41], where the authors address the problem of congestion at nodes on highly used paths. Not only are these nodes' energy supplies overtaxed, but the limited storage space that they have may not be enough to transiently hold all the packets. The proposal is to group the sensor nodes into small clusters, ideally a highly redundant deployment. When high traffic volume is sensed, the cluster head node starts to divert traffic to neighboring nodes. Because the nodes are in the vicinity, the congestion is localized to the specific area without spreading to other nodes. For critical data, the head node can make the decision to replicate the data several times so as to assure its continuity even in the event of a node failure. One drawback is a potential added delay caused by the added store and forward operations within a cluster.

*2.3.2 Storage sharing*

In this section, we cover the aspects of sharing the storage space that exists within an entire wireless sensor network. This is already done as part of geolocated data having as aim to facilitate queries. In our case, we assume a network that does not support in-network queries, but relays all data to the sink for offsite processing. Data produced by a node stays on that node until delivered to the sink. However, in the interest of balancing out space needs, we consider the possibility to forward certain data to other nodes as opposed to applying prioritized data reductions.

Until storage sharing, it was rather straight forward to perform a round of data reduction, i.e., select the data unit with the lowest importance and apply the reduction function. This could be done iteratively to bring the memory usage below a certain expected threshold. With an option to package and send data as a method of storage size recovery, several decision factors appear. How much data to send? How to select the most appropriate data to send? How to select the best node to send the data to? There are no definite answers in situations where external inputs dictate how much or how little data needs to be stored. There are however preferred ways to handle storage space sharing.

## III. FRAMEWORK AND MECHANISMS FOR ONE NODE

*3.1 Components of a WSN node*

In this section, we describe the conceptualized components performing the tasks - known as well as newly proposed - associated with a WSN node.

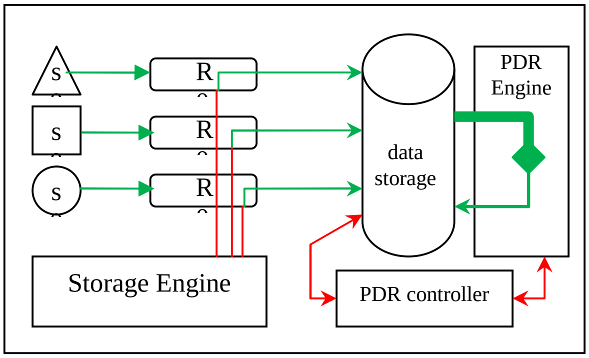



Figure 4. WSN node building blocks.

There are several physical sensors s on a WSN node, each specialized in sensing a specific parameter: temperature, pressure, etc. Each of these physical sensors has registers (R), which are updated to reflect a currently observed parameter vale. The Storage Engine (SE) has the appropriate business logic knowledge to determine at which point to trigger the transfer of data from the registers into the data storage. Prioritized Data Reduction (PDR) is achieved via a controller and an engine. The PDR controller performs periodic checks on the data storage and it can also accept traps from the data storage. When business knowledge dictates, the PDR controller instructs the PDR engine to perform specific data reduction operations.

We first focus on the operations executed inside the PDR engine. For that matter, we first characterize how the SE operates.

*3.2 A model for data classification*

Triggers or conditions are used to control data collection. These conditions trigger the SE to perform transfer operations from R into the data storage.

*3.2.1 Conditions*

We break down the conditions into two categories: value-based and sequence-based. Value-based conditions are evaluated as a whole and can immediately evaluate to TRUE or FALSE:

---

if ((temperature > α) && (humidity > β))

---

Sequence-based conditions are evaluated in sequence. We must establish a chain of TRUE evaluations in order for the entire condition to be considered TRUE. If one element in the sequence evaluates to FALSE, we pause there until the next round of evaluations:

---

```
sequence {
  s1: (temperature < 0C)
  s2: (temperature > 0C)
  s3: (temperature > 10C)
  s4: (temperature > 20C)
  s5: (temperature > 25C)
}
```

---

Sequences are useful in establishing trends. Even though one sequence item may temporarily be FALSE, once we arrive to s5 and it evaluates to TRUE, the sequence is said to evaluate to TRUE.

*3.2.2 Data Collection*

Data collections can be classified in two categories, based on the periodicity of the collection: there are samplings at defined intervals (recurring), or single shot samples (non-recurring).

*3.2.2.1 Recurring Data Collection*

Recurring data collections involve taking specific measurements at defined time intervals. One example of this type of data collection is temperature. We can specify such intervals at microseconds to hours and even less frequent. The sampling rate would depend on the exact use for the data collected and according to the business model. Not all recurring data sampling is enabled by default and continuously done throughout the live of the sensor. There are conditions that can trigger starting or stopping a series of such data collections.

As an example, we assume we are monitoring temperature on the side of a volcano in order to detect abnormally high values. We assume that baseline values are available. Under normal conditions, a few degrees difference warmer than prevailing temperatures may be acceptable, but once the temperature crosses a certain value (hinting of some sort of activity), it becomes interesting to start taking measurements of several factors: sound, land vibrations, gas composition, etc. When the ambient temperature returns to a specific value, it may not be of interest to sample a wide variety of parameters.

Recurring data sampling can be defined by a start condition, a stop condition, a sampled parameter, and a recurrence window.

---

start: (temperature > (baseline + Δ))
stop: (temperature <= baseline)
sample parameter: sound
recurrence: 1ms

---

Once a collection has started, a second instance of the same collection cannot start even though the start condition evaluates as TRUE.

It serves no purpose to sample data any faster than the sensory devices can update registers holding the sensed data.

As a special case, the primitive values TRUE and FALSE can be used as a start condition and a stop condition respectively, and hence a continuous sampling is achieved.

We label as a *recurring data instance (RDI)* a recording of the entire set of data points from the collection start to collection stop, or to current time if the collection has not stopped.

Defining an RDI

---

RDI: ($T_{start}$, $T_{end}$, param, recurrence, resolution, compression), where:
$T_{start}$: start time
$T_{end}$: stop time
param: the parameter being collected
recurrence: how often the collection is done
resolution: how precise the stored value is
compression: boolean stating if the RDI has been compressed
e.g., RDI(2009/12/06 16:43:23, ongoing, temperature, 60 seconds, 0.01C, FALSE)

---

*3.2.2.2 Non-Recurring Data Collection*

Non-recurring data collection happens when a specific condition is met. It leads to a single value being stored every time the condition evaluated to true. Such conditions must be written as a sequence so as to avoid constant firing of the rule and hence leading to a constant parameter sampling.

---

```
sample condition:
   sequence {
      light < 50lx
      light > 10000lx
   }
sample parameter: temperature
```

---

What the example above means is that we are sampling the temperature of a location after the sun has come up and is providing a specific light intensity. The reason we require a value increase from under 50lx to over 10000lx is to establish a trend.

If we simply state the above as :

---

```
sample condition: (light > 10000lx}
sample parameter: temperature
```

---

then we would have continuous temperature sampling once the light goes over 10000lx.

We label as a *non recurring data instance (NRDI)* a non-recurring stored data recording.

Defining an NRDI

---

NRDI: (T, param, resolution, compression), where:
T: recording time
param: the parameter being collected
resolution: how precise the stored value is
compression: boolean stating if the NRDI has been compressed
e.g., NRDI(2009/12/06 16:43:23, temperature, 0.01C, FALSE)

---

*3.3 Primitives for space optimization*

In this section, we introduce primitives which are invoked inside the PDR Engine once the PDR Controller has identified data to be subjected to reduction.

*3.3.1 Compression*

Lossless data compression algorithms are widely available and used [11]. On a normal basis, compression and decompression cause little impact. On a sensor node, compressing certain portions of the data will yield available data space with no information loss. The loss is from a flexibility perspective. Once compressed, the data becomes a blob which should be treated as an atomic entity. The WSN node looses the capacity to discard partial data.

Such an approach is recommended for very critical data, and hence very important, that can never be discarded. Otherwise, it should be used for non-recurring data instances, or for portions of recurring data collections that can be dropped one whole section at a time. A parameterized compression is used to specify which span of a data instance should be compressed:

*Compress[(a, b)](RDI) signals that only the data from time interval a to b is compressed, while the rest remains as initial.*

The non-parameterized compression affects the entire RDI.

---

Usage on non-recurring data:

Compress(NRDI(2007/11/24 13:21:37, humidity, 0.01%, FALSE)) = NRDI(2007/11/24 13:21:37, humidity, 0.01%, TRUE)

Usage on portions of recurring data by first dividing the data instance several data instances:

Compress[(2008/11/11 12:00:00, 2008/11/12 12:00:00)](RDI: (2008/11/10 12:00:00, 2008/11/13 12:00:00, temperature, 3600 seconds, 0.01C, FALSE)) = RDI: (2008/11/10 12:00:00, 2008/11/11 12:00:00, temperature, 3600 seconds, 0.01C, FALSE) + Compress(RDI: (2008/11/11 12:00:00, 2008/11/12 12:00:00, temperature, 3600 seconds, 0.01C, FALSE)) + RDI: (2008/11/12 12:00:00, 2008/11/13 12:00:00, temperature, 3600 seconds, 0.01C, FALSE)

At this point, the second data instance can undergo compression and becomes:

RDI: (2008/11/11 12:00:00, 2008/11/12 12:00:00, temperature, 3600 seconds, 0.01C, TRUE)

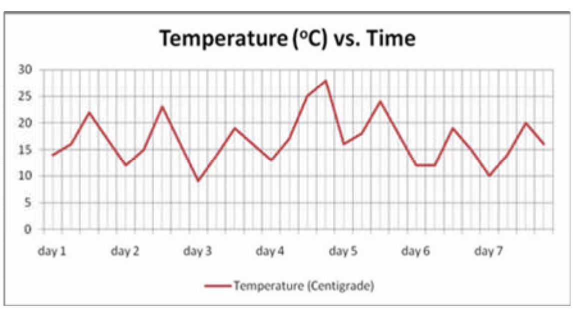
while the first and third data instance retain all original data.

---

### *3.3.2 Thinning*

For a non-recurring data instance, thinning involves simply discarding the collected data. For recurring data, thinning involves discarding a contiguous amount of data that corresponds to a time span of low importance in the case of recurring data instances. This can be used when the collection has a cyclic pattern and a long sampling period gives little additional insight when compared to a somewhat shorter period, or a period with gaps.

To better clarify, we can resort again to the temperature sampling example. Let's assume that we are sampling temperature every minute. This has been going on for five months. Depending on the business case, it may be acceptable, without any significant impact to data significance, to either discard data pertaining to the third operational month, or to discard data pertaining to the third quarter of each operational month. Figure 5 shows the impact of thinning on a data sample.



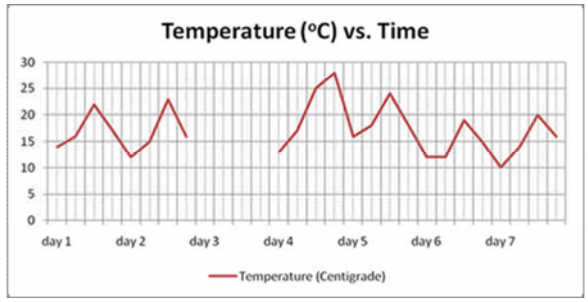




Figure 5. Impact of thinning on data top) initial data bottom) resulting data after thinning.

*Thin[(a, b)](RDI) signals that the data collected between time a and time b are dropped from the data instance.*

---

Usage of thinning on recurring data:

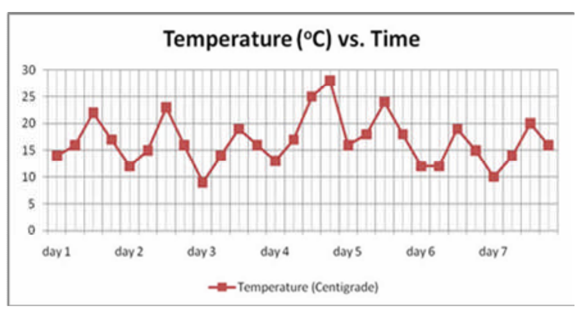
Thin[(2008/11/12 12:00:00, 2008/11/13 12:00:00)](RDI(2008/11/10 12:00:00, 2008/11/17 12:00:00, temperature, 6 hours, 1C, FALSE)) = RDI(2008/11/10 12:00:00, 2008/11/12 6:00:00, temperature, 6 hours, 1C, FALSE ) + RDI(2008/11/13 12:00:00, 2008/11/17 12:00:00, temperature, 6 hours, 1C, FALSE )

---

### *3.3.3 Sparsing*

Sparsing can only be used recurring data collections. If the global pattern of fluctuation in the measurement is an important factor, then it is important not to lose entire spans of information. In such cases, the recurrence window can be widened by means of dropping values at regular intervals. The resolution suffers, but the overall pattern is conserved.

Looking at the same example as in 3.2, instead of dropping a full data set corresponding to day 3, we are going to double the sampling interval for days 2 and 3.



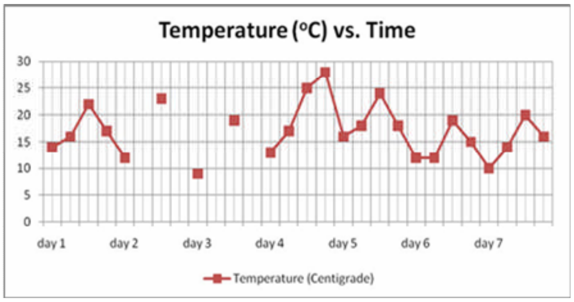



Figure 6. Impact of data sparsing top) on the left with initial data bottom) on the right after sparsing is applied.

*The syntax for sparsing is Sparse[(a,b,r,s)](RDI), which means that for the time interval between a and b, r entries are removed out of every s entries.*

---

Usage of sparsing on recurring data:

Thin[(2008/11/12 12:00:00, 2008/11/14 12:00:00, 1, 2)] (RDI(2008/11/10 12:00:00, 2008/11/17 12:00:00, temperature, 6 hours, 1C, FALSE)) = RDI(2008/11/10 12:00:00, 2008/11/12 6:00:00, temperature, 6 hours, 1C, FALSE) + RDI(2008/11/12 12:00:00, 2008/11/14 6:00:00, temperature, 12 hours, 1C, FALSE) + RDI(2008/11/14 12:00:00, 2008/11/17 12:00:00, temperature, 6 hours, 1C, FALSE )

---

*3.3.4 Grain coarsing*

Data collections can be with very high precision, or can be with lower precision. For example, captured images can be anywhere from black and white to 24 bit images. To a lesser extent, this can be applied to numeric data which contains more precision in an 8 byte floating point representation versus a 2 byte integer representation.

Grain coarsing can be applied to both non-recurring and recurring data. Just like thinning and sparsing, we can opt to apply grain coarsing only to a specific interval of an RDI.

Because precision or resolution depends on the nature of the data, we give the example of an image whose resolution is measured in dpi. To specify a 50% decrease in DPI, the exact call would be GrainCoarse[(50%)](NRDI).

---

Usage of grain coarsing on an image:

GrainCoarse[(50%)](NRDI(2007/03/23 17:32:45, image, 300dpi, FALSE)) = NRDI(2007/03/23 17:32:45, image, 150dpi, FALSE)

---

*3.3.5 Range representation*

In some cases we would like to eliminate most of the collected data for a specific parameter over a certain time range, yet still keep some hint of where values were for that range. In such an instance, we can keep for example the minimum value, the maximum value, the average, as well as the number of values used to compute the average and how far apart in time those values were. Note that the minimum and maximum values are not necessarily correct, but they are the largest and smallest value as far as available data samples.

A case where this can be misleading is for example a recurring data sampling that has undergone extensive sampling. In that case, it is plausible that many of the brief spikes and dips in values are lost.

The range primitive applied to an RDI produce a tuple: Range(RDI) = Tuple($T_{start}$, $T_{end}$, Min, Max, Average). A data span that is represented as a tuple can be considered for further space optimization in the same manner as an NRDI.

*3.6 Usage of primitives*

The proposed space optimization methods are primitives. They are not the last word on a specific data collection. It should be noted that the same data can be subjected multiple times to data reduction, and each time a different mechanism can be deemed appropriate. While data morphs, its importance changes as well.

We take the example of a single sensor node that is deployed in a dangerous climate area. The node is to collect several parameters over a long period of time. At the end, the node is either removed or a sink node is placed in proximity for a brief period in order to retrieve collected data.

The parameters that are to be surveyed are: temperature, barometric pressure, light, humidity, vibrations, and CO2 concentration. There are several recurring data collections as well as triggered single time collections.

We grade data importance from 0 to 1 as a continuous value. The value of 1 is reserved for critical data that under no circumstance should be reduced in size.

Critical data involves a full day's unaltered temperature and barometric pressure sampling every second on the first operational day, and every 10 minutes thereafter with importance 0.5. Except for the first day, recurring recordings of these two parameters are subject to 50% sparsing while at the same time their importance increases by half the interval between current importance and 1.

Whenever a vibration amounting to 2$^{nd}$ degree on the Richter scale, all parameters are collected for one hour at 5 second intervals and bear importance 1. This collection is followed by 24 hours of collections every minute with importance 0.8. This collection can be subjected to range representation based on 10 minute intervals, which data now becomes of importance 1.

With this data set and requirements, the simulation is allowed to proceed both with data reduction and no data reduction. Here is what is found in each case:

*3. 6.1 No data reduction*

Under an approach of no data reduction, the storage space was used up in about 12 operational days. The following were found in the data storage:

- First day of temperature reading every second
- First day of barometric pressure reading every second
- Eleven days of temperature readings every 10 minutes
- Eleven days of barometric pressure readings every 10 minutes
- Data collections related to three vibration shocks: all parameters collected for one hour at 5 second intervals, and 24 hours every minute

*3.6.2 With data reduction supported by proposed primitives*

While allowing for the data reduction, here is what was found on the machine after 15 operational days:

- First day of temperature reading every second compressed
- First day of barometric pressure reading every second compressed
- Fourteen days of temperature readings every 10 minutes out of which the first six had been sparsed (i.e. are now every 20 minutes)
- Fourteen days of temperature readings every 10 minutes out of which the first six had been sparsed (i.e. are now every 20 minutes)
- Data collections related to five vibration shocks, two of which have been subjection to range representation

With all of the above in storage, there are still items that can be removed to make space for incoming data.

*3.6.3 Comparison for freeing the space*

Clearly the data reduction presents opportunity for storing more relevant data in the long run. There are some drawbacks regarding the resolution and recurrence of collected data, but the emphasis is placed on high importance data.

Now that we have proposed procedures for data reduction, the question is when such data reduction should be performed. There are competing constraints to consider:

- We do not want to run the data reduction process too often
- We do not want to get overenthusiastic with data reduction as the sink may soon be available for integral data transmission

The problems faced here are similar to garbage collections in processes such as JVM. The added complexity is that, as opposed to JVM garbage collector which collects true disposable garbage, the process we intend to run recovers space in exchange for some loss of lower importance data or loss of flexibility.

An assumption is made that while the storage recovery process is under way, the sensing devices have enough buffer space to store sensed data until the main processor is ready to take the data to the main storage unit.

*3.6.3.1 The Pessimistic Approach*

The main idea of a pessimistic approach in running a data reduction process is that the space available must always be able to accommodate the biggest possible data influx spike which cannot be dropped based on its importance. This rule can only be broken if the existing stored data has an importance level which makes it final and not subject to reduction.

Let’s denote the total storage space with T, the occupied space by O, and the biggest possible influx spike by S. It follows that $(T - O) >= S$. Technically we can get away with $T - O = S$. However, we risk fluctuating values of O which leads to a repeated invocation of data reduction algorithm. The free space target depends highly on business logic and should be a value higher than S.

*3.6.3.2 The Optimistic Approach*

The optimistic approach keeps all data until it is time to store higher importance data. At that point, storage space will be made available by running a storage space recovery process while the incoming data is still in the sensing device buffers. For this approach, the assumption is made that data reduction is a quick process both in identifying the data subject to reduction, as well as the reduction process. In certain cases this may be true, and hence there is no need to have available space sitting unused just in case there is incoming data that needs to be stored.

## IV. EXTENDING THE SINGLE NODE FUNCTIONALITY

In this section, we take as a starting point the Prioritized Data Reduction (PDR) presented in Section III and expand it to include a complete characterization of the data reduction mechanisms. The overall model of the sensor network under consideration is “dumb data collection”: collection is done by the nodes, processing is done offsite. At this point, data deduplication is not addressed as part of PDR; it is left as future work.

In Section III data importance was introduced as a continuous value assigned to collected data instances used in the data reduction process. The lowest value is 0 and the highest value is 1. In order to assign this value to collected data, the data needs to be divided in manageable pieces. Each such piece, called a *data unit*, has two associated values: *data importance* and *compensation factor*.

### *4.1 Data Units*

Data collections can be both recurring and non recurring. The non recurring collections generate data that is stand alone and considered atomic. It makes sense to consider a non recurring data record as a single data unit.

Recurring data collections have several values over a potentially long period of time. Such a data needs to be divided in intervals that can be treated as a whole during the process of data reduction. We propose that such intervals be devised as contiguous time intervals. There is no need for the intervals to be of equal length. What the data units do is provide a contained set of data on which we can apply the primitives enumerated in Section III.

For the sake of simplicity, we also assume that once a data unit has been delimited within a recurring data collection, it is no longer subject to further division or merger with other data units. There may be flexibility gains from allowing such operations, but this is left as future work.

The lifecycle of a data unit starts with a data collection, which can potentially go through several rounds of data reduction, and ultimately possibly dropped, assuming a long period of sink node unavailability. If at some point, the sink node is available, data is simply unloaded and space freed.

To assist in selecting what data unit to target for data reduction, each data unit is reflected by a data importance, denoted I. Once a data reduction is performed, this needs to be reflected in future importance computations. For this purpose, each data unit is assigned a compensation factor, K.

### *4.2 Data Importance*

Data importance, denoted as I, is a value that numerically reflects the relevance of a data unit for the business case. The raw value of I is primarily used to rank data units in view of data reduction, but the exact magnitude of the difference does not necessarily carry a meaning.

```
START
WHILE ‘storage availability level’ is below a ‘threshold’
    FIND ‘data unit ‘ with ‘lowest importance’
    APPLY ‘data reduction’ to that data unit
    ADJUST the compensation factor
    RECOMPUTE ‘importance’ for any dependent data unit
```

Reduction algorithm 1. High level PDRE process for data reduction.

Figure 3 shows an abstraction of the process that happens within the PDRE when the controller triggers a data reduction operation. The reduction can occur in one or more iterations. In the first step, the data unit with the lowest importance is found and the corresponding reduction method is applied. As a consequence, the compensation factor is adjusted. All other data units whose importance factor depends on the reduced data need to have their importance level recomputed.

### *4.3 Compensation Factor*

The compensation factor, K, is an importance modifier that reflects the data reductions that have already been performed on the specific data unit. The compensation factor, is defined as a percentage value between -100% and +100% and it is used in the following manner:

$$I = \begin{cases} \text{if } K < 0, (1+K)*I' \\ \text{if } K > 0, I' + K*(I_{max}-I) \end{cases}$$

where I is the data importance for a data unit, I’ is the resulting data importance post data reduction, and $I_{max}$ is the maximum possible value for data importance. In our example, we assume that data importance is continuous between 0 and 1; 1 is reserved for data that cannot be reduced under any circumstance.

To illustrate the application of the compensation factor, let’s take an example where a data unit’s importance value is evaluated to 0.75 before the compensation factor is applied.

- for a compensation factor K = 40%

  $I = 0.75 + 0.40(1 – 0.75) = 0.85$

- for a compensation factor K = -40%

  $I = (1 + (-0.40)) * 0.75 = 0.45$

### *4.4 Available Input*

Determining data importance is in the hands of the business model. In this section, we identify input factors that can be used to establish the importance of a data unit.

- *age/collection time*

The data collection time is a relevant factor. Whether in a linear manner, exponential decay, or in some irregular manner, the collection time is relevant. For example, one may be interested to correlate rush hour to atmospheric $CO_2$ levels over a month’s period. In such a case, the time dependency is rather irregular: the data collected just before, during, and just after rush hour are more important than data collected on a week-end.

- *self values*

The very collected values can affect the importance of a data unit. For example, for any parameter that is collected, any value outside of an expected interval may need further investigation. Hence, additional importance can be assigned to data units containing such values.

- *other data units of same instance*

In cases where a parameter's cyclic aberrant values are suspected, the values of a data unit affect the importance of data in other data units of the same data instance.

- *other data instances*

Similar to the case above, if correlation is to be made between multiple parameters, then special values of interest in a data unit of one parameter add importance to data units taken around the same time for other parameters measured.

### *4.5 Summarizing the components of RDI and NRDI*

In order to make the jump from a single node model to a multiple node deployment, we have identified additional functions to better characterize data handling. In this section, we will review the components added to the model described in [12].

- interval production function (for RDI only)

For a recurring data instance, this function is used to produce data units. This function can be rather simple, such as grouping every fixed number of measurements in data units, or it can be more complex resulting in the creation of data units covering variable time spans.

- default compensation factor

The compensation factor is used to reflect already performed data reduction. A default value needs to be specified.

- data importance function

A data importance function which computes the importance of a data unit, taking into account some or all of the factors listed in section A.

- data reduction operation function

A data reduction function which, given a data unit outputs a smaller sized data unit based on its internal use of reduction primitives. It also changes the compensation factor so as to reflect upon future data importance computations.

### *4.6 A single node use case*

Given the above extensions to the single node model, we now consider an example and get into details as far as the specifications for data collection. The approach to data collection reflects human perception with respect to quantifying and formulating data importance. In order to validate such approaches, a preliminary step is to simulate the setup intended for deployment.

#### *4.6.1 Car traffic and $CO_2$ concentration use case*

The main objective of the deployment in the example is to observe data that can be used to correlate car traffic volume with atmospheric $CO_2$ concentration. There is a broad pattern that is expected, with more $CO_2$ production during rush hours where there is more traffic. There are also temporary spikes in the detected levels when a large vehicle passes, such as a large truck. These spikes need to be accounted for.

The deployment consists of many sensors installed in a very large geographical area, but all in proximity of a street. Some of the sensors may be able to reach other for communication, but communication is very costly given the distance. The collection happens using a mobile base on a vehicle with. The frequency of data collection is not predetermined.

The sensors that are deployed have four sensory devices in addition to an internal clock: a gauge for $CO_2$ concentration, a sensor across the street to detect traffic levels, as well as to detect the presence of an oversized vehicle, a gauge for wind speed, and one for wind direction. The sensor is also able to take a panoramic photo and store the picture.

#### *4.6.2 Data Collection Requirements*

- Every 10 seconds, the $CO_2$ concentration is recorded continuously (data instance **A**)
- Matching the above, wind speed and direction is recorded continuously (data instance **B** and **C**)
- When a large car passes, the $CO_2$ concentration is recorded for 5 minutes at 1 second intervals (data instance **D**)
- Matching the above, wind speed and direction is recorded continuously (data instance **E** and **F**)
- If an unexpected spike in $CO_2$ is detected, a panoramic image is recorded (data instance **G**)
- The number of cars passing is recoded for every 10 second intervals (data instance **H**)

#### 4.6.3 Data Reduction Specification

Each of the above data collections need to have specific interval production functions to generate data units, default compensation factor, data importance functions, and data reduction functions.

- data instances **A**, **B**, and **C**

The division in data units is done according to the time of day. The regions of higher interest (i.e. rush) are divided in smaller data units. That way, any data reduction algorithm applied to a data unit affects a smaller data interval.

**Table1.** Data unit size for data instances A,B, and C

| time | label | data unit size |
|---|---|---|
| 9pm – 4am | night | 10 minutes |
| 4am – 6am | pre rush | 2 minutes |
| 6am – 9am | rush | 30 seconds |
| 9am – 11am | post rush | 2 minutes |
| 11am – 2pm | day | 5 minutes |
| 2pm – 4pm | pre rush | 2 minutes |
| 4pm – 7pm | rush | 30 seconds |
| 7pm – 9pm | post rush | 2 minutes |

For the purpose of defining compensation factor, the reduction function, and the data reduction function, we divide the collected data into four sections. The first three sections span 7 days each, and the fourth section covers the remaining time interval. The days being labeled from 0 (most recent), here are additional parameters for the data units:

**Table 1.** Data handling parameters for instances **A**, **B**, and **C**

| span | K | data reduction | data importance |
|---|---|---|---|
| day 0 – day 6 | 50% | 25% trimming | set to 0.95 |
| day 7 – day 13 | 25% | 50% trimming | set to 0.85 |
| day 14 – day 20 | 0% | 75% trimming | set to 0.65 |
| day 21 - end | -25% | 75% trimming | set to 0.5 |

The use of K and the data reduction trimming primitive have been covered above. For the data importance, whenever a data unit moves from one span to another, its importance is set to a fixed value (as noted in the table). Only subsequent data reduction operations will affect this (via the compensation factor).

- data instances **D**, **E**, and **F**

In this case, we select a constant data unit interval for the entire span of the data collection. The data collection is rather shot, so the entire 5 minutes will be treated as a data unit.

For the remaining parameters of a data instance, we divide such instances into the 10 most recent, following 100 most recent, following 1000 most recent, and as a fourth, the remaining instances.

**Table 2.** Data handling parameters for instances **D**, **E**, and **F**

| span | K | data reduction | data importance |
|---|---|---|---|
| 10 most recent | 20% | 50% sparsing | set to 0.9 |
| next 100 | 20% | 50% sparsing | set to 0.75 |
| next 1000 | 20% | 50% sparsing | set to 0.50 |
| after1000 | -20% | 50% sparsing | set to 0.50 |

The same approach as above is used in the case of data importance.

- data instance **G**

This data instance is non-recurring. A recording of this data is a data unit. Similar to the case above, we divide the instances in the 5 most recent, the following 10 captures, and the rest of the captures:

**Table 3.** Data handling parameters for instance **G**

| span | K | data reduction | data importance |
|---|---|---|---|
| 5 most recent | 50% | 20% resolution reduction | 1 |
| next 10 | 50% | 20% resolution reduction | 0.8 |
| after1000 | 50% | 20% resolution reduction | 0.7 |

- data instance **H**

In this instance, we are dealing again with continuous data collection. In this case, counting the vehicles, we divide the collection in even data units. Just like in the first example, the data units are placed in several time spans: three spans are 30 days long, and the fourth span contains the remaining days. The data units are 5 minutes long.

**Table 4.** Data handling parameters for instance **H**

| span | K | data reduction | data importance |
|---|---|---|---|
| day 0 – day 29 | 25% | 50% grain coarsing | set to 0.80 |
| day 30 – day 59 | 15% | 50% grain coarsing | set to 0.70 |
| day 60 – day 89 | -15% | 50% grain coarsing | set to 0.50 |
| day 90 - end | -50% | 50% grain coarsing | set to 0.40 |

#### 4.6.4 Use Case Conclusions

The use case described is quite simple, yet it does the job of properly collecting the data. The model used needs the validation of intense simulation in order to validate the nature of the data it produces in various circumstances.

### 4.7 Data dependency

In the example so far, there was no dependency between the different data collections with regards to data values from a concurrent data collection. In this section, we expand on data instance A from the previous section in relation to data instance B. As a reminder, data collection A is the $CO_2$ concentration sensed, while B is wind speed.

As a business case decision, we decide that certain atmospheric disturbances in the area of the sensor affect the relevance of the $CO_2$ concentration measurements. We are going to look at two aspects to determine dependency: (i) increase in wind speed and (ii) average wind speed within a data unit. We decide that wind acceleration beyond a fixed value *a* affects the importance of $CO_2$ measurements, while an average wind speed beyond a certain level *s* affects the importance of $CO_2$ measurements for four data units.

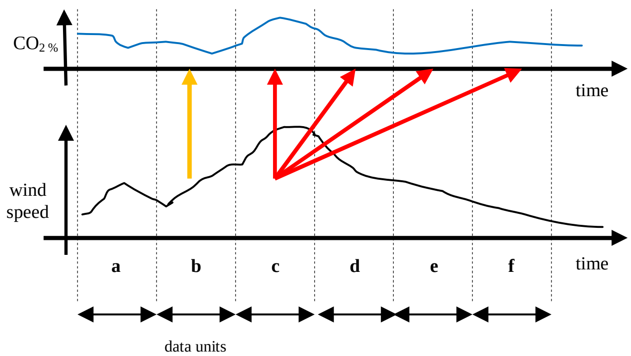



Figure 7. Data dependency across data units.

In Figure 7, several data units, labeled from a to f, are shown with corresponding reading of $CO_2$ concentration and wind speed. Data units b and c on the wind speed graph are of interest as they affect the importance of $CO_2$ readings. In data unit b, the wind acceleration causes an effect on the corresponding concentration reading. In data unit c, the average wind speed affects the data importance of four data units for concentration reading.

On a more specific note, here is a proposal for to compute the data importance mapped to a data unit of $CO_2$ concentration reading. We denote by *CO du* $t_i$ - $t_j$ the $CO_2$ concentration data unit between time ti and time tj, while *WS du* $t_i$ - $t_j$ refers to the wind speed data collection.

var tempI

if age(du $t_i$ - $t_{i+1}$) < 7 days
    tempI ::= 0.95
else if age(du $t_i$ - $t_{i+1}$) < 13 days
    tempI ::= 0.85
else if age (du $t_i$ - $t_{i+1}$) < 20 days
    tempI ::= 0.65
else
    tempI ::= 0.50

if acceleration(WS du $t_i$ - $t_{i+1}$) > *a*
    tempI ::= tempI * 50%

if average(WS du $t_i$ - $t_{i+1}$) > *s*
    tempI ::= tempI * 10%
if average(WS du $t_{i-1}$ - $t_i$) > *s*
    tempI ::= tempI * 20%
if average(WS du $t_{i-2}$ - $t_{i-1}$) > *s*
    tempI ::= tempI * 50%
if average(WS du $t_{i-3}$ - $t_{i-2}$) > *s*
    tempI ::= tempI * 80%

I(COdu $t_i$ - $t_{i+1}$) :: = tempI

Figure 8. Computing data importance with dependency.

At this point, we can generalize the constraints on the data importance computation as pertaining to two categories: (i) internal constraints and (ii) external constraints.

External constraints are caused by factors over which input data has no effect. Such factors are data age and inherent interest in the data depending on the exact purpose of the data collection.

Internal constraints represent inter- and intra-data dependencies. In the case presented above, increased wind renders $CO_2$ concentration less relevant and less usable; there can be events which affect the importance of data at any point along the timeline.

### *4.8 General implementation architecture*

In this section, we present the general building blocks of the data collection process, as well as the prioritized data reduction process. While these processes act on the same data, they run in parallel as uncoupled processes.

Figure 9 presents the control steps for the data collection. Figure 10 presents the steps involved in the data reduction process. The black arrows denote sequence of steps. The red arrows denote control.

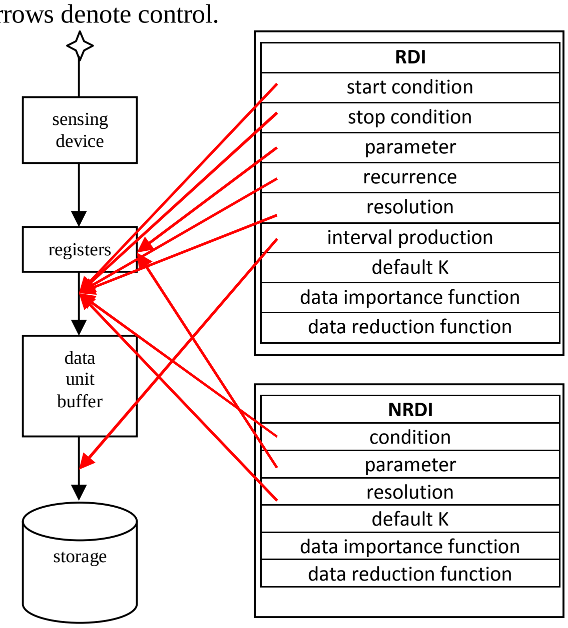



Figure 9. Storage management as per instance specifications.

In Figure 9, we have the physical sensors which make actual measurements. They have associated registers where the data readings are placed. This happens whether data is being collected or not. At this point, depending on the running condition of the recurring data instances, the recurrence interval, and the resolution, the data is captured into a buffer. Enough data is accumulated to construct a full data unit. When complete, the data unit has its importance calculated and is placed into the storage. In case we are following an optimistic approach to data reduction, as described in [12], then we compute the data importance of the data unit in the buffer in order to compare with stored data. In case the data unit in the buffer is the one with lowest importance, then it will be the one to be reduced.

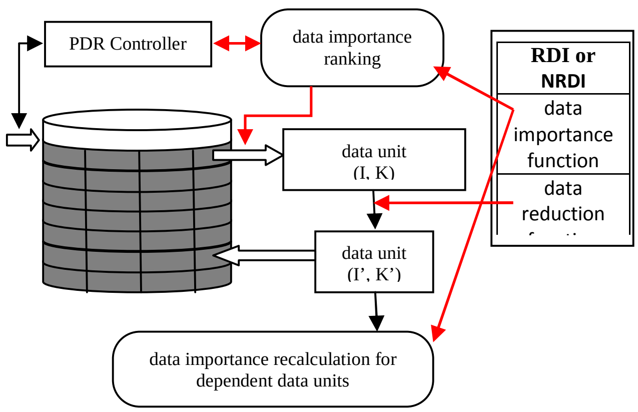



Figure 10. Data reduction based on data instance specifications.

Figure 10 only shows a single sensing unit and a single data instance. In deployments, there can be several sensing units for different parameters, and several data instances that are collected. There can be different instances collecting from the same sensor, but with different specifications, such as resolution for example.

In Figure 10, the flow of data reduction processing is presented. The PDR Controller continuously monitors the amount of data in the storage. If the levels cross a given threshold, then data importance ranking is consulted to identify the least important data. This data unit maps to a specific data instance which carries a data reduction function as well as an importance computation function. After a reduction, the data unit is written back to memory.

## V. DATA HANDLING

### *5.1 Multi-prong consideraitons*

#### *5.1.1 Energy considerations*

When encountering limited data storage space, a node has two options, (i) to reduce the size of the data by applying some primitives, or (ii) to relocate some data. In this section, we consider the factors that affect the decision.

Negotiations to find an appropriate host for data require energy. As the packets involved in negotiations don't contain data, they are fairly short so they don't consume much energy. The energy cost needed for the transfer of the data depends directly on the size of the data transferred. The node needs to save enough energy to eventually transfer the entire contents of the data to the sink node once the sink node becomes available.

Energy is wasted if a portion of data is relocated only to be transferred to the sink shortly after. In the case of unreliable sinks or sinkless deployments, expected access to sink is hard to compute. It can be done with statistical data, but reliability can vary. If the sink operates on a predefined schedule, it is feasible for this to be taken into consideration. A similar case to a predefined uptime schedule is a mobile sink. With some uncertainty, the sink will travel along a route with high probability of being in specific areas at certain times.

#### *5.1.2 Data considerations*

The There are characteristics that make data appropriate for relocation. The size is one factor, as we want to have a sizable impact as a result of investing energy in the negotiation part. Other factors relate to the dependency between data units.

The importance calculation function should depend on self values or on time only. In that way, we can preserve the importance calculation function after the data has been moved. In some cases, the importance of a data unit depends on other data units of the same collection. It is conceivable that all the data units with importance interdependence relations be subject to a data transfer as a whole.

The reverse is also true, i.e., the data units to be transferred are used in computing the importance level of other data units. Bundling everything in the data to be relocated is an acceptable solution.

A special case is the situation where yet uncollected data can affect the importance of data units already collected and tagged for relocation. Such data is best not moved as this would imply changed to the importance calculation function.

The parameter K was introduced in Chapter 4 as a compensation factor to affect the importance value for data units that have undergone data reductions. A positive K will increase the importance of post reduction data, while a negative K will decrease it. The data that is considered for a move needs to have a positive K so as to make it through potential data reductions it will undergo on the receiver node.

#### *5.1.3 Receiver node considerations*

The conditions of the receiver node affect how the received data will be handled on that node. The potential recipient of the data needs to have a good situation of its storage space as well as a good outlook for future situations. The recipient also needs to have a good amount of energy left so as not to compromise the long term survival of the data being sent.

The number of active collections and the amount of data they produce is an important factor. If there are many collections going on, the space on the receiving node is quickly filled up. Even if these collections produce low importance data units, a large number of ongoing collections generate lots of data reductions and data relocations. We want to avoid causing such events.

Inactive collections are collections whose start condition is not met. Some of these are predictable as to when collection actually starts, such as collections that only depend on time to be activated. Others depend on data values detected by sensors. Having a large number of inactive collections makes a node less desirable to act as a data recipient. These collections can start unexpectedly and produce data that competes for space with the relocated data.

Finally, the general importance profile of the stored data units are a factor in deciding if a node is a good recipient of relocated data. If most data on the receiver node is low importance, then the node is a good recipient of data. However, the nature of the importance functions on that node is to be taken into consideration. An event could trigger a recomputation that changes the data importance to higher values.

*5.2 Sending vs. Reducing: the decision*

The decision to favor data reduction vs. data relocation is subject to network's owner decisions. In this section, we provide a parameterized approach to handle this task. We specify three functions as follows:

- **SF(i)** specifies the fitness of a node *i* to act as a data source
- **RF(i)** denotes the fitness of a node *i* to act as a data receiver
- **DF({du})** denotes the fitness of a set of data units for a data transfer operation

We also specify a threshold value t such that:

*if (a*SF(n) + b*RF(m))*DF({du}) > t, then node n should initiate a data transfer of the {du} set of data units to node m, where a and b are parameters to give more or less weigh to each of the functions*

**SF(i)** is defined as a function of the source node's ability and opportunistic interest to spend energy on data movement related operations: negotiations and actual data movement. We define as LD(i) the percent of data stored on node *i* having an importance that is low and decreasing, that is, an importance smaller than a fixed value, decreasing in time, and a negative K. The K does not have to be negative for a fixed number of initial reductions, but it needs to be negative for remainders of the reductions or the reductions are quickly reaching a point where data is entirely dropped. Let *e* be the minimum residual energy required for a node n to send its data contents and *E* the current amount of energy in storage. The value of SF(i) directly related to available energy and to the lack of low and decreasing importance data units. We define SF(i) as follows:

$$SF(i) = k*(1-LD(i)) + j*(E/e) \qquad (1)$$

where k and j are parameters to adjust the weights of the two factors.

**RF(i)** is defined as a function to quantify the potential receiver's node fitness to act as a receiver. With respect to energy, RF(i) behaves just like SF(i). With respect to the stored data profile, it behaves in an opposite manner. In addition to these two aspects, RF(i) needs to account for the effect of active collections as well as inactive data collections.

$$RF(i) = k*S*LD(i) + j*(E/e) - k*(Da + p*Di) \qquad (2)$$

where Da is the storage debit of active collections and the Di is the potential storage debit of inactive collections. The constant p is used to decrease the impact of inactive collections.

**DF(i)** reflects on the fitness of data to be subjected to transfer. As a limiting factor, we have established that the bundle of data units must be self enclosed as far as importance calculation dependency. We propose the following formula for evaluating a data unit's fitness for transfer:

$$DF(i) = I*(trend(K) - k*|M - s|) \qquad (3)$$

where M is the ideal data transfer size, s is the size of the current data slated for transfer, k is an adjustment parameter, and trend(K) is a function that evaluates the value trend of K in time and as potentially affected by reductions within the transferred data. The entire result relates to the overall importance I of the data bundle considered for transfer.

With these formulas, we have captured the essence of the decision making process of data transfer vs. data reduction. The approach is heavily parameterized as weighing different factors varies from one business case to another (Figure 11).

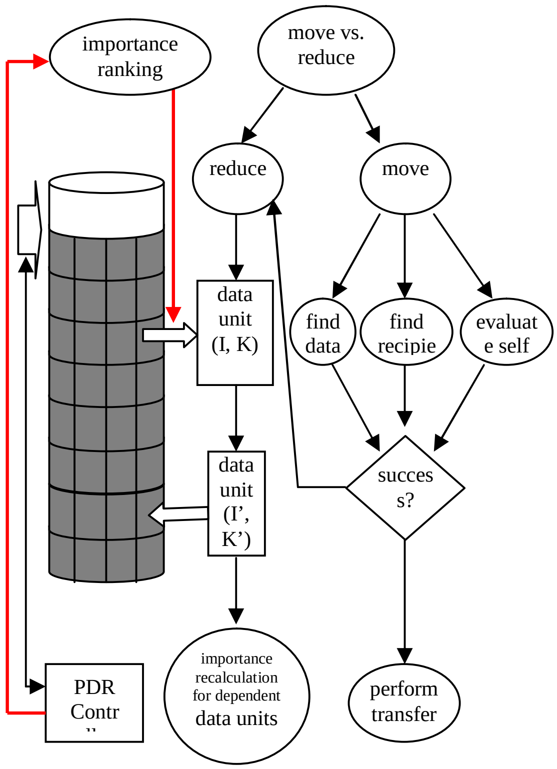



Figure 11. Data reduction via move or prioritized reduction.

We can assume that process to evaluate the source not for fitness to participate in a data send is straight forward: the value of the SF(i) is computed. Finding a recipient and finding appropriate data to send are more complex as they involve selecting from a pool of possibilities.

Finding data to transfer is a complex process. There can be a large number of data units, and therefore, considering all possibilities can be a lengthy process. We propose a heuristic on how to approach data targeting for relocation with the understanding that an exhaustive process may offer a better solution but at a higher cost. Data units that are targeted for transfer are incrementally added to the set of data for transfer as we seek to fulfill the requirements for self-enclosure for importance function evaluation. We stop as we reach a data size in the vicinity of M. In certain cases, we reach a dead end and can no longer have hope to find appropriate data to move.

The set of data units to be moved is incrementally increased in steps as follows:

- high importance and positive K data units on which no other data units' importance depends
    - o data units' importance dependence limited to self values
    - o data units' importance dependence limited to self values and time
- high importance data units with positive K whose importance depends only on other data units already collected (i.e., future data units collected have no impact on the importance of this data)
- recursively add the data units on which the above units depend for importance calculation
- as we reach a value close to M, we stop adding data units to the set of data targeted for a data transfer

We realize that recursively adding data units may be getting out of hand. If the minimum amount of self dependent data units amount to a very large amount of data, we can safely give up the idea of a transfer and opt for a data reduction.

The simplest approach to finding a recipient for data is to query the status of all reachable nodes, evaluate their RF(i) function and select the best option. Depending on the deployment, querying the immediate neighbors only may be artificially limiting the amount of potential recipient candidates for data relocation. Attempting to go the extra distance via additional hops requires the cooperation of other nodes along the path. These nodes need to commit energy reserves towards the data relocation process. Given that we have already selected the data that will be moved, there is a clear expectation as to what amount of energy would be required from the transient nodes during the data move.

Finding an appropriate receiver for the data follows the following steps:

- decide how many hops away we want the data to potentially go
- broadcast a message to the effect of finding a host, including the number of hops; if more than immediate neighbors are considered, include the data size to be able to obtain commitment along the transfer route
- reachable nodes reply with their current RF(i) computation
- select the best RF(i), compute (a*SF(n) + b*RF(m))*DF({du}) (see section 5.2.6) and if result is satisfactory, proceed with the transfer
- if the computation yields an unsatisfactory result, then data reduction is undertaken instead of transfer

*5.2.1 Validation and calibration of heuristics*

In the above section, we have proposed a series of heuristics to quantize the fitness of nodes to act as receiver or sender, as well as the fitness of data to be transferred. The proposed formulas are parameterized so they can be tweaked to several circumstances. In this section, we are going to evaluate the trends that the formulas generate under different circumstances depending on the statistical distribution of importance and compensation factor of stored data units

We recall that the formula to decide the fitness of a node to act as a sender is:

$$SF(i) = k*(1-LD(i)) + j*(E/e)$$

By its definition, the fitness function depends on the raw amount (not ratio) of high importance data and level of energy required to relay the entire data load towards the base station. We consider the scenario of different statistical distributions of data importance on a node, and in each case, we vary the amount of energy available. We consider the cutoff for LD(i) to be at the importance level of 0.5. Hence, in the graphs below, the blue bars represent (1-LD(i)).

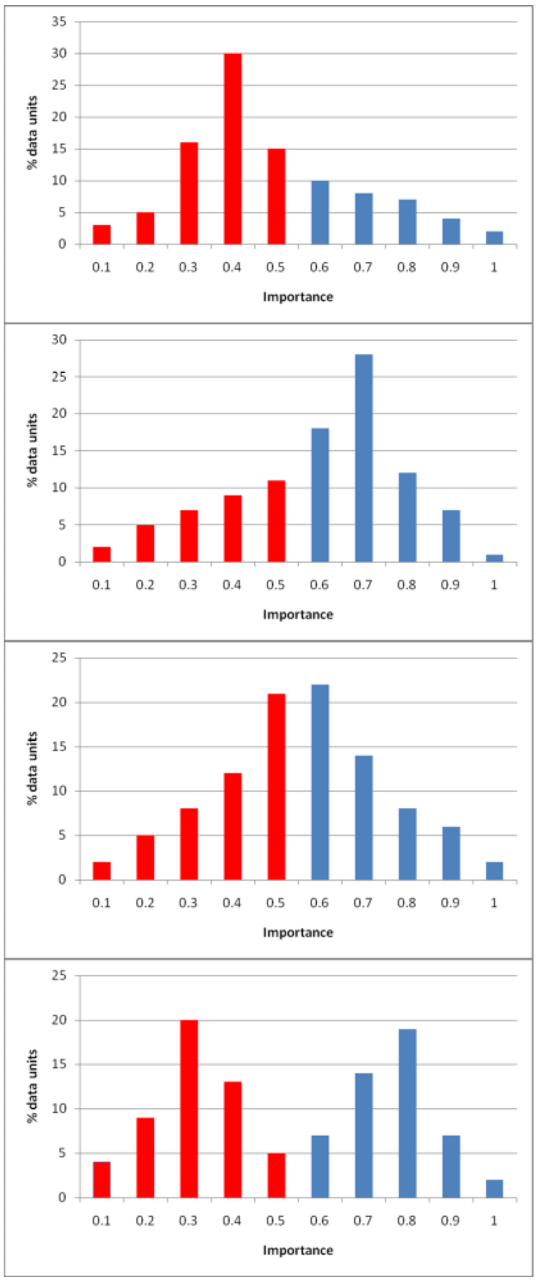



Figure 12. Data distribution scenarios for validation of SF(i).

The image above shows the scenarios considered with respect to the distribution of importance values. We have considered skewed, bell curve, and bimodal.

The initial proposal was that the amount of energy available would have a linear impact on the value of SF(i). This works well at points where the available energy levels E are within a few orders of magnitude of *e*, the energy needed for a full data transfer. At higher orders of magnitude (i.e orders of thousands, and above), the linear growth of j*(E/e) poses a problem as it identified a node as a "good node to send data" mostly based on its energy levels. If the nodes deployed fall in such a class, then the problem can be alleviated by using a logarithm to attenuate de higher orders of magnitude:

$$SF(i) = k*(1-LD(i)) + j'*\log(E/e) \tag{1'}$$

Experiments with different LD(i) values only changes the balance between defining good source nodes via SF(i) and balanced by the ability to find receiver nodes via RF(i) defined in equation 2.

Along the same lines, simulations were run to further analyze the proposed formula for RF(i). We notice an important difference between SF(i) and RF(i): the formula for SF(i) depends on a percentage of storage while the RF(i) formula depends on the value of raw storage. The idea is that we don't want to put at a disadvantage nodes that, as per a deployment decision, have less storage space than others.

Analysis of RF(i) brings up a matter similar to SF(i) in that excessive energy levels can drive up the value of the function and artificially flag it as a good receiver node. In reality, this excess of energy may hide data space availability issues on the receiver node. it was found that in reality, aside from using a logarithm to attenuate the energy effect, a product rather than a sum gives a better assessment for a node's ability to receive data. Therefore, RF(i) as presented in equation 2 can be refined as:

$$RF(i) = j*\log(E/e)\ (k*S*LD(i) - l*(Da + p*Di)) \tag{2'}$$

The evaluation of data fitness for transfer was defined in equation 3 as

$$DF(i) = I*(trend(K) - k*|M - s|)$$

Statistical evaluation on this formula with a linear relation between the trend of the compensation factor (trend(K)) and the deviation of the data size from an ideal size (|M-m|) is a good approximation. Since there are already strict conditions on assembling a set of data units for relocation, it would be counterproductive to discount the selected data set in case there are variations from the ideal size. Hence, we opt for a linear dependence on size deviations.

Additional simulations were conducted on the formula assembling the three heuristic items:

(a*SF(n) + b*RF(m))*DF({du})

We need both a pair of nodes to participate in the data transfer and a data set. If either of these items scores low with our evaluation, we want to prevent the data relocation. Hence, a product is appropriate.

For the purposes of the simulations in this section, we have compared trends in values. Specific parameter values can be assigned for very specific deployment scenarios, which must be validated on a case by case basis via simulation.

### 5.2.2 *Relaxing data dependency for transfer*

In the approach described above, we have restricted the data movement to data units whose importance function can be evaluated even after the move. This restricts the set of data units that can be targeted for relocation. While not going into the details of a full solution, we present in this section two approaches that can be used to mitigate the relocation of data units

A first approach is to simply replace the importance function for the relocated units with a version that does not depend on other data units. This basically changes the business case approach and may not be feasible while retaining a reasonable amount of intended resolution.

A more refined approach is to use expected values to replace the parameters inside of the importance function. It is less than desirable and requires some statistical approach in deciding what the expected values are. In the end, this approach may not yield a good compromise either because we may be interested in situations where unexpected values are encountered.

## 5.3 *Redundancy*

Up to this point, we have not addressed the issue of data redundancy. Having multiple copies of data in a wireless sensor network is important to balance the high incidence of node failure. In the framework presented in this thesis, having multiple copies of the same data runs counter to idea that we seek to make maximum use of a limited storage space.

### *5.3.1 Implementing redundancy*

The implementation of redundancy is done at the specification of the data instanc. The additional parameters that we include relate to the number of that we want to store for that specific data instance. Hence, the data instance definition becomes:

Definition of an RDI with redundancy

---

*RDI: ($T_{start}$, $T_{end}$, param, recurrence, resolution, compression, copies), where:*
*$T_{start}$: start time*
*$T_{end}$: stop time*
*param: the parameter being collected*
*recurrence: how often the collection is done*
*resolution: how precise the stored value is*
*compression: boolean stating if the RDI has been compressed*
*copies: integer stating on how many nodes the data needs to be stored*
*e.g., RDI(2009/12/06 16:43:23, ongoing, temperature, 60 seconds, 0.01C, FALSE, 4)*

---

Figure 13. RDI definition with redundancy.

Defining an NRDI with redundancy

---

*NRDI: (T, param, resolution, compression, copies), where:*
*T: recording time*
*param: the parameter being collected*
*resolution: how precise the stored value is*
*compression: boolean stating if the NRDI has been compressed*
*copies: integer stating on how many nodes the data needs to be stored*
*e.g., NRDI(2009/12/06 16:43:23, temperature, 0.01C, FALSE, 3)*

---

Figure 14. NRDI with redundancy.

As data is collected one data unit at a time, these data units are stored locally (master copy) and copies are also stored on as many nodes as specified. These redundant nodes are not tied to one specific recurring data instance for example. The node collecting the data can select any other node to store the copies on. The process can be batched as node to node negotiations can be very costly for a small data segment at a time.

### *5.3.2 Importance calculation*

We have seen how each data instance has a function that can be used to compute the relative importance of each data unit produced. When we deal with several copies of the same data unit, we need to specify a different importance value for each of the copies. Redundant copies have less importance than master copies. For this reason, in addition to the factors listed in section 4.8.4, we propose an importance decay function which is applied to the importance calculation of redundant copies. For example, if we have a linear decay, the adjusted value of a data unit importance would be:

$$D(I(du)) = (1/r) * I(du), \quad (4)$$

where $r$ is the redundancy level

As nodes collect data and go through data reductions, some of the redundant copies of specific data units may be dropped. This is normal as nodes prioritize data units as a function of importance. However, depending on the environments in which the redundant copies end up, higher level redundancy copies may disappear before lower level ones. In that case, we need to attach a message trigger to update the redundancy level of the copied of the dropped data unit. A limited flood mechanism can be used as it is not expected for data units to migrate very long distance from the point of origin.

Since redundant copies are targeted for transfer, the function used in calculating the redundant copies importance level must be computable on the receiving node. That restricts the computation function in a way similar to the restrictions imposed by load sharing. In consequence, if the data importance calculation function depends on other stored data, then redundant copies need to carry a completely new function, as opposed to a function simply affected by a decay function.

#### *5.3.3 Load sharing under redundancy*

The objective of redundancy is to protect the data from eventual sensor node failures. We therefore have several copies stored in the network. As a result of using load sharing to mitigate storage issues, we may at some point attempt to store more than one copy of a data unit on the same node. This should be prevented as it defeats the purpose of redundancy.

## VI. CONCLUSION AND FUTURE WORK

In general, there are no specific solutions other than stating that one needs a systematic approach to data aging. Current strategies do not address the problem from the perspective of sink node unavailability, but rather from the perspective of an intelligent WSN, which gives the possibility of executing queries through the entire network.

In this article, primitives were proposed in order to deal with increasing amounts of data stored by a sensor node. The scope is to have the basic mechanisms to gracefully discard lower importance data, or lose some flexibility and data resolution, for the benefit of higher importance data. The results presented in the example are encouraging. With a more complex business case, the algorithms can get more complex, but the overall objective remains the same.

The internal operation of a sensor node consumes little energy compared to the requirements for transmission. Hence, the energy issue has been disregarded with respect to operations taking place without transmission.

Data importance becomes a factor of the entire network. While the entire sensor network works together towards the data gathering task, there is competition amongst individual nodes for access to the storage available in the sensor network.

When going from the single node focus to the multiple node scenario, the positioning of the nodes is a factor. The distance matters for transmission power. If the nodes are placed at predetermined positions, the preferred communication links and energy requirements can be pre computed. If the placement of nodes is non-deterministic, self-organization algorithms are needed to guide communication paths for access to other nodes' storage.

We presented a methodology for prioritizing data processing in WSNs. In sensor networks with intermittent connections to a sink node, managing the limited storage size becomes a relevant task.

In our case, the constant availability of the sink node is not a given, which reflects the reality in which sensor networks often operate. We have proposed that the data be divided into manageable data units. These data units are evaluated and ranked in terms of importance. When it is time to reduce the data load, the lowest importance data units are subjected to a specified data reduction function.

An example was shown with the use case of a very large area $CO_2$ monitoring WSN serviced by a travelling data sink node. Several data were collected: $CO_2$ concentration, car traffic, wind speed and direction. A complete solution was proposed using only external constraints on the importance of the data. One of the data collections was further considered with respect to internal data constraints.

The expansion of the model to several node systems was studied. While some nodes may experience spikes in data production, other nodes may simply just collect low relevance monitoring data. We considered this case for collaborative storage and proposed solutions for data handling. The context in which importance evaluations and reduction functions operate so far is specific to the node that has produced the data. When we consider sharing other nodes' storage space, data that is being housed on a different node needs to be sent with proper instructions for handling. Internal dependencies can no longer be enforced; they need to be rephrased or simply dropped.

As a final step to having a robust solution, redundancy was addressed addressed. We have assumed smooth operation of the nodes with no equipment loss or malfunctions. This can very well be the case in a remote area. The degree to which redundancy is required and how importance of redundant copies is computed need further exploration.

The decisions taken by the node while handling data make no assumption regarding scheduled, probabilistic, or statistical availability of the sink node. This additional information can affect the manner in which data reduction is applied, as well as the storage occupancy level at which data reduction processes are triggered.

Overall, the proposed solution and framework for a single node case are now robust enough to provide flexibility for future extensions. Data reduction in unattended sensor networks with intermittent or non reliable connections is an important computation when considering data storage and data aging.

We proposed an optimization function using prediction to map the use of data reduction primitives, optimization parameters (K, I) and dependency constraints (contextual or bounding). The model considers a probability that a variation of a variable is correlated with a variation of another variable. A variant of average values can also be considered. A simple use case had shown the nature of dependencies and computation challenges for two correlated readings.

We have presented a proposal for prioritized data reduction in the light of a multi-node deployment. In such a case, it becomes feasible that a node facing storage space shortages look first at offloading some data to neighboring nodes. We have proposed a mechanism by which a node can quickly assess the situation and decide between the option of local data reduction or a date transfer.

The opportunity for redundancy was also evaluated. As wireless sensor networks are often deployed in unreliable and dangerous areas, node failure is high. Data redundancy reduces the possibility of critical data loss caused by node failures. We presented a mechanism to evaluate the importance level of data units that are redundant copies rather than master copies of the collected data.

Accurate evaluation of the model requires extensive simulations, where combinations of the primitives and data parameters are combined with various type of constraints. We estimate that finding some correlation patterns will favor the use of average values, leading to a reasonable computation effort.

## REFERENCES

[1] Edgar H. Callaway, Jr. "Wireless Sensor Networks: Architectures and Protocols" , CRC Press, LLC, 2004

[2] Feng Zhao, Leonidas Guibas "Wireless Sensor Networks: an Information Processing Approach" 1st edition, Morgan Kaufmann, 2004

[3] Joon Ahn, Bhaskar Krishnamachari "Fundamental scaling laws for energy-efficient storage and querying in wireless sensor networks" The Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing, May 22-25, 2006, Florence, Italy, MobiHoc 2006:334-343

[4] Kyungseo Park, Ramez Elmasri "Query Classification and Storage Evaluation in Wireless Sensor Networks" 22nd International Conference on Data Engineering, April 3, 2006, Atlanta, GA, USA, ICDE Workshops 2006:35

[5] Norbert Siegmund, Marko Rosenmüller, Guido Moritz, Gunter Saake, and Dirk Timmermann. "Towards Robust Data Storage in Wireless Sensor Networks" Proceedings of Workshop on Database Architectures for the Internet of Things (DAIT). Birmingham, England, July 2009

[6] Joao Girão, Dirk Westhoff, Einar Mykletun, Toshinori Araki: "TinyPEDS: Tiny persistent encrypted data storage in asynchronous wireless sensor networks" Ad Hoc Networks (ADHOC) 5(7):1073-1089 (2007)

[7] E. Mykletun, J. Girao and D. Westhoff "Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks," IEEE International Conference on Communications, 2006, Istanbul, Turkey

[8] Majid I. Khan, Wilfried N. Gansterer, Günter Haring "In-Network Storage Model for Data Persistence under Congestion in Wireless Sensor Network". First International Conference on Complex, Intelligent and Software Intensive Systems, April 2007, Vienna, Austria, CISIS 2007:221-228

[9] Yongxuan Lai, Hong Chen, Yufeng Wang "Dynamic balanced storage in wireless sensor networks" 4th International Workshop on Data Management for Sensor Networks, September 2007, Vienna, Austria, DMSN 2007:7-12

[10] Davide Brunelli, Luca Benini, Clemens Moser, Lothar Thiele "An Efficient Solar Energy Harvester for Wireless Sensor Nodes" Design, Automation, and Test in Europe, March 2008, Munich, Germany, DATE 2008:104-109

[11] P. Ratanaworabhan, Jian Ke, M. Burtscher "Fast lossless compression of scientific floating-point data" Data Compression Conference, Salt Lake City, USA, DCC 2006. Proceedings 28-30 March 2006 Page(s):133 - 142 Digital Object Identifier 10.1109/DCC.2006.

[12] C. Dini and P. Lorenz, "Primitive Operations for Prioritized Data Reduction in Wireless Sensor Network Nodes", Proceedings of the 2009 Fourth International Conference on Systems and Networks Communications, September 2009, Porto, Portugal, ICSNC 2009, pp. 274-280

[13] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring", International Workshop on Wireless Sensor Networks and Applications, September 28, 2002, Atlanta, Georgia, WSNA'02, pp. 88-97

[14] E. H. Callaway, Jr. "Wireless Sensor Networks: Architectures and Protocols" , CRC Press, LLC, 2004

[15] M. I. Khan, W. N. Gansterer, and G. Haring, "In-Network Storage Model for Data Persistence under Congestion in Wireless Sensor Network:, First International Conferenece on Complex, Intelligent and Software Intensive Systems, April, 2007, Viena, Austria, CISIS'07, pp. 221-228

[16] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "Collaborative storage management in sensor networks", International Journal of Ad Hoc and Ubiquitous Computing, Volume 1, Issue 1/2 (November 2005), pp.47-58

[17] S. Tilak, W. Heinzelman, and N. Abu-Ghazaleh, "Storage Management Issues for Sensor Networks", Poser at International Conference on Network Protocols (ICNP '03), Atlant, Georgia, November 2003

[18] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets", ACM SIGCOMM Computer Communication Review, Volume 33 , Issue 1 (January 2003), p. 137 - 142

[19] N. Siegmund, M. Rosenmuller, G. Moritz, G. Saake, and D. Timmermann, "Towards Robust Data Storage in Wireless Sensor Networks", IETE Tech Rev [serial online] 2009 [cited 2010 Jan];26:335-40.
Available: http://tr.ietejournals.org/text.asp?2009/26/5/335/55280

[20] Y. Diao, D. Ganesan, G. Mathur, and P. Shenoy, "Rethinking Data Management for Storage-centric Sensor Networks", Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar CA, January 7 - 10, 2007.

[21] S. Tilak, N. Abu-Ghazaleh, and W. B. Heinzelman, "Storage management in sensor networks". Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions, IEEE/Wiley. pp. 257-281.

[22] C. Dini and P. Lorenz, "Primitive Operations for Prioritized Data Reduction in Wireless Sensor Network Nodes", Proceedings of the 2009 Fourth International Conference on Systems and Networks Communications, September 2009, Porto, Portugal, ICSNC 2009, pp. 274-280

[23] C. Dini and P. Lorenz, "Prioritizing Data Processing in Wireless Sensor Networks", Proceedings of the 2010 Sixth International Confenrce on Networks and Services, March 2010, Cancun, Mexico, ICNS 2010, pp. 23-31

[24] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate Data Collection in Sensor Networks usingProbabilistic Models", Proceedigns of the 22nd International Conference on Data Engineering. ICDE 2006, Atlanta, USA http://www.cs.umd.edu/~amol/papers/icde06.pdf [Retrieved: August 20, 2010]

[25] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring", International Workshop on Wireless Sensor Networks and Applications, September 28, 2002, Atlanta, Georgia, WSNA'02, pp. 88-97

[26] C. Guestrin, P. Bodik, T.R., P. Mark, ans S. Madden, "Distributed regression: an efficient framework for modleing sensor network data", IPSN, 2004

[27] S.Madden, M.J. Franklin, J.M. Hellerstein, and W. Hoing, "Tag: a tiny aggregation service for ad hoc sensor networks" , SIGOP Oper. Syst. Rev. 36(SI):131-146, 2009

[28] S. Nath, P.B. Gibbons, S. seshan, and Z.R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks", Proceedings of the 2nd ACM Conference on embedded Networked Sensor Systems, SenSys 2004, Baltimore, MD, USA, 2004

[29] J. Hellerstein and W. Wang, "Optimization of in-network data reduction", DMSN, 2002

[30] M. I. Khan, W. N. Gansterer, and G. Haring, "In-Network Storage Model for Data Persistence under Congestion in Wireless Sensor Network:, First International Conferenece on Complex, Intelligent and Software Intensive Systems, April, 2007, Viena, Austria, CISIS'07, pp. 221-228

[31] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "Collaborative storage management in sensor networks", International Journal of Ad Hoc and Ubiquitous Computing, Volume 1, Issue 1/2 (November 2005), pp. 47-58

[32] Y. Diao, D. Ganesan, G. Mathur, and P. Shenoy, "Rethinking Data Management for Storage-centric Sensor Networks", Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar CA, January 7 - 10, 2007.

[33] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-driven data acqusition in sensor networks", VLDB, 2004

[34] I. lazaridis and S. Mehtotra, "Capturign sensor-generated time series with quality guarantees", ICDE, 2003

[35] S.M. Graham Cormode, M. Garofalakis, and R. Rastogi, "Holistic aggregates in a network world: Distributed tracking of approximate quantiles, SIGMOD, 2005

[36] Y.-Ae¨ . Le Borgne, et al., "Adaptive model selection for time series prediction in wireless sensor networks", Signal Process. (2007), doi:10.1016/j.sigpro.2007.05.015

[37] S. Tilak, W. Heinzelman, and N. Abu-Ghazaleh, "Storage Management Issues for Sensor Networks", Poster at International Conference on Network Protocols, ICNP 2003, Atlanta, Georgia, 2003

[38] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets", ACM SIGCOMM Computer Communication Review, vol. 33, issue 1, January 2003, pp.137-142

[39] N. Siegmund, M. Rosenmuller, G. Moritz, G. Saake, and D. Timmermann, "Towards Robust Data Storage in Wireless Sensor Networks", IETE Tech Rev, 2009, vol. 26, issue 5, pp.335-40. Available: http://tr.ietejournals.org/text.asp?2009/26/5/335/55280 [accessed August 2010]

[40] M. Aly, N. Morsillo, and K. Pruhs "Zone sharing: a hot-spots decomposition scheme for data-centric storage in sensor networks" Second International Workshop on Data Management for Sensor Networks, DMSN 2005, Trondheim, Norway, pp.21-26

[41] Y. Lai, H. Chen and Y. Wang "Dynamic balanced storage in wireless sensor networks" 4th International Workshop on Data Management for Sensor Networks, DMSN 2007, Vienna, Austria, DMSN, pp.7-12

# A Coupled Random Search-Shape Grammar Algorithm for the Control of Reconfigurable Pixel Microstrip Antennas

Adrian Muscat
*Dept of Communications and Computer Engineering*
*University of Malta, Malta*
*Email: adrian.muscat@um.edu.mt*

Joseph A. Zammit
*Dept of Communications and Computer Engineering*
*University of Malta, Malta*
*Email: jozamm@gmail.com*

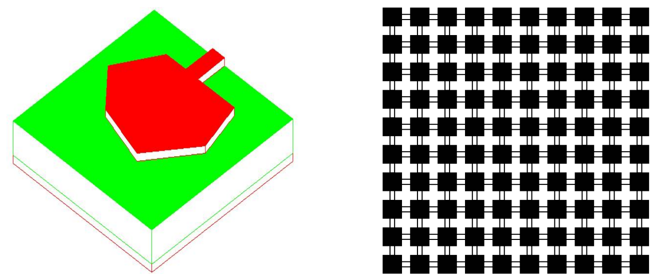
***Abstract*—Algorithms are necessary to reconfigure pixel antennas in real time. These algorithms must carry out an efficient search to yield the electrical specifications demanded by radio transceivers. To address this problem, an algorithm that combines a random search method with a shape grammar model is proposed. The performance of the algorithm is compared to a Genetic Algorithm with and without a pruning mechanism. Results demonstrate that the algorithm goes through a relatively very small number of iterations to converge to the antenna specifications and consequently outperforms the genetic-based searches.**

***Keywords*-Reconfigurable Pixel Antenna; Algorithm; Control; Shape Grammar**

## I. Introduction

This paper deals with algorithms applied in the control of pixel microstrip antennas. The shape grammar based algorithm described in [1] is extended to a wider range of microstrip patch shapes and its performance is compared to a Genetic Algorithm (GA) and a GA augmented with a pruning mechanism.

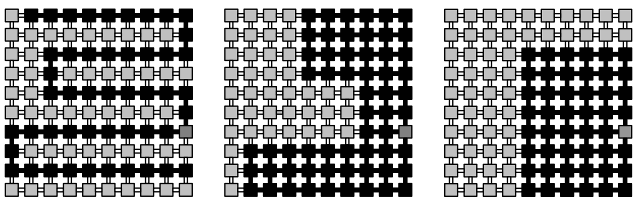
The reconfigurable pixel microstrip antenna (RPMA) is a derivative of the microstrip antenna which consists of a radiating element backed by a ground plane. The radiating element is usually positioned $0.3$ to $4mm$ above the ground plane and is physically supported either on a dielectric substrate or suspended under a superstrate. The radiating element can take the form of any useful shape and the final design of the antenna depends on the physical space that has been allocated to the antenna for the particular application as well as on the electrical characteristics specified. Fig.1(a) depicts a microstrip antenna with an arbitrarily shaped patch. To a certain extent, the antenna electrical properties depend on the shape of the radiating element. Subsequently, an antenna that is required to resonate at different frequencies requires the development of complex shapes that steer the current though the antenna in different paths for each resonant frequency. The design of such antennas is complex and requires many experimental iterations and modifications to produce the final prototype. An antenna that can alter its shape, and hence the electrical characteristics, in real-time under the control of an algorithm is therefore highly desirable. The delivery of such devices would mean that a single device re-configures itself to operate over a wide frequency range, optimizes the signal strength and/or changes polarization in a short period of time as demanded by the system [2].



(a) (b)

Figure 1. (a) Microstrip patch antenna structure with an arbitrary patch shape, and (b) Reconfigurable Pixel Microstrip Antenna (RPMA) that can change the radiator shape in real time.

Reconfigurable antennas can be classified either according to their physical composition or according to the fundamental parameters that can be modified and to what extent these parameters can be varied. Reconfigurable antennas are physically composed of switchable elements and tuning devices. Additionally, a control circuit and an algorithm is required to operate the switches. Simple re-configurable antennas contain very few switching devices to switch a parameter over a few discrete values and hence do not really need a complex smart control algorithm. On the other hand the pixel microstrip antenna is a much more complex antenna that can tune the parameters over a wide range and can for example switch to any arbitrary frequency or optimize the signal strength received or polarization diversity in real-time. This type of antenna can function as a "universal" antenna that can be applied to spectrum monitoring, in security situations to rapidly scan a wide bandwidth or in a cognitive radio. The major advantage over fixed wideband antennas is the ability to tune to particular narrow bands and/or change the direction of the main lobe. Fig.1(b) depicts the general pixel microstrip patch structure, consisting of an $n \times m$ array of pixels, interconnected with switches. The switches can then be turned $ON$ or $OFF$ to synthesize an arbitrary patch shape. Various shapes synthesized from a $10 \times 10$ pixel array are shown in Fig.2.



(a) (b) (c)

Figure 2. Antenna shapes synthesized with a RPMA (a) low, (b) medium, and (c) high frequency of operation.

The patch shapes synthesized by a RPMA are discrete and not continuous. This results in a structure whose frequency of resonance can be tuned in discrete steps and could lead to gaps along the frequency band. Various techniques have been proposed to mitigate these problems, for example the use of switchable shorting posts and the use of a tunable reactive component . However these add to complexity, have a negative effect on the antenna efficiency and limit the number of possible configurations [3]. Shorting posts are however useful in size reduction. On the other hand some degree of fine tuning can be obtained by taking advantage of the inherent reactive properties in the pixels and by suitably perturbing the shape a better match can be found. Furthermore it is usual and desirable to add a final matching stage to the antenna port [4]. A RPMA with a tunable matching circuit is described in [5]. It is also convenient and practical to have the probe feed at a fixed position. In this case the shape has to be rotated to suit the position of the feed. In summary, there are two aspects that the control algorithm has to address, (1) the minimization of switching, contributing to a longer lifetime, shorter convergence time and lower power consumption, and (2) making full use of reconfigurability to minimize the burden on the final tuning circuit.

A pixel reconfigurable antenna system therefore consists of the reconfigurable hardware parts and the control algorithm, on which the transient performance largely depends. Most of the research effort to-date has been targeted towards the hardware issues and the development of efficient control algorithms still needs to be adequately addressed.

In this section the reconfigurable pixel microstrip antenna is described and its inherent limitations and problems are discussed. Furthermore the tasks and targets for the control algorithm are defined. The rest of the paper is organized as follows. Section II describes related work and motivation. Section III reports a preliminary study that indicates the effectiveness of embedding domain knowledge in a search algorithm and discusses the importance of an algorithm that deals directly with shape. Section IV is a detailed description of the algorithm and discusses the results obtained. The last section outlines further work required to solve remaining issues.

## II. Related Work

The research field in reconfigurable microstrip antennas is very active with several research groups seeking the best possible structure to dynamically modify the characteristics efficiently. The most salient design issues from the point of view of hardware include, distortion due to non-linear properties of devices, overall loss in structure and the overall lifetime of the devices. Due to their linear properties micro-electro-mechanical-switches (MEMS) are very suitable switching devices for this application and prototypes have been proposed and built [5], [6] using MEMS. Cetiner et al [5] designed an $8 \times 8$ pixel patch antenna whose resonant frequency could be tuned to two bands 4.1GHz and 6.1GHz and for each band the polarization could be changed to linear, right or left hand polarization. These operating modes were achieved by switching on or off different pixels in the antenna. The main application of this pixel patch antenna is in MIMO systems to enhance the overall operation. Grau et al [7] developed a software defined pixel antenna which could be tuned over a range of range 1GHz. The pixels were connected together using MEMS switches. The pixels are switched in such a way that a circular structure is created and the radius of the circular structure yields the operating frequency and polarization. A competing structure consisting of ceramic pistons that push in and out metallic pixels is reported in [8]. The latter structure avoids the use of switches.

In the RECAP project ([9], [10] and [11]) a fully reconfigurable monopole planar antenna that operates from 0.8GHz to 1.5GHz is developed. Pringle at al describe a pixel patch antenna, where each pixel in the antenna is electrically small (less than $\lambda$/50) and interconnected by an optically switched FET. The antenna is $22.5cm$ by $22.0cm$ and has a bandwidth of 800MHz. The antenna was designed with left-right symmetry and in total consists of 208 switches. A genetic algorithm together with a Finite-Difference-Time-Domain (FDTD) model were used to derive the operating frequency and radiation pattern off-line. Due to mismatches in the feed the antenna gain was smaller than simulated.

A Self Structuring Antenna (SSA) [12] is another type of reconfigurable wire antenna. Instead of having an array of switches to connect patches together, the SSA is made up of a number of wires or patches connected with a series of interconnections to increase the antenna's size. Using a microprocessor, a feedback system and a Genetic algorithm the SSA can be configured over a large bandwidth.

Most of the work carried out on the control algorithm considers the Genetic Algorithm (GA) in the optimization of signal strength and operation at different frequencies. In [13] and [14], the GA is applied to tune the antenna for maximum signal strength on-situ and in [12] and [10] the GA is used to tune the antenna over a large range of frequencies. Most of the work related to the RPMA is influenced by the encoding technique proposed in [15]. In this case the patch shape is decomposed into square pixels and a binary bit string defines the presence or absence of a pixel. This binary bit string acts as a chromosome in a GA formulation. Genetic algorithms are useful to explore poorly-understood search spaces. They are coded with the minimum of domain knowledge and assumptions, and use domain independent genetic operators to explore the search space. They can be very robust and persistent throughout the search but are very slow to converge to a result and are inefficient for local optimization. Systems that rely solely on the GA, therefore, require hundreds of iterations to reach their goals, as shown in [13] and [15].

The RPMA can be thought of as an artificial antenna designer in a constrained environment. Therefore developments in antenna CAD can be considered for adoption to reconfigurable antennas. Patnaik et. al. apply Neural Networks (NN) to reduce the design time of a reconfigurable $130^0$ balanced bowtie antenna that can be described as a binary bit string of length 18 bits [16]. For the RPMA of fig.1(b), the equivalent chromosome length is 180 bits rendering this method computationally expensive. Several methods like pruning and structuring of the search space can be applied to improve the GA efficiency for CAD purposes. In [17], a Knowledge-Based Genetic Algorithm (KBGA) is developed to reduce the time required to evolve novel microstrip antennas. The KBGA selects antenna design heuristics (rules of thumb) that influence the genetic operators. The KBGA in [17] is hard-coded as a set of **if-then** rules and there is no provision for machine evolution of the rules.

In the case of the RPMA the various shapes that can yield the required characteristics are generally known and therefore a good proportion of the search space is known, fig.2 is evidence of this. The fixed antenna feed and the discrete nature of the shapes do however make the problem a formidable one, and require algorithms that can perturb the shape rather than the values of the geometrical dimensions. The changes in shape can be done efficiently by the application of domain knowledge, in the same way as human designers carry out the design task. In [18] shape-function grammars are proposed for use in antenna Intelligent Computer Aided Engineering (ICAE) software to model and synthesize microstrip antennas during the preliminary or conceptual stage of the design.In [1] a control algorithm that combines a search method with an approximate and fast to compute qualitative antenna model is proposed. Fig.3 is a block diagram of the proposed algorithm. The novelty in

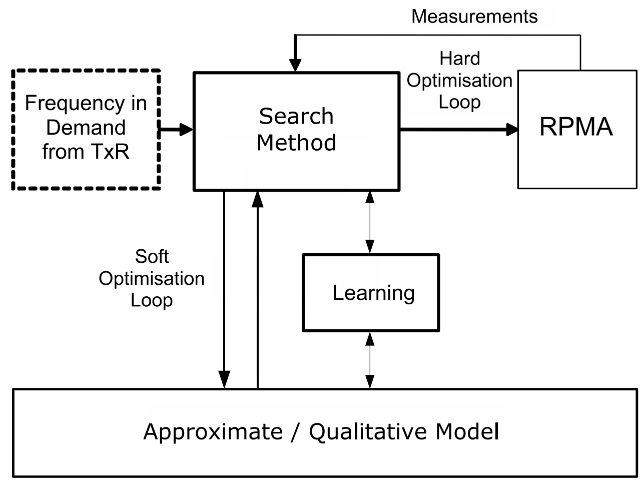



Figure 3. Block diagram for the algorithm modified from [1]. This algorithm introduces an approximate/qualitative model in the search loop.

this algorithm is the addition of the approximate/qualitative model to the standard setup that consists of a search algorithm that operates with measured feedback, such as described in [13] and [14]. Referring to fig.3, the search algorithm receives inputs from the transceiver and consults the *approximate/qualitative model* to obtain a valid shape that is likely to satisfy the inputs. Measured feedback is used to modify the shape or suggest a new shape. The *learning* option tunes the qualitative model if necessary. This approach mimics the informal and intuitive *cut and try* method used extensively by compact antenna designers. In this method the designer starts with a set of candidates derived with the help of an informal model. Guided by general antenna domain knowledge, the designer modifies the shapes in an iterative way to obtain the required electrical characteristics. At the end of the cycle the best prototype is chosen for production. The qualitative model should in general be inexpensive in computational terms and deals directly with shape and form. The choice of the search method depends on certain aspects. For example if the qualitative model is a crude one, a GA is appropriate, where as if the model yields a more accurate representation, then a neighborhood search algorithm should result in a more efficient system. In [1] a shape grammar model is combined with a random search and the performance of the algorithm is demonstrated on narrow-meander-line structures. In this paper the qualitative model is extended to generate and analyze wide-meander line patch antennas as well as shorted patch antennas.

## III. The Genetic Algorithm in Control of the RPMA

The Genetic Algorithm (GA) is currently the standard method for tuning a RPMA as discussed in section II. In this section the Genetic Algorithm (GA) is used in search of shape configurations that yield the required frequency. Two types of Genetic Algorithms are coded. The first is a traditional GA and the second is a GA with a pruning mechanism that prunes out candidates that are highly unlikely to yield the target values. The results in this section are compared to results obtained using the shape-grammar based algorithm in section IV

### A. The Genetic Algorithm

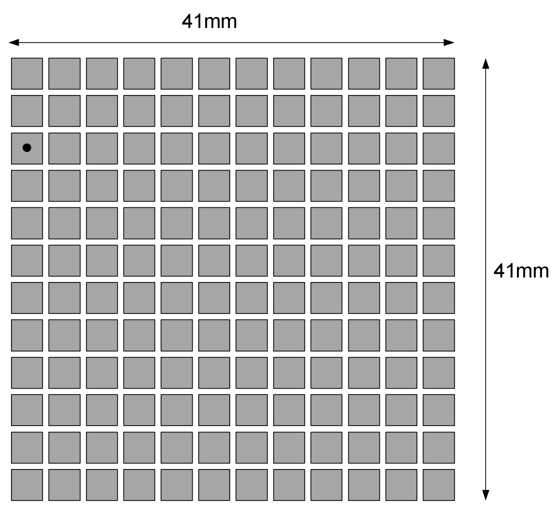



Figure 4. General structure for the Reconfigurable Pixel Microstrip Antenna for the GA experiments.

In this subsection the GA is used to investigate the search space of the reconfigurable pixel antenna. The goal is to investigate the number of generations or iterations required to find a shape that yields the given resonant frequency. A $12 \times 12$ RPMA is adopted for this experiment and each pixel is $3.41 \times 3.41mm$ in size. The pixels can be switched on and off to create the desired shape. The state of the pixel is coded as a '1' or a '0' in the GA chromosome. This encoding technique is first described in [15]. Fig.4 shows the structure of the reconfigurable antenna. In a practical reconfigurable antenna one would represent the switches in the chromosome, but this would end up in doubling the length of the chromosome and will not benefit this experiment. The total length and breadth of the antenna is $41 \times 41mm$. The radiating element is suspended in free space $3mm$ above the ground plane and has a nominal resonant frequency of $3.3GHz$. The GA was designed with a crossover probability of $0.7$ and an exponentially decreasing probability of mutation. At the start of the GA run, the mutation rate is $0.01$ and decreases to a constant $0.001$ by the $10^{th}$ generation. The GA was coded in C# and coupled with CST Microwave studio using Object Linking and Embedding (OLE). The fitness function is a function of the resonant frequency and return loss. As the structure resonates nearer to the target frequency the fitness score increases exponentially. For a return loss less than $-10dB$ an exponential fitness function was used, for a return loss higher than $-10dB$ the fitness score is a constant. Thus a structure with a resonant frequency near the target and a high degree of matching is considered to be very fit and has a higher probability of reproducing. Roulette wheel selection, that considers all individuals, is used to select the next population. The chromosome length is $143$ bits, given that the feed pixel is permanently 'ON'. A population of $40$ chromosomes is chosen as a good tradeoff between a suitable population size and computational time necessary to complete a population run. The GA was given $4$ target frequencies to reach as shown in Table I.

Table I
PERFORMANCE FIGURES FOR THE GENETIC ALGORITHM

| Target Frequency / GHz | Final Frequency / GHz | Population Runs | Number of Configurations |
|---|---|---|---|
| 3 | 3.01 | 10 | 400 |
| 2 | 2.08 | 70 | 2800 |
| 1 | 1.70 | 70 | 2800 |
| 0.7 | 1.70 | 70 | 2800 |

The GA is most successful for the $3.0GHz$ case. It manages to find a solution by considering $400$ candidates. This case is close to the nominal resonant frequency of $3.3GHz$. The results agree with results in [15] and [19], were the target frequencies are close to the nominal resonant frequency. The $2.0GHz$ case is also successful although the number of candidates considered is now $2800$. On the other hand the GA finds it very difficult to find suitable antenna configurations as the target frequency got smaller. For the $1.0GHz$ and the $0.7GHz$ target frequencies the GA fails to find a solution after considering $2800$ candidates. Fig.5 shows the best structures that the GA delivered for the successful frequencies. For smaller frequencies more pixels had to be switched off, in order to lengthen the path. The path length needed for resonance at a lower frequency is longer than that required for a higher frequency and the GA has to search for a longer time to get closer to a good shape. The GA performance therefore does not scale well with frequency.

### B. Pruning the GA Search Space

The second experiment introduces a pruning mechanism that improves the convergence of the search, with special consideration to low frequency targets. The idea is to eliminate those individuals that are highly unlikely to contribute towards reaching the goal. To perform such

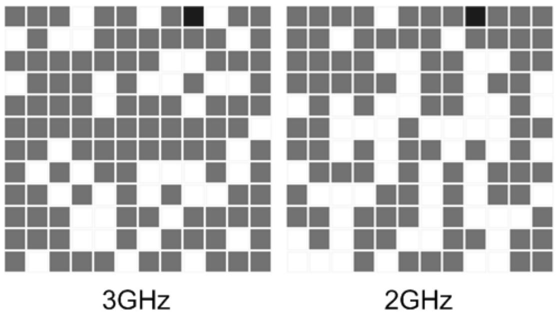



Figure 5. Best individuals found by the GA for the 3.0GHz and 2.0GHz cases.

pruning an approximate model is required. The chromosome is converted into a graph and the path from the feed to each pixel is calculated. A breadth-first search algorithm is used to find the path lengths from the feed position. The resonant frequency of the antenna is controlled by the length of the path between two potentially radiating edges and structures with a long pixel length are preferred for the next generation. Additionally during the process of selecting the next generation offsprings that are characterized by a path length that is similar or larger to the average length of the previous population are preferred. This was done to push the population towards structures with longer path lengths. The GA with the pruning mechanism was tested at a target frequency of $1.0GHz$. Table II shows a comparison of the results with and without pruning. The results show a marked improvement in computational time but the GA still finds it difficult to converge to the required result. This is an indication that the pruning model is not adequate.

Table II
PERFORMANCE FIGURES FOR THE GENETIC ALGORITHM WITH AND WITHOUT PRUNING

| Pruning | Target Frequency / GHz | Final Frequency / GHz | Population Runs | Number of Configurations |
|---|---|---|---|---|
| No | 1 | 1.7 | 70 | 2800 |
| Yes | 1 | 1.52 | 40 | 400 |

### C. Conclusions from GA experiments

The GA converges for frequencies that are $60\%$ or higher than the nominal frequency. There are various reasons why the GA fails to achieve the objective. An antenna will resonate at the lowest frequency along the longest path. If the crossover operation or the mutation operation does not modify the longest path, then the offspring will not be an improvement on its parents. The probability of the latter occurring is quite low if the reconfigurable antenna is encoded into a long chromosome. In summary the GA has difficulty in evolving longer winding paths that can support the lower resonant modes. The number of iterations to converge to the target can be effectively reduced by pruning the search space. The choice of pruning algorithm influences the subset of shapes searched by the antenna. In this case where the pruning algorithm looked for the longest paths the GA was then constrained to search in that subspace. The failure in finding low-frequency compact structures can therefore be attributed to an inefficient or incompatible encoding technique. What is required is an encoding technique that deals directly with shape.

## IV. THE COUPLED RANDOM SEARCH - SHAPE GRAMMAR ALGORITHM

The discussion in this section is grounded on the goal to develop an algorithm that can efficiently tune the RPMA over the range of mobile frequencies that span from a few hundred MHz to a few GHz. The problem is formulated as a search for a patch shape that yields the required frequency of operation, while minimizing the amount of hardware switching taking place. Referring to fig.3, the search method and the approximate/qualitative model are implemented with a random search method and a shape grammar with feedback model. Fig.6 shows a system diagram for the algorithm. The search algorithm receives inputs from the transceiver and instructs the *shape grammar model* to suggest a valid shape that is likely to satisfy the inputs. At this point in time the search method accepts or rejects the suggestion, depending on whether it falls or not within a specified range (soft loop). If accepted the candidate is hardware switched and measured feedback is used to accept the candidate or reject it (hard loop). This random search works on the premise that the accuracy of the shape grammar model is within the acceptance range and the designs are therefore close to the intended targets. As used here the shape grammar model reduces the global search problem to a local random search. Experiments show that within $10 - 30$ iterations a solution is found. The rests of this section describes shape grammars in general and the RPMA shape grammar model. Finally the algorithm is evaluated and demonstrated with three examples.

### A. Shape grammars

Shape grammars have been originally developed to generate architectural designs and art works that pertain to a particular style [20], [21]. Architectural designs are characterized by a well-defined function-shape decomposition such that the generating engine first satisfies the functional requirements and then proceeds to generate the physical shapes in a particular style. However compact microstrip antennas, like many engineering artifacts, do not have such a straightforward shape-function decomposition and instead

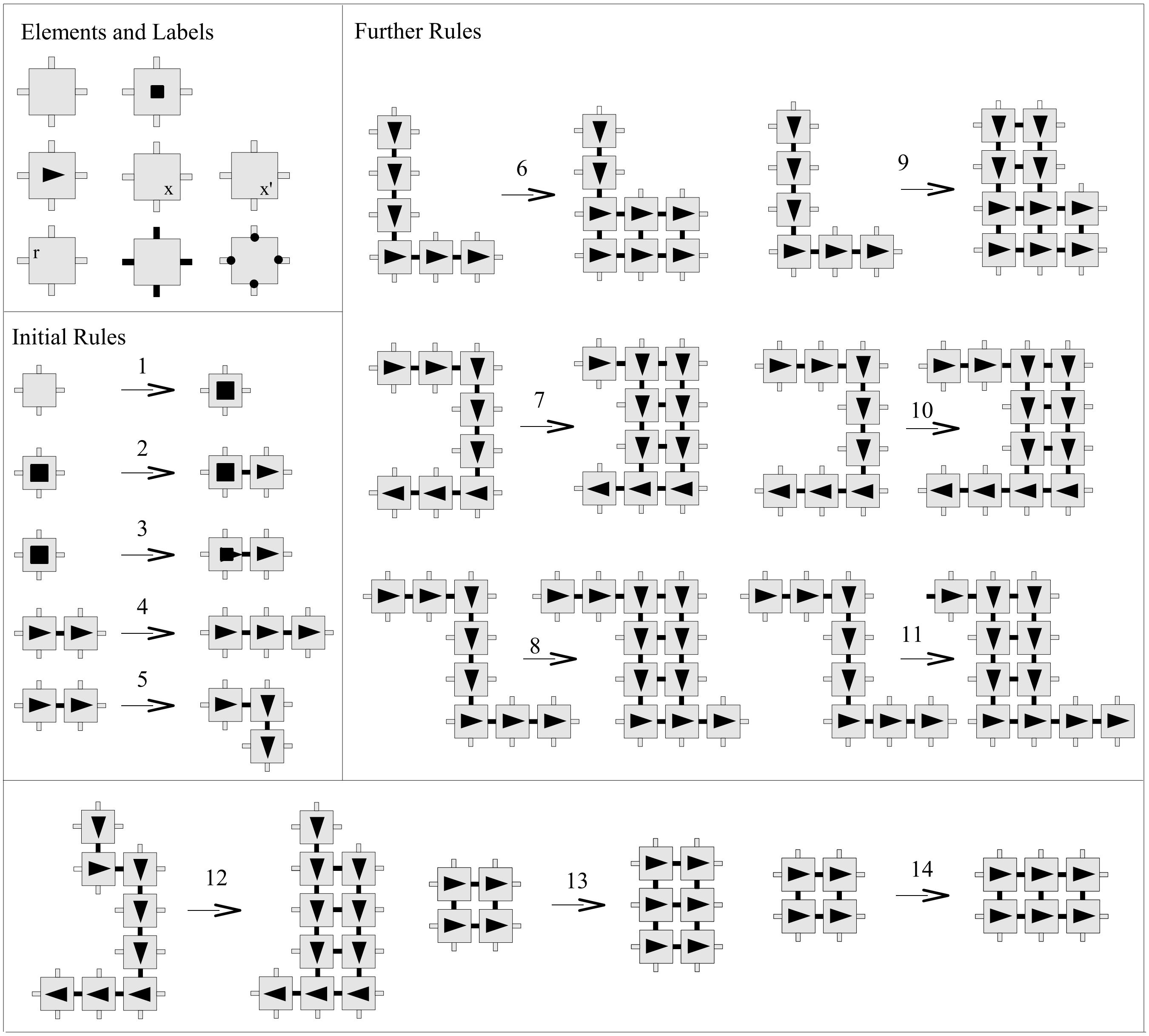



Figure 7. The reconfigurable pixel antenna shape grammar. This grammar produces shapes that are intended to operate at the fundamental mode of resonance.

exhibit a strong function-shape coupling, such that a small change in the topology or form can result not only in a significant change in performance, but can also render the design invalid. Agarwal and Cagan[22] propose shape grammars as a new framework for geometry-based engineering expert systems and express their belief that it is the class of products characterized by a strong function-shape coupling that stand to gain most from formal design tools because the lack of a pre-defined generation sequence limits human designs to only a small set of valid configurations.

In [23] a coupled shape-function grammar is proposed for the design of micro-electromechanical resonators. This is used to generate designs that function as resonators. The functional requirement is enforced by making sure that the grammar includes all the necessary elements that are required for the device to function and validity is ensured in this respect. The form of the device is then altered by choosing the rules at random or through a search process. This grammar does not include any feedback and this means that the design generated may not necessarily meet or be close to meeting all the required specifications for the intended applications.

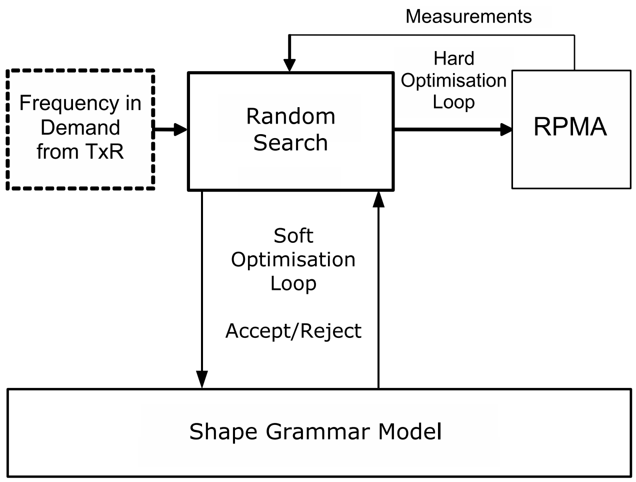



Figure 6. Block diagram for the control algorithm based on a random search method and a shape grammar model.

On the other hand, the microstrip antenna shape grammar described in [18] which is intended to be used as a generative engine that helps the antenna designer generate and combine shapes during the first phase of the design process, does include a feedback mechanism. This feedback consists of a transmission-line-model formula coupled to the shape attributes to approximate the frequency of operation and the input impedance.

*B. The pixel antenna shape grammar*

In this paper the microstrip shape grammar described in [18] and extended in [24] is modified to suit the RPMA. Fig.7 shows the RPMA antenna shape grammar. The grammar is a discrete and labeled 2D grammar. A shape grammar consists of four components; the initial shape, a finite set of shapes, a finite set of symbols and a finite set of shape rules. The basic shape is the square pixel with four switches, one on each side. Rectangles emerge and combine from the union of adjacent pixels. The grammar evolves a shape as two labeled branches or paths coming out of the probe feed or a shorting post. The paths formed are thus composed of cascaded rectangles. The grammar ensures that no branch overlaps or touches another branch and there are no loops along one path. This setup is suitable to model structures resonating at the fundamental mode, including shorted patch structures.

Fig.8 shows some examples generated by the grammar. The initial shape of fig.8(a) is evolved by applying the initial rules. The application of the initial rules (*rules 1 to 5*) generate a one-pixel-wide antenna shape. Referring to fig.8(a) the first rule applied is *rule 1*, followed by *rule 3*. *Rule 1* defines the first pixel as the location for the antenna feed port. *Rule 3* defines the first branch. This branch is evolved further by successively applying *rules 4, 5, 4, 4, 5 and 4*. The second branch is defined by first applying *rule 2* to the feed pixel and further evolved with a double application of *rule 4*. The labels in fig.8 are not all shown for clarity. Fig.7 defines the different labels. The black square defines a feed pixel. The black arrow indicates the direction of a path starting from the feed pixel through to the end of the path, indicating which switch is next turned ’ON’. The ’x’ label is the path label. So for two paths there are two path labels, for example ’x’ and ’y’. The path label is primed along the path and not primed at the end of the path. Thus the arrow and the path label allow an algorithm to follow the path. Rectangles are formed by switching an array of switches. The symbol ’r’ is a unique integer number given to an emergent rectangle as defined by the shape rules. The black dot label defines the interface between rectangles. These labels are used by the algorithm that manages the generation of the shape and to calculate the effective end to end path length, which result is used for the approximate formula model. Fig.9 shows a fully labeled case generated by the application of the *initial rules*.

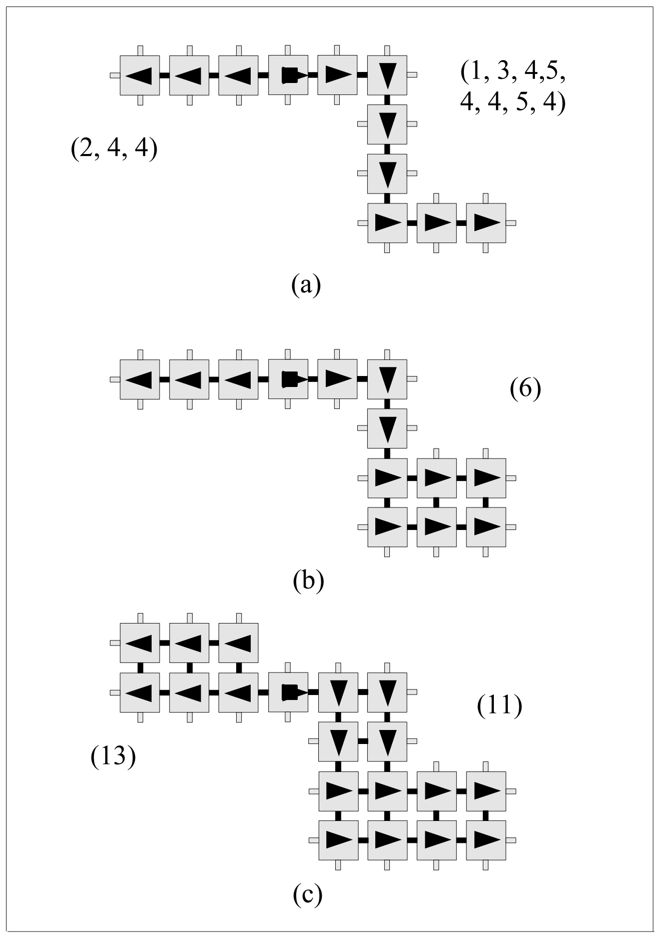



Figure 8. Examples of shapes generated by the grammar of fig.7. The rules applied are shown in brackets at the side of each branch.

The structure is further evolved by the application of the *further rules* given in fig.7. The structure of fig.8(a) is considered as an initial shape. The rules (6 to 12) can be either chosen at random or via a rule selection algorithm. In this experiment priority is given to evolving the path ends and followed by sections closer to the feed. Fig.8(b) depicts a structure derived using such a rule selection algorithm. In this case *rule 6* is applied. Applying *rule 11* to the branch on the right and then *rule 13* to the branch on the left results in the structure of fig.8(c).

Furthermore standard shapes, like the L-shape and C-Shape, can be defined by a subset of rules, including a limit on the number of certain rule instances that can be applied. In this way the control algorithm can test whether for example an L-shape fits for a given frequency, given that the feed position is defined. Additionally the line grammar is implicitly used to explore the design space and therefore takes into consideration other concurrent shapes when multi-feeds are present.

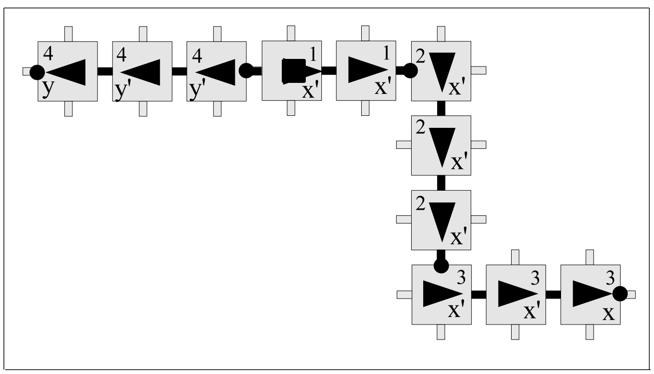



Figure 9. A labeled structure evolved with the initial rules.

As discussed in section III the qualitative model should deal directly with shape and develop a candidate in a rather formalized way. The RPMA shape grammar fulfills this requirement. The shapes generated consist of one feed and two non-touching branches. The feed is at a fixed position while the branches can take any shape. To generate a shape the grammar starts from the feed and evolves two branches that can take any path, subject to constraints. Additionally the shape grammar can perturb a shape in an apparent smart way, mimicing the action of a human being who applies knowledge to find a solution.

## C. Approximate Analysis

The shape grammar model relies on the capability of being able to carry out an approximate analysis of the structure. Labels are given to rectangles, paths, feed points and radiating edges. These labels and their positions are used to obtain the shape attributes; length and widths of rectangles, path length from feed to radiating edges or path ends and the presence and location of shorting posts and feeds. The shape attributes are then used to approximate the electrical characteristics (frequency of operation and input impedance) that are used as feedback by the rule selection algorithm. The quantitative relationship between the shape attributes and the electrical characteristics is described by a formula based on the transmission line model.

The resonant frequency depends on the length of the current path as well as on the degree of *meandering*. A function derived from the transmission line model (modified from [18]) is used to obtain an approximation for the resonant frequency in $GHz$, eqn.1.

$$f_0 = 300/2(L_e + (2a_0 + a_1 NC)L_p) \qquad (1)$$

where $L_e$ is the effective path length in $mm$, $NC$ refers to the integer number of corners along the path and $L_p$ is the width of the pixel in $mm$. $L_e$ is the end-to-end distance obtained by joining the midpoints of the rectangle edges along the path, as shown in fig.10. The coefficients $a_0$ and $a_1$ are fitted with a training set obtained from measurements, which in this study is substituted by a full-wave FDTD model [25]. For the structure of the RPMA shown in fig.1(a) these coefficients are manually optimized using a training set for the lower frequency bands, $a_0 = 0.55$ and $a_1 = -0.02$. The lower frequency band was chosen since, as previously shown, is the most challenging for a search algorithm.

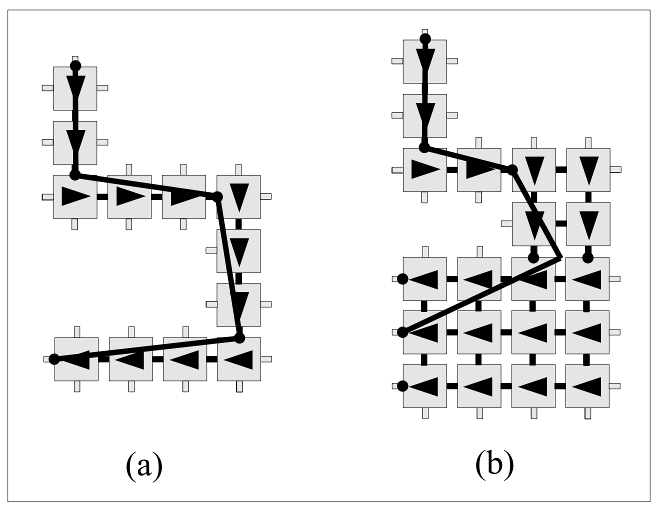



Figure 10. Evaluation of the *effective length*.

The driving point impedance depends mostly on the relative position of the feed point along the current path. The model does not yield a numerical value for the input impedance or input reflection coefficient but gives an indication of how far away it is from $50\Omega$, the system impedance. The input impedance estimate is obtained by considering the ratio of the effective length for branch 'x' and the effective length for branch 'y', $(L_x : L_y)$. From experiments it is concluded that a good match is obtained when the ratio $L_x : L_y$ is in the range of 0.7 to 0.85.

Furthermore the $L_x : L_y$ ratio value as well as $a_0$ and $a_1$ in eqn.1 can be tweaked by the learning method shown in fig.3 with measured feedback.

### *D. Evaluation Methodology and Results*

The effectiveness of the control algorithm in achieving the targets desired is demonstrated on three cases (a) pixel-wide meander-line shape resonating at $1.3GHz$, (b) wide meander-line shape at $1.8GHz$, and (c) shorted wide-meander-line shape at $0.9GHz$.

For the first experiment a $10 \times 10$ pixel antenna is assumed. The total size of antenna patch is $41mm \times 41mm$ and the size of each pixel is $3.6 \times 3.6mm$. The height of the substrate ($\epsilon_r$=1.0) is $3.0mm$. Eight candidate pixel-wide meander-line shapes that resonate at $1.3GHz$ are generated using *Algorithm Synthesize A*, fig.11, which together with the *shape grammar rules* is encoded into C++ code. The candidates generated are then simulated with an FDTD model, which is used as a benchmark and replaces measurements. The closest candidate to the required resonant frequency is chosen. Table III shows the resonant frequencies and errors for these eight candidates under the set of columns *Set A*. Candidate *#1* is the closest to $1.3GHz$ with a target error of $0.143\%$. The bandwidth of such an antenna is approximately $3\%$ and therefore this candidate is valid. This selection task is carried out by the *random search* method. If no shape satisfies the requirements then the search method requests more candidates and repeats the process. The same results can also be used to tune the model coefficients. Table III shows a second *set B*, where the best candidate is off the frequency mark by $0.263\%$. Fig.12 depicts these two candidates. This result demonstrates how effective a grammar based qualitative model can be in reducing the number of switching iterations required.

Table III
RESULTS SETS A AND B

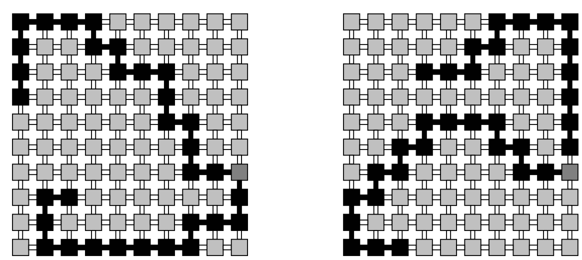
| # | Freq Model | Set A<br>Freq FDTD | Target Error | Freq Model | Set B<br>Freq FDTD | Target Error |
|---|---|---|---|---|---|---|
| 0 | 1.286 | 1.232 | 5.248 | 1.265 | 1.323 | 1.736 |
| 1 | 1.296 | 1.298 | 0.143 | 1.292 | 1.261 | 3.011 |
| 2 | 1.292 | 1.243 | 4.353 | 1.265 | 1.224 | 5.837 |
| 3 | 1.282 | 1.335 | 2.686 | 1.295 | 1.236 | 4.952 |
| 4 | 1.279 | 1.345 | 3.475 | 1.276 | 1.277 | 1.755 |
| 5 | 1.268 | 1.260 | 3.056 | 1.292 | 1.240 | 4.600 |
| 6 | 1.292 | 1.272 | 2.128 | 1.289 | 1.303 | 0.263 |
| 7 | 1.262 | 1.315 | 1.126 | 1.292 | 1.259 | 3.164 |

In the second experiment the quest is a patch shape resonating at $1.8GHz$. The antenna is a $12 \times 12$ pixel structure and the total size of the square antenna patch is $41mm \times 41mm$ with a pixel size of $2.9mm \times 2.9mm$. The patch is at a height of $3.0mm$ above the ground plane. The candidate shapes are generated using the full grammar and the algorithm is given in fig.13. For this run the coefficients are tweaked to $a_0 = 0.6$ and $a_1 = -0.1$. Table IV lists the first 30 candidates in the run. The model formula is less accurate in this case and therefore on average a greater number of candidates is necessary for the algorithm to converge. The best candidate is off the frequency mark by $0.556\%$ and for this case occurs at the $25^{th}$ iteration. These two candidates are shown in fig.14.

In the third experiment the quest is a patch shape resonating at $0.9GHz$. The antenna is a $10 \times 16$ pixel structure and the total size of the square antenna patch is $26mm \times 41mm$ with a pixel size of $2.05mm \times 2.05mm$. The patch is at a height of $3.0mm$ above the ground plane. The candidate

01 Set frequency of resonance and desired input impedance;
02 Start Synthesis
03 Define feed position, *rule 1*;
04 Define branch '$x$', *rule 2*;
05 Define branch '$y$', *rule 3*;
06 Obtain an estimate for the input impedance;
07 If the input impedance estimate is greater than the target, extend the shortest branch from the set $x$,$y$, *rules 4 or 5*; if branch is not extensible goto line 10;
08 If the input impedance estimate is less than the target, extend the longest branch from the set $x$,$y$, *rules 4 or 5*; if branch is not extensible goto line 10;
08 Obtain an estimate of the resonant frequency;
09 If frequency estimate is greater than the target value goto line 06;
10 End Synthesis
11 Return estimate of resonant frequency and input impedance;

Figure 11. Algorithm Synthesize A: Generates meander-line elements whose width is equal to one pixel.



(a) (b)

Figure 12. Best candidates in set A and B.

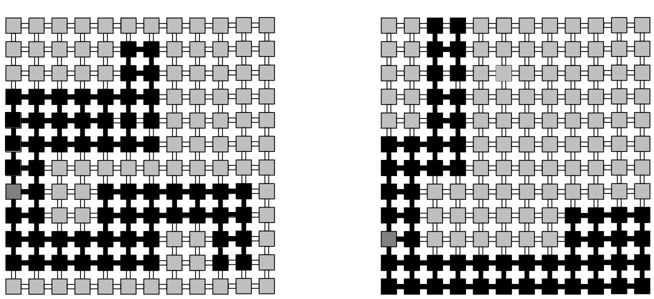
01 Set frequency of resonance and desired input impedance;
02 Start Synthesis;
03 Call Algorithm Synthesize A to generate an initial shape;
04 Define and reset subsetFlag to FALSE;
05 For Each branch from the set $x$,$y$ do
06 {
07 For Each rectangle along a branch (starting from the end) do
08 {
09 Build a subset of applicable rules from the set 6 ... 13;
10 If a subset is not NULL set subsetFLAG to TRUE;
11 Choose a rule from the subset and apply it with a probability of $P_a = 0.8$;
12 }
13 }
14 If subsetFLAG == FALSE goto line 16;
15 Goto Line 04 with a probability of $P_r = 0.7$
16 End Synthesis
17 Obtain an estimate for the input impedance;
18 Obtain an estimate of the resonant frequency;
19 Return estimate of resonant frequency and recomputed input impedance

Figure 13. Algorithm Synthesize B: Generates meander-line elements whose width is greater than one pixel.

shapes are generated with the full grammar, as in fig.13, and in this case a shorting post is added. Equation 1 is therefore adjusted to,

$$f_0 = 300/4(L_e + (2a_0 + a_1 NC)L_p) \tag{2}$$

Table IV lists the first 30 candidates in the run. The best candidate is off the frequency mark by $0.889\%$ and for this case occurs at the $19^{th}$ iteration. These two candidates are shown in fig.15.

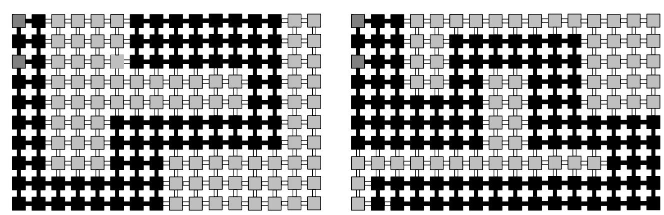


(a) (b)

Figure 14. The best two candidates resonating at 1.8GHz.

Table IV
CANDIDATES FOR THE 1.8GHZ SET AND THE 0.9GHZ SET

| | | 1.8GHZ Set | | | 0.9 GHz Set | |
|---|---|---|---|---|---|---|
| # | Freq Model | Freq FDTD | Target Error | Freq Model | Freq FDTD | Target Error |
| 0 | 1.774 | 1.755 | 2.481 | 0.912 | 0.928 | 3.089 |
| 1 | 1.817 | 1.858 | 3.228 | 0.889 | 0.873 | 2.997 |
| 2 | 1.866 | 1.871 | 3.962 | 0.914 | 0.950 | 5.559 |
| 3 | 1.856 | 1.914 | 6.327 | 0.936 | 0.957 | 6.300 |
| 4 | 1.774 | 1.751 | 2.716 | 0.989 | 1.023 | 13.619 |
| 5 | 1.764 | 1.692 | 6.025 | 0.855 | 0.848 | 5.767 |
| 6 | 1.787 | 1.828 | 1.542 | 0.937 | 0.881 | 2.109 |
| 7 | 1.908 | 2.088 | 15.982 | 0.937 | 0.999 | 11.016 |
| 8 | 1.742 | 1.637 | 9.030 | 0.918 | 0.982 | 9.074 |
| 9 | 1.831 | 1.890 | 5.006 | 0.851 | 0.878 | 2.446 |
| 10 | 1.805 | 1.827 | 1.527 | 0.833 | 0.936 | 3.953 |
| 11 | 1.789 | 1.730 | 3.911 | 0.936 | 0.850 | 5.527 |
| 12 | 1.705 | 1.688 | 6.244 | 0.894 | 0.918 | 1.963 |
| 13 | 1.815 | 1.775 | 1.388 | 0.823 | 0.861 | 4.389 |
| 14 | 1.892 | 1.785 | 0.833 | 0.967 | 0.925 | 2.747 |
| 15 | 1.723 | 1.605 | 10.807 | 0.959 | 1.033 | 14.752 |
| 16 | 1.750 | 1.638 | 8.976 | 0.827 | 0.748 | 16.943 |
| 17 | 1.875 | 1.723 | 4.271 | 0.884 | 0.879 | 2.385 |
| 18 | 1.806 | 1.864 | 3.558 | 0.884 | 0.863 | 4.157 |
| 19 | 1.856 | 1.757 | 2.381 | 0.831 | 0.892 | 0.889 |
| 20 | 1.914 | 2.019 | 12.152 | 0.894 | 0.878 | 2.456 |
| 21 | 1.786 | 1.819 | 1.040 | 0.892 | 0.877 | 2.508 |
| 22 | 1.735 | 1.523 | 15.372 | 0.912 | 0.952 | 5.830 |
| 23 | 1.951 | 1.763 | 2.029 | 0.862 | 0.824 | 8.445 |
| 24 | 1.781 | 1.560 | 13.345 | 0.864 | 0.853 | 5.273 |
| 25 | 1.975 | 1.810 | 0.556 | 0.877 | 0.923 | 2.512 |
| 26 | 1.930 | 1.936 | 7.562 | 0.982 | 0.951 | 5.680 |
| 27 | 1.806 | 1.896 | 5.340 | 0.835 | 0.810 | 10.014 |
| 28 | 1.822 | 1.778 | 1.196 | 0.928 | 0.884 | 1.778 |
| 29 | 1.772 | 1.994 | 10.766 | 0.876 | 0.766 | 14.853 |



(a) (b)

Figure 15. The best two candidates resonating at 0.9GHz.

## V. CONCLUSIONS AND FURTHER WORK

A novel algorithm that is based on a shape grammar model and a random search is proposed for the efficient control of the reconfigurable pixel microstrip antenna (RPMA). The shape grammar is coupled with formula transmission line models to generate shapes that are valid in terms of frequency of operation as well as input impedance. The shape grammar is intuitively derived from microstrip antenna design knowledge. The shape grammar is most useful in generating irregular meander-line antenna shapes commonly exploited in compact antennas. The performance of the algorithm is demonstrated on three prototypes and is compared to a Genetic Algorithm (GA) with and without a pruning mechanism. The proposed control algorithm outperforms the GA by a factor of 100 and by a factor of 10 when the pruning mechanism is switched on, in the terms of the number of iterations required to converge. Furthermore the algorithm scales well as the frequency of operation decreases, whereas the GA does not.

The experiments reported in this paper highlight the fact that the transmission line models adopted for the approximate analysis of a shape are most accurate for the pixel-wide meander-line examples. As the shape is modified via the further rules the accuracy degrades. This is the reason why more iterations are required for the wider lines. The width of the line has an effect on the input impedance and to a lesser extent on the frequency of resonance. Furthermore the width of the line is not a constant, but varies. Some improvement can be obtained by re-tuning the coefficients, but in general a formula model that takes into consideration the widths of various sections may perform better. Additionally it may be useful for the algorithm to be able to learn on the accuracy of the model and predict the number of iterations required. Further work is required to verify the performance of such models.

The demonstration in this paper is carried out using a random search, where the search method obtains a candidate from the shape grammar, and based on measured results accepts or rejects the candidate. This process continues till an acceptable solution is found and there is no attempt to modify the shape based on immediate measured feedback. It will be interesting to see whether the inclusion of such a shape annealing mechanism results in less iterations.

## REFERENCES

[1] A. Muscat and J. Zammit, "An efficient algorithm for the control of reconfigurable pixel microstrip antennas," in *Advances in Circuits, Electronics and Micro-electronics, 2009. CENICS '09. Second International Conference on*, pp. 44 –47, 11-16 2009.

[2] C. G. Christodoulou, D. A. Anagnostou, and L. M. Feldner, "Re-configurable antennas," in *IEEE International Workshop on Anti-counterfeiting, Security, Identification.*, pp. 0–12, April 2007.

[3] A. Muscat and J. A. Zammit, "Reconfigurable antenna structure for a wideband cognitive radio," in *IET Seminar on Cognitive Radio and Software Defined Radios: Technologies and Techniques*, pp. 1–5, IET, September 2008.

[4] S. Kingsley, D. Ireland, S. O'Keefe, R. Langley, and L. Liu, "In search of the perfect handset antenna," in *Antennas and Propagation Conference, 2008. LAPC 2008. Loughborough*, pp. 62 –65, 17-18 2008.

[5] B. A. Cetiner, H. Jafarkhani, J.-Y. Qian, H. J. Yoo, A.Grau, and F. D. Flaviis, "Multifunctional reconfigurable mems integrated antennas for adaptive mimo systems," *IEEE Communications Magazine*, vol. 42, pp. 62–70, December 2004.

[6] A. Grau, L. Ming-Jer, J. Romeu, H. Jafarkhani, L. Jofre, and F. D. Flaviis, "A multifunctional mems-reconfigurable pixel antenna for narrowband mimo communications," in *Antennas and Propagation International Symposium*, pp. 489–492, IEEE, June 2007.

[7] A. Grau, J. Romeu, L. Jofre, and F. De Flaviis, "A software defined mems-reconfigurable pixel-antenna for narrowband mimo systems," in *Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on*, pp. 141 –146, 22-25 2008.

[8] E. Walton, E. Lee, Y. Bayram, A. Duly, B. Salisbury, G. Bruce, and B. Montgomery, "Reconfigurable antenna arrays using pixel elements," *29th Annual Symp. of the Antenna Measurement Techniques Association. St. Louis, MO:AMTA*, November 2007.

[9] L. Pringle, P. Friederich, S. Blalock, G. Kiesel, P. Harms, D. Denison, E. Kuster, T. Fountain, and G. Smith, "Gtri reconfigurable aperture design," in *Antennas and Propagation Society International Symposium, 2002. IEEE*, vol. 1, pp. 473 – 476 vol.1, 2002.

[10] L. N. Pringle, P. H. Harms, S. P. Blalock, G. N. Kiesel, E. J. Kuster, P. G. Friederich, R. J. Prado, J. M. Morris, and G. S. Smith, "A reconfigurable aperture antenna based on switched links between electrically small metallic patches," *IEEE Transactions on Antennas and Propagation*, vol. 52, pp. 1434–1445, June 2004.

[11] L. Pringle, P. Harms, S. Blalock, G. Kiesel, E. Kuster, P. Friederich, R. Prado, J. Morris, and G. Smith, "The gtri prototype reconfigurable aperture antenna," in *Antennas and Propagation Society International Symposium, 2003. IEEE*, vol. 2, pp. 683 – 686 vol.2, 22-27 2003.

[12] C. Coleman, E. Rothwell, and J. Ross, "Self-structuring antennas," in *Antennas and Propagation Society International Symposium, 2000. IEEE*, vol. 3, pp. 1256 –1259 vol.3, 2000.

[13] D. S. Linden, "Optimizing signal strength in-situ using an evolvable antenna system," in *Proceedings of the 2002 NASA/DOD Conference on Evolvable Hardware*, 2002.

[14] Z. Min, L. Xiao-Wu, and W. Guang-Hui, "Preliminary research of the reconfigurable antenna based on genetic algorithms," in *3rd International Conference on Computational Electromagnetics and Its Applications*, pp. 137–140, November 2004.

[15] J. Johnson and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *Antennas and Propagation Magazine, IEEE*, vol. 39, pp. 7 –21, aug 1997.

[16] A. Patnaik, D. Anagnostou, C. G. Christodoulou, and J. C. Lyke, "A frequency reconfigurable antenna design using neural networks," *Antennas and Propagation Society International Symposium*, pp. 409–412, July 2005.

[17] A. Muscat, *The Design of Low Gain, Wideband and Multiband Antennas Employing Optimisation Techniques*, ch. Optimisation Based Design, pp. 120–174. Queen Mary University of London, January 2002.

[18] A. Muscat, "A shape-function grammar approach for the synthesis and modelling of pixel-microstrip-antennas," in *Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP '09. Third International Conference on*, pp. 23 –28, 11-16 2009.

[19] N. Herscovici, M. F. Osorio, and C. Peixeiro, "Miniaturization of rectangular microstrip patches using genetic algorithms," *IEE Antennas and Wireless Propagation letters*, vol. 1, pp. 94–97, 2002.

[20] G. Stiny and J. Gips, *Algorithmic Aesthetics: Computer Models for Critisism and Design in teh Arts*. Berkeley and Los Angeles, California: University of California Press, 1978.

[21] U. Flemming, "More than the sum of parts: the grammar of queen anne houses," *Environment and Planning B: Planning and Design*, vol. 14, pp. 323–350, 1987.

[22] M. Agarwal and J. Cagan, "On the use of shape grammars as expert systems for geometry-based engineering design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 14, pp. 431–439, 2000.

[23] M. Agarwal and J. Cagan, "A micro language: generating mems resonators by using a coupled form-function shape grammar," *Environment and Planning B: Planning and Design 2000*, vol. 27, pp. 615–626, 2000.

[24] A. Muscat, "A shape grammar with feedback generative model for the design of compact microstrip antennas," in *International Journal On Advances in Systems and Measurements, accepted for publication.*, 2010.

[25] A. Muscat, *The Design of Low Gain, Wideband and Multi-band Antennas Employing Optimisation Techniques*, ch. FDTD Model for The Patch Antenna, pp. 68–118. Queen Mary University of London, January 2002.

**www.iariajournals.org**

**International Journal On Advances in Intelligent Systems**
ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2679

**International Journal On Advances in Internet Technology**
ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
issn: 1942-2652

**International Journal On Advances in Life Sciences**
eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
issn: 1942-2660

**International Journal On Advances in Networks and Services**
ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
issn: 1942-2644

**International Journal On Advances in Security**
ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS
issn: 1942-2636

**International Journal On Advances in Software**
ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2628

**International Journal On Advances in Systems and Measurements**
ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
issn: 1942-261x

**International Journal On Advances in Telecommunications**
AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
issn: 1942-2601