



## 216715 NEWCOM<sup>++</sup>

### DR.11.2

#### Intermediate report on resource management issues and routing/forwarding schemes for opportunistic networks

**Contractual Date of Delivery to the CEC:** T0+18

**Actual Date of Delivery to the CEC:** T0+18

**Editor(s):** Sergio Palazzo (CNIT-CT)

**Participating institutions:** Bilkent/KHAS, CNIT-CT, CNIT-BO, CNIT-PD, CNIT-TO, CNRS-LAAS, ISMB, IST-TUL, KAU, NKUA/IASA, PUT, UPC, RWTH

**Contributors:** L. Galluccio, A. Leonardi, G. Morabito, C. Rametta (CNIT-CT); F. Fabbri, R. Verdone (CNIT-BO); A. Zanella, F. Zorzi (CNIT-PD); C.-F. Chiasserini, C. Casetti (CNIT-TO); T. Pérennou (CNRS-LAAS); R. Scopigno, M. Spirito, F. Fileppo, L. Pilosu, M. Franceschinis (ISMB); L. Pedrosa, J. Soares, R. Rocha, L. Ferreira, L.M. Correia (IST-TUL); A. Kassler, A. Brunstrom, M. Cavalcanti de Castro, J. Karlsson (KAU); A. L. Moustakas, P. Kazakopoulos (NKUA/IASA); R. Krenz (PUT); J. Riihijarvi (RWTH)

**Internal Reviewer(s):** S. Benedetto (ISMB), R. Ferrus (UPC), I. Korpeoglu (Bilkent/KHAS)

**Workpackage number:** WPR11: Opportunistic Networks

**Nature:** R

**Total Effort Spent:** 8 m/m

**Dissemination Level:** Public

**Version:** 1 (Final)

#### Abstract:

This deliverable presents an overview of the activities carried out within the NEWCOM<sup>++</sup> Workpackage WPR.11 during the first 18 months. We provide a description of the on-going currently consolidated Joint Research Activities (JRAs) and the main results so far obtained. We also address some considerations on the future activities which are expected to start during the second year of NEWCOM<sup>++</sup>.

**Keyword list:** Opportunistic networks, localization, analysis, transport, peer-to-peer, connectivity, testbed, capacity, resource description language.

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	List of Acronyms . . . . .	6
1.2	Glossary . . . . .	7
<b>2</b>	<b>Opportunistic Localization and Tracking</b>	<b>9</b>
2.1	Research activities . . . . .	9
2.2	Related works . . . . .	9
2.3	Linear matrix inequality opportunistic localization scheme . . . . .	10
2.3.1	Scenario and main assumptions . . . . .	10
2.3.2	Results . . . . .	12
2.3.3	Future work . . . . .	14
2.4	Analysis of self-localization error models and opportunistic enhancement . . . . .	15
2.4.1	Future work . . . . .	16
<b>3</b>	<b>Mathematical Modeling of Intermittent Behavior in Opportunistic Networks</b>	<b>17</b>
3.1	Research activities . . . . .	17
3.2	Open issues and goals . . . . .	22
<b>4</b>	<b>Transport Layer Issues in Opportunistic Networks</b>	<b>23</b>
4.1	Research activities . . . . .	23
4.2	Packet aggregation in wireless multi-hop networks . . . . .	23
4.2.1	Proposed packet aggregation algorithm . . . . .	24
4.2.2	Simulation results with aggregation . . . . .	25
4.3	Enhanced MAC layer . . . . .	27
4.3.1	Proposed enhancements to MAC layer . . . . .	28
4.3.2	Simulation results with the enhanced MAC layer . . . . .	30
4.4	Future work . . . . .	32
<b>5</b>	<b>Peer-to-peer Techniques in Opportunistic Mesh Networks</b>	<b>33</b>
5.1	Research activities . . . . .	33
5.2	Bamboo: handling churn over the Internet . . . . .	33
5.3	Georoy: a location-aware P2P algorithm . . . . .	34
5.3.1	The Viceroy algorithm . . . . .	35
5.3.2	ID and level assignment . . . . .	36
5.3.3	Overlay construction . . . . .	36
5.3.4	Routing . . . . .	36
5.3.5	Overlay maintenance . . . . .	36
5.3.6	Viceroy's properties. . . . .	37
5.3.7	Georoy . . . . .	38
5.3.8	Mobility management procedures . . . . .	39
5.4	Ongoing activity and future work . . . . .	39
<b>6</b>	<b>Opportunistic Connectivity: the Impact of Nodes Mobility</b>	<b>41</b>
6.1	Research activities . . . . .	41
6.2	Related works . . . . .	42
6.3	Scenario description and simulation setup . . . . .	42
6.3.1	Scenario description . . . . .	42
6.3.2	Simulation setup . . . . .	43
6.4	Performance metrics . . . . .	44
6.4.1	Inter-contact time . . . . .	44

6.4.2	Other performance metrics . . . . .	46
6.5	Numerical results . . . . .	47
6.6	Discussion . . . . .	48
6.7	Conclusions and future work . . . . .	49
<b>7</b>	<b>Heterogeneous and Opportunistic Wireless Mesh Networks</b>	<b>50</b>
7.1	Capacity analysis of opportunistic WMNs using collision domains and multi-radio WMNs	50
7.1.1	Collision domains in 802.16 WMN . . . . .	50
7.1.2	Multi-radio WMNs . . . . .	51
7.1.3	Future work . . . . .	55
7.2	Wireless Mesh Networks with Mobility Support . . . . .	55
7.2.1	Reference Scenario . . . . .	56
7.2.2	Comparing different routing protocols . . . . .	56
7.2.3	Future work . . . . .	59
<b>8</b>	<b>Resource Description Language</b>	<b>61</b>
8.1	Research activities . . . . .	61
8.2	Concept . . . . .	61
8.3	Scenarios . . . . .	62
8.3.1	Cross-layering . . . . .	62
8.3.2	Modular middleware solution . . . . .	62
8.3.3	Community . . . . .	63
8.4	RDL within WPR.11 JRAs . . . . .	63
8.5	Format specification . . . . .	64
8.6	Conclusions . . . . .	64
<b>9</b>	<b>Experimental Activities</b>	<b>66</b>
9.1	Experimental scenarios . . . . .	66
9.1.1	Mesh . . . . .	66
9.1.2	WSN . . . . .	68
9.2	Testbeds . . . . .	70
9.2.1	ISMB mesh testbed . . . . .	70
9.2.2	KAUMesh . . . . .	71
9.2.3	ISMB WSN testbed . . . . .	72
9.2.4	Tagus-SensorNet . . . . .	73
9.3	Emulation of opportunistic networks . . . . .	75
9.3.1	The KauNet emulation system . . . . .	75
9.3.2	Future work: emulation of opportunistic networks . . . . .	76
9.4	Contribution to WPR11 JRAs and other WPRs . . . . .	77
9.4.1	Mesh . . . . .	77
9.4.2	WSN . . . . .	79
<b>10</b>	<b>Conclusions</b>	<b>80</b>

## 1 INTRODUCTION

The NEWCOM++ Workpackage WPR.11 deals with the emerging paradigm of opportunistic communication, which enables nodes and user devices to self-configure and exploit resources of separate network systems according to the needs of specific application tasks.

At T0+6 we have released the first deliverable DR11.1, where we provided an overview of the current state of the art of research on opportunistic networks and introduced a common framework for reference models and performance metrics. In that deliverable, we also highlighted the main research issues in opportunistic networks and we discussed challenges and possible solutions.

More specifically, we have identified the following areas of research: mobility characterization and discovery algorithms; routing and forwarding techniques; novel scheduling, resource allocation and MAC schemes.

Finally, we described the Joint Research Activities (JRAs) that have been planned in the context of WPR11. For each JRA, we provided the participating institutions, the open issues, the objectives and the work organisation.

Each JRA has been developed through face-to-face meetings, conference calls, and exchange visits of researchers belonging to the participant institutions. The intermediate results obtained in each JRA have been presented by the partners during the WPR11 meetings held in Lisbon and Barcelona on November 2008 and April 2009, respectively.

This document summarizes the work carried out during the first 18 months in the context of WPR11 with particular emphasis on the joint research activities which can be considered consolidated at this moment.

This deliverable has been written in a cooperative manner. Partners involved in each specific JRA have provided the contents of the related section. Then, each section has been edited by a section editor, which usually corresponds to the catalyst partner of that JRA, and finally the entire document has been harmonized by the editor of the deliverable.

The rest of the document is organized as follows:

- Section 2 deals with the JRA "Opportunistic Localization and Tracking"
  - Contributors: CNIT-PD, CNRS-LAAS, ISMB, RWTH
- Section 3 deals with the JRA "Mathematical Modeling of Intermittent Behavior in Opportunistic Networks"
  - Contributors: NKUA/IASA, CNIT-CT
- Section 4 deals with the JRA "Transport Layer Issues in Opportunistic Networks"
  - Contributors: CNIT-CT, KAU, CNRS-LAAS
- Section 5 deals with the JRA "Peer-to-peer Techniques in Opportunistic Mesh Networks"
  - Contributors: CNIT-CT, KAU
- Section 6 deals with the JRA "Opportunistic Connectivity: the Impact of Nodes Mobility"
  - Contributors: CNIT-BO, RWTH
- Section 7 deals with the JRA "Heterogeneous and Opportunistic Wireless Mesh Networks"
  - Contributors: PUT, IST-TUL, KAU, CNIT-TO, ISMB
- Section 8 deals with the JRA "Resource Description Language"
  - Contributors: IST-TUL, CNIT-BO

- Section 9 deals with the JRA "Experimental Activities"
  - Contributors: ISMB, KAU, IST-TUL

Each Section reports a short summary about the evolution of the JRA and describes the work carried out by each partner inside the JRA, together with the main results obtained to date and the envisaged future work.

Finally, in Section 10 some conclusions are drawn.

## 1.1 List of Acronyms

<b>3G</b>	3rd Generation
<b>ACK</b>	Acknowledgement
<b>ADSL</b>	Asymmetric Digital Subscriber Line
<b>AoA</b>	Angle of Arrival
<b>AODV</b>	Ad hoc On Demand Distance Vector
<b>AP</b>	Access Point
<b>API</b>	Application Programmers Interface
<b>AVP</b>	Attribute-Value-Pair
<b>BATMAN</b>	Better Approach To Mobile Ad-hoc Networking
<b>CBR</b>	Constant Bit Rate
<b>CCDF</b>	Complementary Cumulative Distribution Function
<b>CSMA/CA</b>	Carrier Sense Multiple Access with Collision Avoidance
<b>CTS</b>	Clear To Send
<b>CW</b>	Contention Window
<b>DCF</b>	Distributed Coordination Function
<b>DHT</b>	Distributed Hash Table
<b>DIFS</b>	Distributed Inter Frame Space
<b>DTN</b>	Delay Tolerant Network
<b>FMIPv6</b>	Fast Mobile IPv6
<b>FTP</b>	File Transfer Protocol
<b>FTSP</b>	Flooding Time Synchronization Protocol
<b>GPS</b>	Global Positioning System
<b>GPSR</b>	Greedy Perimeter Stateless Routing
<b>IEEE</b>	Institute of Electrical and Electronic Engineers
<b>INS</b>	Inertial Navigation System
<b>IP</b>	Internet Protocol
<b>JRA</b>	Joint Research Activity
<b>KLV</b>	Key-Length-Value
<b>LMI</b>	Linear Matrix Inequality
<b>LPL</b>	Low Power Listening
<b>LQI</b>	Link Quality Indicator
<b>MAC</b>	Medium Access Control
<b>MCMI</b>	Multi-Channel Multi-Interface
<b>MEMS</b>	Micro Electro-Mechanical Systems
<b>MIPv6</b>	Mobile IPv6
<b>MOS</b>	Mean Opinion Score
<b>MPLS</b>	Multi Protocol Label Switching
<b>MTU</b>	Maximum Transmission Unit
<b>MULE</b>	Mobile Ubiquitous LAN Extension
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>OLSR</b>	Optimized Link State Routing
<b>OS</b>	Operating System
<b>P2P</b>	Peer-to-Peer
<b>PC</b>	Personal Computer
<b>QoS</b>	Quality of Service
<b>RDF</b>	Resource Description Framework
<b>RDL</b>	Resource Description Language
<b>RF</b>	Radio Frequency
<b>RMS</b>	Root Mean Square

<b>RRM</b>	Radio Resource Management
<b>RSSI</b>	Received Signal Strength Indication
<b>RTS</b>	Request To Send
<b>RTT</b>	Round Trip Time
<b>SIFS</b>	Short Inter Frame Space
<b>SOC</b>	Self Organized Criticality
<b>TCP</b>	Transmission Control Protocol
<b>ToA</b>	Time of Arrival
<b>UDP</b>	User Datagram Protocol
<b>USB</b>	Universal Serial Bus
<b>VLAN</b>	Virtual Local Area Network
<b>VoIP</b>	Voice over Internet Protocol
<b>W3C</b>	World Wide Web Consortium
<b>WCETT</b>	Weighted Cumulative Expected Transmission Time
<b>WLAN</b>	Wireless Local Area Network
<b>WPR</b>	Research Workpackage
<b>WiFi</b>	Wireless Fidelity
<b>WMN</b>	Wireless Mesh Network
<b>WSN</b>	Wireless Sensor Network
<b>WWW</b>	World Wide Web
<b>XML</b>	Extensible Mark-up Language

## 1.2 Glossary

- **Brownian movement** is the seemingly random movement of particles suspended in a fluid (i.e. a liquid or gas). It takes the name from Scottish botanist Robert Brown.
- **Contact opportunity**: due to the node mobility or the dynamics of the wireless channel, a node might make contact with other nodes at an unpredicted time. Since contacts between nodes are hardly predictable, they must be exploited opportunistically for exchanging messages between some nodes that can move between remote fragments of the network.
- **Delay Tolerant Network (DTN)** is an overlay network which supports the interoperability of heterogeneous networks which may be characterized by one or more of the following characteristics: intermittent connectivity, long or variable delay, asymmetric data rate, high error rate.
- **Mesh** networks can be seen as one type of ad hoc network where nodes can connect to each other via multiple hops, and they generally are not mobile.
- **MULEs**: mobile entities which pick up data from sensors when in close range, store it, and drop it to wired access points.
- **Oppnets** constitute the category of ad hoc networks where diverse systems, not originally employed as nodes of an oppnet, join it dynamically in order to perform certain tasks they have been called to participate in.
- **Opportunistic communication** enables nodes and user devices to self-configure and exploit resources in extremely dynamic networks. This paradigm encompasses features and methods that are especially suitable in both disconnected environments, in which islands of connected devices suddenly appear, disappear and reconfigure dynamically, and pervasive networking scenarios, where epidemic data exchanges occur among mobile devices in temporary proximity.
- **Random pedestrian mobility model** is a mobility model which describes the movement of pedestrians and it is inspired by the Brownian movement.

- **Random Waypoint (RWP)** model is a commonly used synthetic model for mobility, e.g., in Ad Hoc networks. It is an elementary model which describes the movement pattern of independent nodes by simple terms. Briefly, in the RWP model: each node moves along a zigzag line from one waypoint to the next; the waypoints are uniformly distributed over the given convex area, e.g. unit disk; at the start of each leg a random velocity is drawn from the velocity distribution; optionally, the nodes may have so-called "thinking times" when they reach each waypoint before continuing on the next leg, where durations are independent and identically distributed random variables.

## 2 OPPORTUNISTIC LOCALIZATION AND TRACKING

*CONTRIBUTORS: CNIT-PD, CNRS-LAAS, ISMB, RWTH*

The present JRA involves researchers belonging to four institutions, namely CNIT-PD, CNRS-LAAS, ISMB and RWTH Aachen University. The aim of the JRA is to devise, design and analyze novel opportunistic schemes aimed at enhancing the self-localization and tracking functionalities of some mobile nodes in a given area. In the following, an extensive report of the results obtained is given.

### 2.1 Research activities

Localization and Tracking are very interesting problems that have been deeply studied in several different contexts, thanks to the large set of possibilities and optimization that might be enabled by knowing the geographical position of the nodes in a communication system. Despite the considerable effort spent on these topics, a number of open problems still need to be investigated.

In this JRA, we tackle the problem from a different and rather new perspective: exploiting the device heterogeneity and the opportunistic communication paradigm to improve the self-localization and tracking capabilities of the nodes. In this way, we can also use cheap devices due to opportunistic information exchange between nodes with different localization capacity. In particular, we focus in indoor scenarios, where localization is still an open problem due to impossibility of using GPS system.

We envision an indoor scenario where nodes have different mobility patterns, including static (Access Points, Beacons, static Motes), periodic and/or pre-planned (elevators, mobile stairs, stairlifts, robots), preferential (errand girls, office workers, maintenance men), and random. Furthermore, nodes might be equipped with a different number of wireless communication interfaces, such as Bluetooth, WiFi, 3G, Mote, and so on. We also assume that some nodes might be equipped with sophisticated localization modules (as cricket, MEMS, indoor GPS), whereas others might be able to perform only a simple (and rather unreliable) RSSI-based localization. All these nodes are assumed to be able to seamlessly and opportunistically interact to obtain certain objectives and, in particular, to improve their own localization estimate.

The JRA will investigate how the localization error of a node can be reduced by opportunistically exchanging information with other nodes.

The JRA is essentially divided in two parts: the first one has the objective of understanding and modeling the mobility of the nodes in a realistic indoor scenario and the opportunistic interaction between two nodes; the second one is focused on the localization problem and aims at modeling the localization enhancements that can be obtained by exploiting the opportunistic data exchange.

### 2.2 Related works

Self-localization problem has been investigated in a number of papers. Most common localization methods consist in measuring the power of the received RF signal (RSSI), the Time of Arrival (ToA) or the Angle of Arrival (AoA) of the RF signals from the beacons. In this way, every node estimates a set of distances from the beacons and, then, guesses its position by means of lateration and triangulation techniques [1, 2] or by using statistical estimation methods [3]. Overviews of localization techniques based on RSSI and ToA measurements can be found in [4–6]. Multi-step localization techniques, which involve a number of successive refinement phases, have been proposed by Savarese [7] and Savvides [2]. Other solutions leveraging on specialized and complex hardware and infrastructure are given in [8–10]. When nodes (either static or mobile) can detect each other, then it is possible to devise cooperative position estimate techniques, which are very well studied in robotics. In [11] the authors utilize Markov localization for self-localize nodes and, then, probabilistic methods to synchronize robots estimate when they have a contact. Collective localization based on a distributed Kalman Filter is proposed in [12], whereas an anchor-free approach where robots infer their position estimate on the basis of the only information exchanged among them is proposed in [13].

## 2.3 Linear matrix inequality opportunistic localization scheme

CONTRIBUTORS: CNIT-PD, CNRS-LAAS

This activity has resulted in two joint works. One has been presented at the NEWCOM++ - ACoRN Joint Workshop, Barcelona, March 30-April 1, 2009, and the other has been submitted to IEEE International symposium on Intelligent Signal Processing - 26-28 August 2009 Budapest, Hungary, with the following title: Francesco Zorzi, GuoDong Kang, Tanguy Pérennou and Andrea Zanella - Opportunistic Localization Scheme Based on Linear Matrix Inequality.

### 2.3.1 Scenario and main assumptions

In our scenario, we consider a system made of mobile *Nodes* equipped with a common communication device (WiFi, Bluetooth or ZigBee). We suppose one node, called *User*, is not capable of self-localization, whereas the other nodes, named *Peers*, can perform self-localization with a certain accuracy that, in general, varies in time. A given Peer  $i$  can maintain a list of past *self-positioning estimations*. The problem we address is how self-positioning estimations of Peers can be used by a User to estimate its own position.

Every node in the network is equipped with a common wireless communication interface that is used for (opportunistic) data exchange. Radio propagation is described by means of a simple unit-disk model, according to which the radio transmission is always correctly received within a distance  $R$  (*coverage range*) from the transmitter, whereas it is not received at longer distances. Although the unit-circle model is known to be oversimplified, it permits to isolate the performance analysis from the characteristics of the radio interface that, at this stage of the work, is left generic.

We assume that nodes can communicate only during a certain period of time, the so-called *Scan Phase*, which may correspond to an interlaced Inquiry/Scan phase of Bluetooth [14] or to the Active Scanning procedure of IEEE 802.11 systems [15]. The scan phase is repeated with period  $T$ , asynchronously and independently by each node, so that the offset between the scan phases of two nodes can be modeled as a random variable with uniform distribution in the interval  $(0, T)$ . The ratio between the scan phase and the entire cycle time  $T$ , is called *duty cycle* and denoted by  $\delta$ . Whereas the scan period  $T$  is the same for all the nodes, we suppose that each node can fix its own duty cycle depending on the requirements and the management policy of that node.

We suppose that opportunistic data exchange can occur (in a negligible time) only when the scan phases of the two nodes overlap in time. Furthermore opportunistic data exchange also requires the nodes to be mutually in range. We assume that opportunistic interaction immediately takes place as soon as both conditions are satisfied. Such an event is coined *rendez-vous*.

We assume that peer nodes have “native” self-positioning capabilities, provided by some (non opportunistic) scheme. Accordingly, we denote by  $P_i$  and  $\hat{P}_i$  the real and the self-estimated position of peer  $\#i$ . Peers can be classified in different classes, depending on their native self-localization accuracy. For simplicity, we assume that the estimation error  $e_i = \|P_i - \hat{P}_i\|$  can be modeled as the module of a 2-dimensional Gaussian Random Variable  $[x(t) \ y(t)]$ , with zero mean and variance  $\sigma^2$ . The variance depends on the localization class that for simplicity we assume to be the same for all nodes during simulations. Moreover, the error model considers two possible characteristics: correlation among consecutive estimations (considering a tracking-based technique) and degradation of the estimate in time, so that the positioning error is better modeled as a stochastic process  $e_i(t)$ , with the following characterization:

- At the time  $t = 0$ , the positioning error  $e_i(0)$  is the module of a zero mean 2-D Gaussian Random Variable  $[x(0) \ y(0)]$ , with standard deviation  $\sigma(0)$
- At the time  $t > 0$ ,  $e_i(t)$  is calculated from the two coordinates  $[x(t) \ y(t)]$  drawn according to the correlated Gaussian distribution:

$$f(x(t)|x(t-1);\rho) = \frac{\exp\left[-\frac{\bar{x}(t)^2 - 2\rho\bar{x}(t)\bar{x}(t-1) + \bar{x}(t-1)^2}{2(1-\rho^2)}\right]}{2\pi\sigma(t)\sigma(t-1)\sqrt{1-\rho^2}} \quad (1)$$

where  $\bar{x}(t) = \frac{x(t)}{\sigma(t)}$  and  $\bar{x}(t-1) = \frac{x(t-1)}{\sigma(t-1)}$ . The parameter  $\rho$  is the correlation coefficient, which can vary in the interval  $[0, 1]$ , where  $\rho = 0$  means independent samples and  $\rho = 1$  means completely correlated (equal) samples. The same applies for the  $y$  coordinate.

The accuracy can degrade following the equation  $\sigma(t) = \sigma(0) + \alpha t$ , where  $\alpha$  is the drift of the estimation error.

During a rendez-vous, peer nodes send packets containing their estimated positions  $\widehat{P}_i$  and the class of accuracy  $\sigma^2(t)$ . This information may then be used by the User node to estimate its own position by means of the opportunistic localization mechanism described below.

As mentioned, the User node resorts to opportunistic localization to infer its geographical position. The opportunistic-positioning process requires the User to stop and stay at a fixed position for a given time interval  $W$ , during which the node collects the information opportunistically exchanged with passing-by Peer nodes. The localization time  $t$  is measured in number of scan periods, starting from  $t = 1$ . The opportunistic position estimation works in the following two stages.

1. At every scan period  $t$ , the User collects self-positioning estimations  $\widehat{P}_i(t)$  from each peer that are within radio range and whose duty cycles overlap the User's duty cycle (rendez-vous). Let  $eb_i = \max_t(e_i(t))$  denote an upper bound on the error between exact and estimated position of Peer  $i$ , so that

$$\|\widehat{P}_i(t) - P_i(t)\| \leq eb_i \quad \text{for } t \geq 1 \quad (2)$$

Furthermore, let  $P_u(t)$  be the exact position of the User. Assuming that communication is feasible only when the nodes are within the coverage range  $R$ , we then have

$$\|P_u(t) - P_i(t)\| < R \quad (3)$$

Therefore, for each Peer  $i$  within the range of the User at time  $t$ , inequalities 2 and 3 yield the following triangular inequality

$$\|P_u(t) - \widehat{P}_i(t)\| \leq R + eb_i \quad (4)$$

Collecting the inequalities (4) for all the peers in the coverage range of the User we get a Linear Matrix Inequality (LMI) that can be solved with standard techniques [16].

The resulting solution is used as a *raw (LMI) estimation*  $\widehat{P}_{u,r}(t)$  of the user position. Fig. 1 shows how  $\widehat{P}_{u,r}(t)$  is generated at cycle  $t$ , assuming that only  $P_1$  and  $P_2$  are within the User's range at time  $t$ .

2. When  $t > 1$ , the user can compute the barycenter of the primary estimations computed since  $t = 1$ . We define this barycenter as the self-positioning estimation of the user at time  $t$ :

$$\widehat{P}_u(t) = \frac{\sum_{k=1}^t w_k \widehat{P}_{u,r}(k)}{\sum_{k=1}^t w_k}, \quad t \geq 1 \quad (5)$$

where  $w_k$  is a weighting coefficient which is proportional to the number of Peers that have contributed to the  $k$ th raw LMI estimate.

This second stage is illustrated in Fig. 2, which shows how  $\widehat{P}_u(1)$ ,  $\widehat{P}_u(2)$  and  $\widehat{P}_u(3)$  are generated from  $\widehat{P}_{u,r}(t)$ ,  $t = 1, 2, 3$ , with all weights  $w_k$  equal to 1.

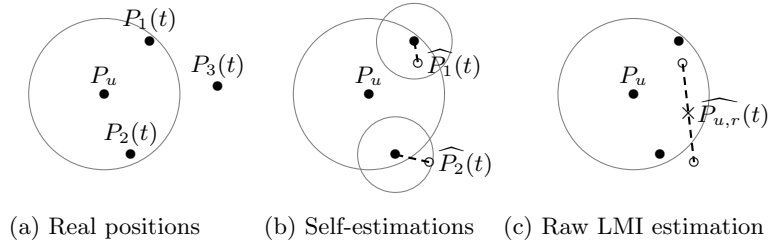


Figure 1: Raw LMI-only estimation.

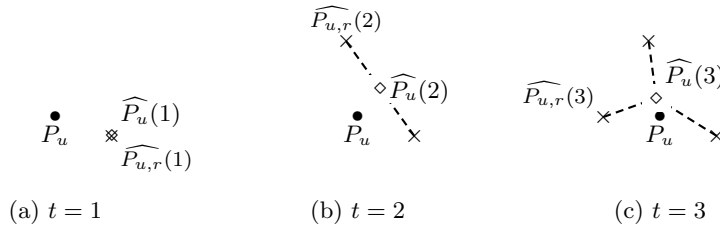


Figure 2: LMI+barycentric estimation.

We have made numerous experiments with this model, and observed that in most cases, the self-positioning estimation improves over time. We therefore use the estimation only after a warm-up time denoted  $wu$  and measured in scan periods starting at  $t = 1$ .

### 2.3.2 Results

Our reference case involves  $N = 100$  peer nodes moving in a  $100 \text{ m} \times 100 \text{ m}$  square and one user node remaining at the center of this square. Peers and user share the same radio range  $R = 10$  meters, so that only a fraction of Peers are within range of the user at each time.

Peers and user also have the same scan period  $T = 1$  second and the same duty cycle  $\delta = 50\%$ , so that duty cycles are always partially overlapped. The scan period of the user starts at  $t = 0$  while the scan period of each peer starts with an offset uniformly distributed in  $(0, T)$ .

The self-positioning estimations of each peer are generated as follows. First, the trajectory is computed using the Random Pedestrian Mobility Model defined in [17]: this model is inspired by the Brownian movement, modified so that speeds are drawn from a Gaussian distribution  $N(1.2, 0.2)$  and at each time step the next direction is chosen in front of the pedestrian, *i.e.* in another Gaussian distribution centered on the previous direction, with a small standard deviation arbitrarily set to  $\sigma_{\text{dir}} = \pi/6$ . The trajectory is kept within the considered square area. Second, for each position a self-estimation is produced using the peer self-positioning model defined in Section 1. In the reference case, the accuracy class of each peer has been set to  $\sigma = 1$  meter and it is assumed constant over time, *i.e.*  $\alpha = 0$  m/s. Furthermore, the self-positioning estimates are not correlated, *i.e.*  $\rho = 0$ . In practice, each peer self-position estimation at cycle  $t$  is randomly drawn in a disc centered around the exact position of the peer at cycle  $t$ , using a 2D Gaussian distribution;  $eb_i$  is the value such that  $[0, eb_i]$  is the 99 % confidence interval for the positioning error module  $e_i(t)$ . Different settings for the self-positioning model will be tried later in this section.

User, placed in the center of the area, estimates its position using the opportunistic localization model defined in Section 5. The opportunistic-localization time for the user is set to  $W = 2$  minutes and the warm-up time is set to  $wu = 30$  seconds. We will also see what happens for shorter and longer waiting times. The performance of the User's opportunistic-positioning scheme is evaluated in terms of distance between real and estimate position  $\|P_u - \hat{P}_u(t)\|$ .

The accuracy  $A$  of each run is the average localization error after the warm-up time. The results, in terms of localization error of the user node, strongly differ from one run to the other. The mean of the accuracy over 30 runs is  $\mu_A = 1.13$  m and the standard deviation is  $\sigma_A = 0.45$  m, while the worst case

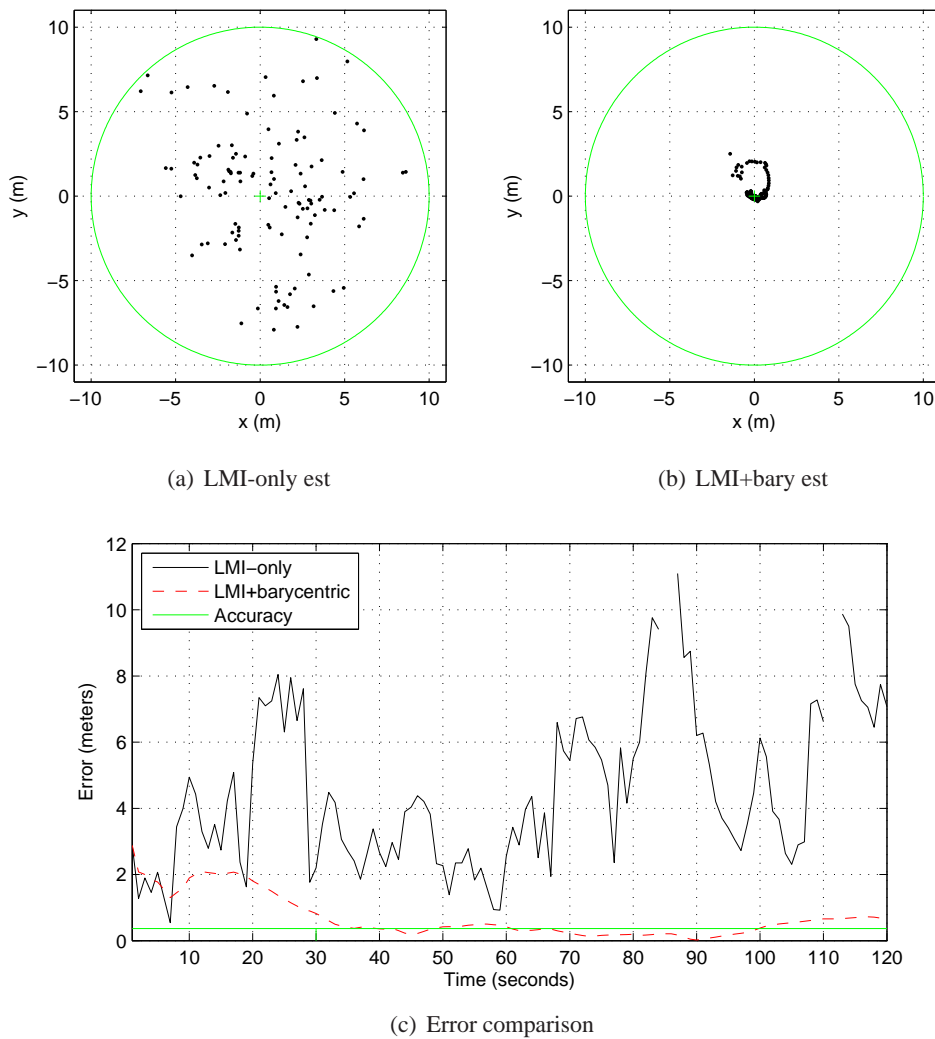


Figure 3: Reference case run #1/30.

has an accuracy of 2.28 m. These wide variations are likely to be ascribed to the different trajectories of peers in different runs. In fact, depending on the random seed of the run, peers may be widely spread in space, thus permitting good LMI-only localization and, in turn, good LMI+barycenter estimation, or they may be unevenly distributed in the area forming a small number of groups, a situation that yields to poor LMI-only localization and, consequently, to a degradation of LMI+barycenter performance.

To better understand the behavior of the protocol, we report in Fig. 3(a) the successive user's raw LMI estimations for a single run and in Fig. 3(b) the self-localization estimations of the user using LMI and barycenter algorithm. In Fig. 3(b), the oldest plots are "far" from the user position and gradually get closer, while in Fig. 3(a) old and new positions are equally distributed around the user position. The barycentric estimation clearly improves over time, and is better than the raw one. This is remarked in Fig. 3(c), where the reader can compare the evolution of the raw error  $\|P_u - \widehat{P}_{u,r}(t)\|$  and the error of the barycentric approach  $\|P_u - \widehat{P}_u(t)\|$ . The run-wide accuracy  $A$  is also plotted.

In most runs, the accuracy of the barycentric estimation tends to improve over time: each additional raw LMI estimation contributes to improve the estimation, since new information is added.

Then we study the impact of difference parameters on the accuracy  $A$  of the localization.

Taking into account the duty-cycle, we change the value of  $\delta$  using  $\delta = \{20\%, 40\%\}$ . As expected, accuracy improves when the duty cycle increases thanks to the higher number of peer self-positioning estimations that improves the performance of the raw LMI location estimation scheme and, in turn, the

barycentric estimation. Also energy consumption is strongly affected by duty cycle, but we do not consider this aspect in our study.

Then, we consider three different parameters: first, the correlation coefficient  $\rho$  among successive self-positioning estimations of each peer; second, the self-positioning accuracy class  $\sigma$  of peers; third, the accuracy drift  $\alpha$  of peers. We found out that the parameters have negligible impact on the accuracy of the opportunistic localization scheme that, hence, proves to be rather robust to localization errors of Peers. This is likely due to the fact that, despite the errors, the positions provided by the Peers form a uniform “cloud” of points around the User. Then, applying the barycentric scheme, the User always localizes itself near the center of such a cloud. To verify this conjecture, however, we plan to consider in future work other error models for peers estimation, such as model for podometers, or for MEMS-based inertial navigation systems, or for RSS-based landmarks.

In previous papers [17, 18], we also studied the impact of other parameters; we showed that the accuracy of the user self-positioning scheme degrades when: the amount of peers within range ( $N$ ) decreases, the range threshold  $R$  increases or the peers mean speed  $\mu_{\text{speed}}$  decreases. We re-evaluate these parameters and others quickly here.

For the setup used here, using 50 peers gives a mean accuracy of 1.76 m while 200 peers give a mean accuracy of 0.77 m (this is not as overcrowded as it may seem, if you think of a station, a big mall or a conference room for instance: in a  $100\text{m} \times 100\text{m}$  square, this gives  $50 \text{ m}^2$  per peer). Of course, the more peers there are with random trajectories, the more communication opportunities there are, and the more information are fed to the LMI system, which induces better estimations.

Another way to improve the accuracy is to increase the waiting time of the user: 5 minutes lead to an accuracy of 0.91 m. In that case, the barycentric estimation takes into account more and more raw LMI estimations, thus giving less weight to bad raw estimations. On the contrary, reducing to 1 minute degrades the accuracy to 1.67 m.

We also changed the radio coverage range. A 5 m range leads to an accuracy of 1.01 m, while a 20 m range leads to an accuracy of 1.86 m. This is not an intuitive result, since a larger range would mean more opportunities for sharing information. However, these additional positions are more far away from the user, which increase both the raw LMI error and the barycentric error.

Finally, we also changed the mean peer speed. If peers are slow (0.6 m/s) the accuracy degrades to 2.14 m. If peers are fast (3 m/s) the accuracy improves to 0.68 m. When the speed increases, positions taken into account will largely vary between two successive LMI-only estimations. This diversification of spatial information improves the behavior of the barycentric estimation.

In all the cases considered in this study, we obtained a localization error lower than 2.5 meters that can be reduced to less than 1 meter with an accurate tuning of the system parameters. In particular, the duty cycle of the opportunistic-scan phase has been observed to have a significant impact on the user self-positioning estimation: the shorter the duty cycle the less the rendezvous probability with peers and, in turn, the lower the localization accuracy. Furthermore, we observed that the proposed opportunistic localization scheme is rather robust to the self-positioning error model for Peers. In fact, the correlation, the standard deviation and the drift of the self-positioning error do not significantly affect the localization accuracy, provided that the algorithm is performed over the data gathered with a large enough number of opportunistic exchanges.

### 2.3.3 Future work

In order to complete the work, we are intending to do some improvements. A first issue to take into account is the interference and in general the impact of the channel behaviour between nodes and how this can affect the performance. Furthermore, a more realistic set-up can be investigated, involving different types of peer nodes, e.g. access points with well-known positions but only partial coverage and mobile peers carrying cheap Inertial Navigation Systems (INS) which accuracy drifts over time. We will also implement the opportunistic meeting model defined in [19] that applies to peer meetings. It is also possible to take into account different self-localization models and opportunistic update, as discussed

in next section.

## 2.4 Analysis of self-localization error models and opportunistic enhancement

*CONTRIBUTORS: CNIT-PD, RWTH, ISMB*

The second task, which is (to some extent) complementary to the first one, mainly addresses the aspects concerning the modeling of the self-localization error and the techniques to gain advantage from the info received from other nodes. The mobility and the link establishment aspects, therefore, might be superseded considering some simple (though acceptable) models (single interface with given coverage range, simple distribution of the in-range time).

The main idea is the meeting model and the opportunistic update published in [19].

In this work, a very simple but accurate scenario is drawn: nodes move and they can exchange information about their localization estimate, i.e. the estimated position and the reliability of that estimation, according to the meeting model that provides a periodic scan phase following a duty-cycle. Moreover they can infer the distance from the communication.

In a first step we assumed a very common scenario. Self-localization error is modeled as a 2D-Gaussian Random Variable, with zero mean and variance  $\sigma^2$ ; the distance estimation is inferred through the received power of the packet according to a path loss and shadowing channel model following this equation

$$P_{Rx} [dBm] \simeq P_{Tx} + K - 10\gamma \log_{10} \left( \frac{d}{d_0} \right) + \psi(t, \mathbf{x}).$$

where  $P_{Tx}$  is the nominal transmission power (in dBm),  $K$  is a unitless constant that depends on the environment,  $d_0$  is a reference distance for the antenna far field,  $d$  is the real distance between transmitter and receiver,  $\gamma$  is the path loss coefficient and  $\psi(t, \mathbf{x})$  is a gaussian variable with zero mean and variance  $\sigma_\psi^2$  that models the shadow fading. Therefore the estimated distance is modeled as

$$\hat{d} = d \cdot 10^{\frac{\psi(t, \mathbf{x})}{10\gamma}}$$

The opportunistic update is performed using a Maximum Likelihood approach.

Results are interesting and we would like to better understand how different estimation models perform using the same update algorithm and how different update algorithms can apply to different models in order to increase accuracy.

To better appreciate the effect of opportunistic localization, we define the *opportunistic gain* metric  $\Delta_i$ ,  $i = A, B$ , as

$$\Delta_i = \frac{\sigma_i \sqrt{\pi/2} - \tilde{\epsilon}}{\sigma_i \sqrt{\pi/2}} \quad (6)$$

where  $\tilde{\epsilon}$  is the mean localization error after the opportunistic localization, whereas  $\sigma_i \sqrt{\pi/2}$  is the mean localization error of the node obtained by using the native localization scheme.  $A$  denotes the better localized node compared to  $B$ . Therefore,  $\Delta$  represents the relative gain in the localization error obtained by using the opportunistic scheme.

Fig. 4 reports the results obtained for an heterogeneous scenario, when varying the duty cycle  $\delta$  and the coverage range  $R$ . We observe that also in this case the best performance is obtained by setting  $\delta = 0.5$  for every coverage range, though the performance improves for small value of  $R$ , as expected. Nonetheless, the scheme offers a 20% of gain even when  $R = 4$  m, which is a reasonable distance for this type of interactions.

Fig. 5 shows the same results, but for an homogeneous scenario in which both nodes have poor native self-localization capabilities. The results are substantially similar to those of the former case, though the curves are now more compacted and the relative error gain is reduced. Nonetheless, we observe that in this situation, the initial localization errors are very large, so that a gain of 25% is appreciable.

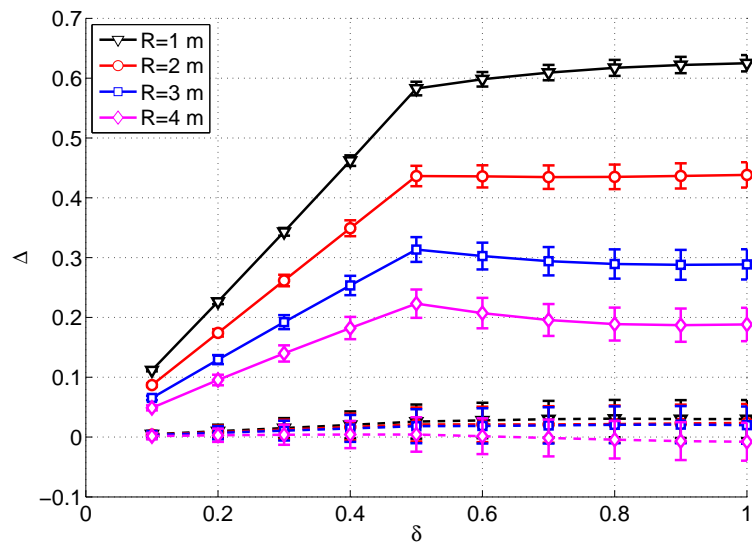


Figure 4: Relative localization error gain after an opportunistic update for different values of range  $R$  in the heterogeneous scenario ( $\sigma_\psi = 5$ ). Solid/dashed lines refer to node A/B.

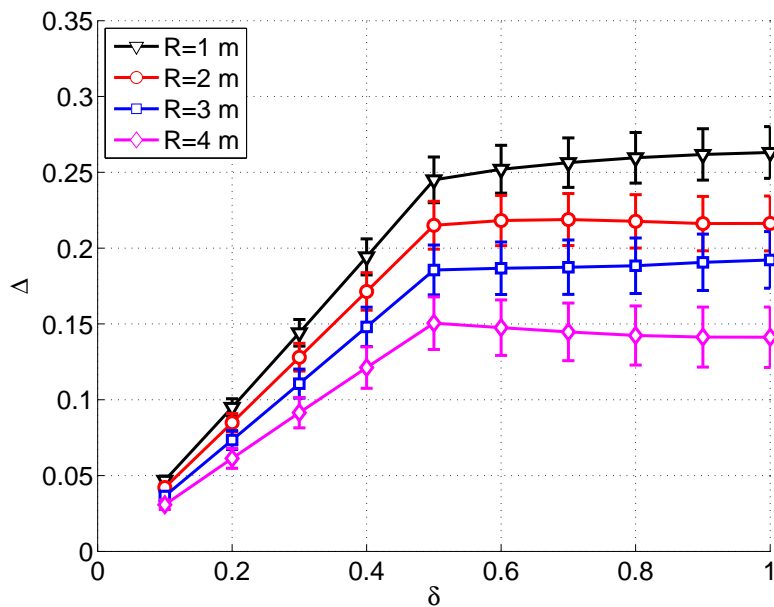


Figure 5: Relative localization error gain after an opportunistic update for different values of range  $R$  in the homogeneous scenario ( $\sigma_\psi = 5$ ).

#### 2.4.1 Future work

We are working in order to find out other realistic self-localization error models, that follow some real localization application, like Cricket or Radar system. We have a lot of real data measured using our testbed so that we can model the behaviour of localization error.

On the other hand, we are trying to figure out some statistical approaches for the opportunistic update based on Monte-Carlo method and particle filtering, in order to develop a more refined technique to exploit the information that nodes have.

### 3 MATHEMATICAL MODELING OF INTERMITTENT BEHAVIOR IN OPPORTUNISTIC NETWORKS

CONTRIBUTORS: NKUA/IASA, CNIT-CT

This JRA involves researchers belonging to two institutions, namely NKUA/IASA and CNIT-CT. The main objective of this research activity is a study of intermittent dynamics in opportunistic networks using methods and models of statistical and mathematical physics. In the following, a report of the results obtained is given.

#### 3.1 Research activities

In an opportunistic network, connections between network nodes are not constant in time. Links may fade in a hostile environment, connections are rearranged because of mobility or power constraints, nodes can be activated and deactivated in a periodic or irregular manner. The flow of information through such a network takes on stochastic characteristics that can impact network performance. Phenomena like phase transitions, critical behavior and self-organization, which are usually associated with states of matter, may emerge and cause a network to exhibit a variety of behaviors in different regions of its parameter space. A solid understanding of the stochastic dynamics underlying these behaviors will be useful in providing intuition and quantitative tools for designing and optimizing such networks at every level.

There are a lot of factors that influence opportunistic network performance. These include mobility, intermittent connectivity/link fading, periodic activation/deactivation of nodes, opportunistic use of resources and others. This calls for a modular approach in the research of effective mathematical models for opportunistic network behavior. The impact of each factor must be studied separately before their synergies can be assessed. We start by concentrating on the effects of link fading, constructing one- and two-dimensional models of networks in which data flow is interrupted by the random fading of links between nodes.

In the 1d model, we have  $N$  nodes in a circular periodic arrangement, with data moving in one direction only. This can be easily generalized to the case of data flowing in both directions and/or an open line arrangement for the nodes, the dynamics being essentially the same. We take time to be made up of discrete time steps, and in each time step a packet of information arrives externally or is generated at node  $i$ , with a destination node  $j$  and a time stamp. If  $i = j$  the packet is 'absorbed' immediately. Otherwise, it enters in the queue of that node. In each time step, every node attempts to forward the oldest (bottom) packet in its queue to the top of the queue of the next node, with packets absorbed as soon as they reach their destination. The fading of the links is simulated by assigning a probability  $p$  to the event of having an open link between each pair of consecutive nodes. Links are open or closed independently of each other. In the extreme case  $p = 1$  the flow is uninterrupted, while at  $p = 0$  packets never leave the source node. In this scenario the network is characterized by two variable parameters,  $p$  and  $N$ .

We construct a mean field theory that describes the dynamics of averages in this model, by comparing the average flow of packets in and out of every node. If  $q_i(t)$  is the length of the queue of node  $i$  at time  $t$ , then, on average:

$$q_i(t+1) = q_i(t) + \frac{1}{N} \left(1 - \frac{1}{N}\right) + p \left(1 - \frac{2}{N}\right) \Theta(q_{i-1}) - p\Theta(q_i) \quad (7)$$

where  $\Theta(q) = 1$  when  $q > 0$  and 0 otherwise. The first variation term describes the external inflow of packets, the second is the inflow from node  $i - 1$  and the third is the outflow to node  $i + 1$ . The negative  $\frac{1}{N}$  correction in the first term takes into account the probability that source and destination are the same, while the negative  $\frac{2}{N}$  in the second term removes the packets coming into node  $i$  that are actually destined for it. Naively this would be  $\frac{1}{N}$ , but a simple combinatorial argument that counts the number of possible source nodes for the packets at node  $i$  destined for node  $j > i$  (modulo  $N$ ) shows the proportion to be

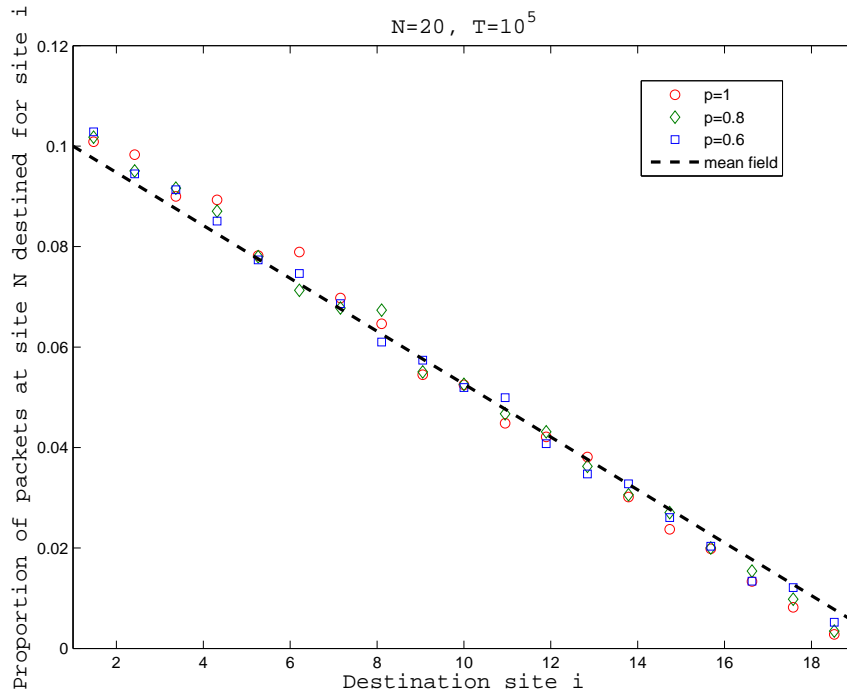


Figure 6: Proportion of packets at node  $N$  destined for node  $i$  in steady state.

$\frac{2(N-(j-i))}{N(N-1)}$ . For  $j = i + 1$  this is  $\frac{2}{N}$ . In the 1d case this counting argument agrees remarkably well with the simulation data, as can be seen in Fig.6.

Summing over all the nodes in the network we have for  $Q(t) = \sum_{i=1}^N q_i(t)$ :

$$Q(t+1) = Q(t) + (1 - \frac{1}{N}) - 2p\theta \quad (8)$$

where  $\theta = \frac{\sum_{i=1}^N \Theta(q_i)}{N}$  is the average number of nodes with a nonempty queue in the system. Since  $\theta \leq 1$ , the network reaches a stable functioning state only when

$$p \geq p_c = \frac{1 - \frac{1}{N}}{2}. \quad (9)$$

The critical probability  $p_c$  approaches  $1/2$  for large  $N$ . For  $p < p_c$  the system becomes unstable and queues increase linearly with time. The critical value of  $p$  predicted by mean field theory in (9) is borne out by the numerical data, shown in Fig.7.

This means that fluctuations do not significantly shift the value of  $p_c$  in 1d. They do impact crucial performance metrics however, such as the latency of packets, which is close to its average estimate  $\frac{N}{2p}$  only for large values of  $p$  and indeed diverges at  $p = p_c$  (Fig.8). The analysis of these fluctuations both in space and time (buffer correlations, response functions) also determines other important metrics, like buffer length distributions and reliability, and is one of our current research tasks.

Applying the same concepts to a 2d network we can examine the effects of intermittent connectivity in a setup that is both theoretically richer and more practically relevant. We consider an  $N \times N$  square lattice of nodes in the topology of a torus, with information flowing in only one direction in each dimension (bi-directional, linear variants are not essentially different) with one new data packet randomly entering the network at each time step while each node attempts to forward its oldest packet to one of its nearest neighbors, and if this fails to the other. At every step, each link is open with probability  $p$ , independently of all others. Each packet follows one of the minimum length paths to its destination, where it is removed

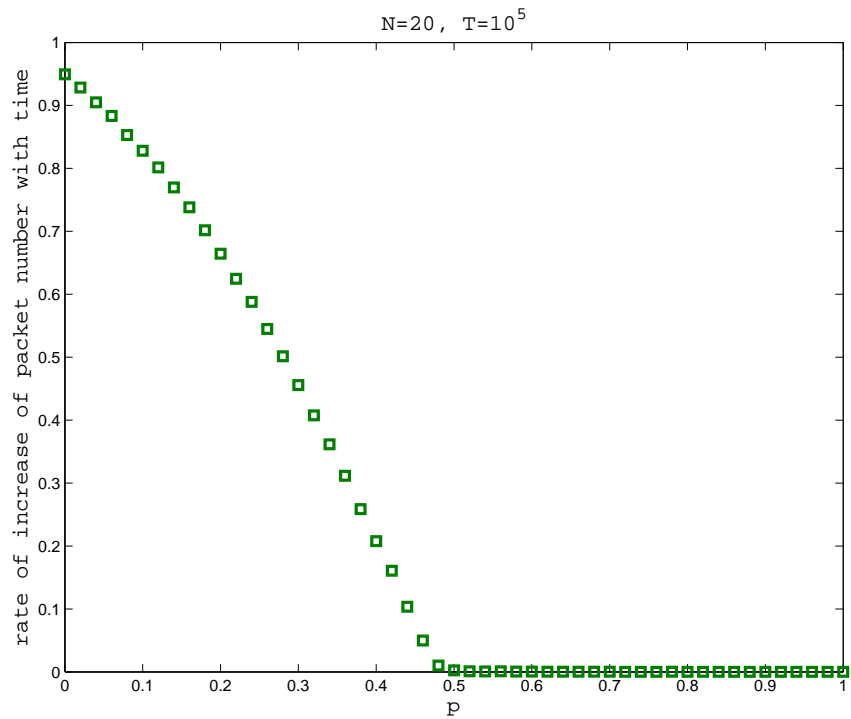


Figure 7: The rate of increase of the number of packets goes to zero at  $p = p_c$ . The network enters a steady state for  $p > p_c$ .

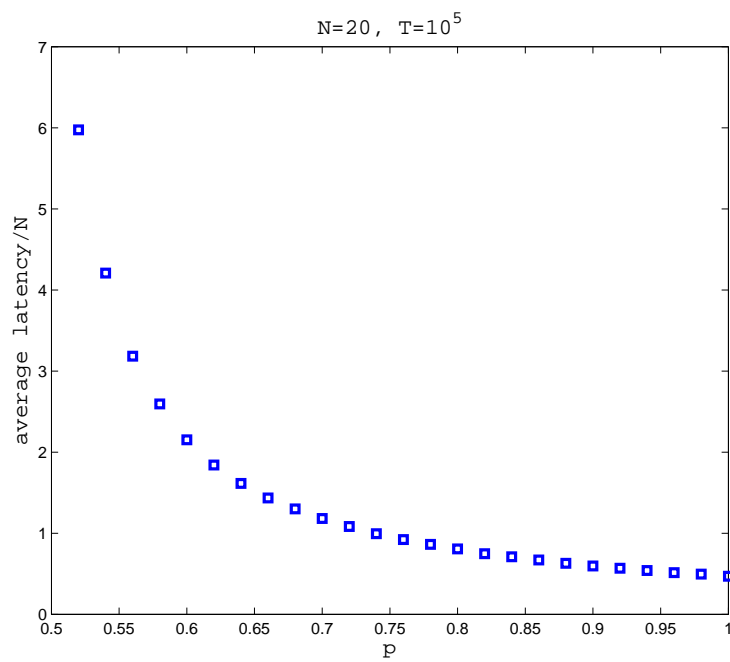


Figure 8: Normalized latency in the 1d model. The latency diverges for  $p \leq p_c$ .

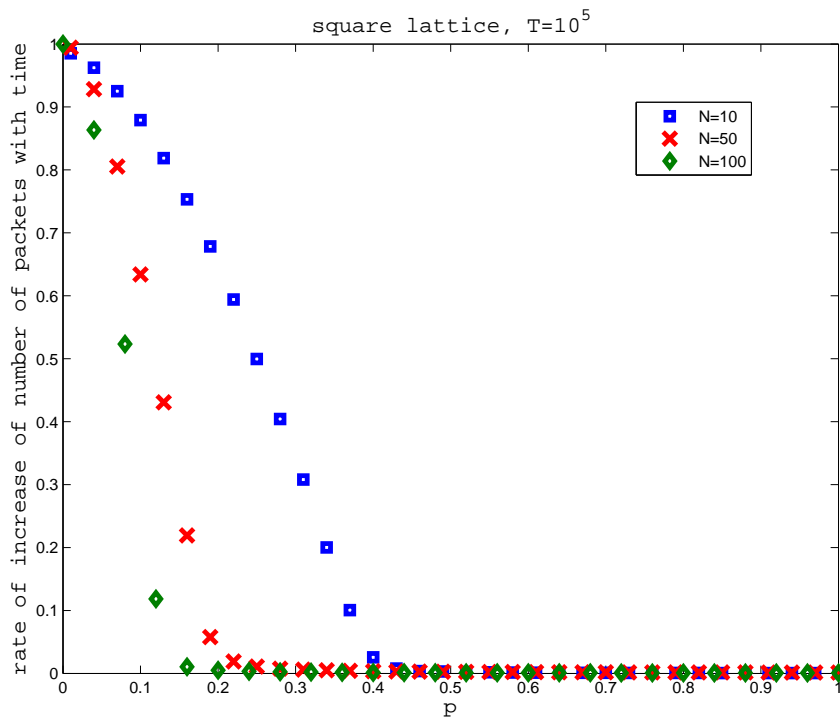


Figure 9: Rate of increase of the number of packets in the 2d model. The critical probability depends strongly on  $N$ .

from the pool. A mean field theory can also be constructed in this case and includes terms of second order in  $p$ . For node  $i|j$  we find:

$$\begin{aligned}
 & q_{i|j}(t+1) = \\
 & = q_{i|j}(t) + \frac{1}{N^2} \left(1 - \frac{1}{N^2}\right) + \frac{1}{2}(2p-p^2) \left(1 - \frac{4(N-1)}{N(N^2+2N-3)}\right) (\Theta(q_{i-1|j}) + \Theta(q_{i|j-1})) - (2p-p^2)\Theta(q_{i|j})
 \end{aligned} \tag{10}$$

or, if we look at the whole system at large  $N$ :

$$Q(t+1) = Q(t) + 1 - 4(2p-p^2)\theta \tag{11}$$

which predicts a critical open link probability of  $p_c \simeq 0.134$ . As before,  $\theta$  is defined as the average number of nodes with nonzero queue. Here, however,  $\frac{1}{N}$  connections are stable, changing the result for small but relevant values of  $N$ , as can be seen in the numerical simulation results in Fig.9. The critical probability is much closer to the mean field theory prediction for  $N = 100$  than for  $N = 10$ . These  $\frac{1}{N}$  corrections come both from combinatorial factors and from the details of the routing algorithm. For example, a node might attempt to forward a packet first along the direction where there is greater distance to be covered towards the destination node, or it might pick it randomly. We have used these two approaches in simulations and they give similar but not identical results. An analysis of these  $\frac{1}{N}$  corrections and also the fluctuations that determine important performance metrics like latency (shown in Fig.10) and reliability are our current research task. For both the 1d and 2d models, a precise characterization of the phase transition between the stable and unstable modes of the network is also of interest, though this might be more of a theoretical exercise, since it would be hard to fine-tune  $p$  to its critical value in an opportunistic network environment.

Opportunistic networking becomes *necessary* when it is not possible to establish an end-to-end path between the source and the destination. This can occur in various wireless communication scenarios. For example, it occurs when the network is sparse and therefore the nodes located between the source

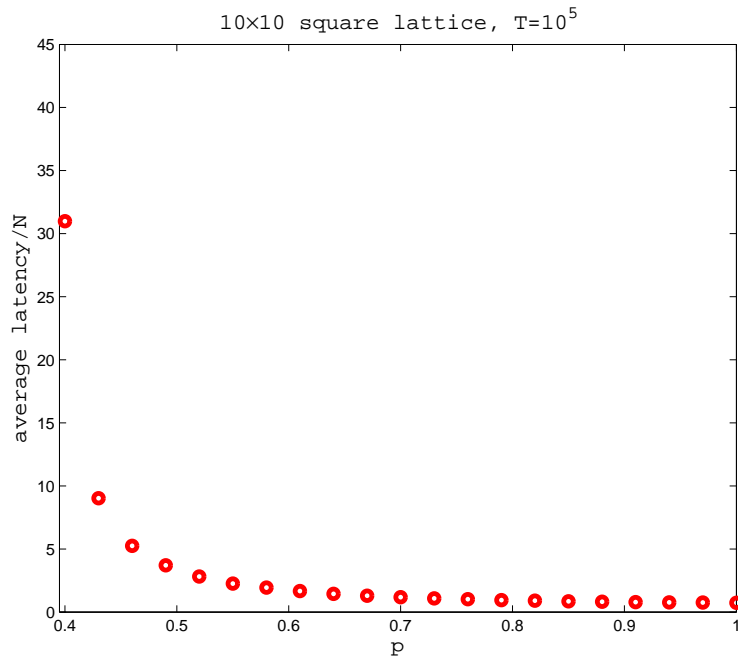


Figure 10: Normalized latency in the 2d model.

$S$  and the destination  $D$  are not sufficient to relay the packet all the way to  $D$ . In this case, networking can occur thanks to the mobility. In fact, intermediate mobile nodes can relay packets received by their current neighbors to other nodes as they become neighbors until the final destination is reached. Another possible scenario is when nodes are energy constrained and spend most of the time in an idle state so as to increase energy efficiency. In this case a node can forward the packet to its next relay only when such node becomes active.

We focus on the latter scenario. We will consider a network, modeled as a two dimensional lattice, in which node in position  $(i, j)$  is connected through a link to nodes in positions  $(i - 1, j)$ ,  $(i, j - 1)$ ,  $(i, j + 1)$ , and  $(i + 1, j)$ . Each of the links is active with a probability  $p$  that depends on the time interval spent by nodes in idle (i.e. OFF) and active (i.e. ON) states. In fact, in a first approximation  $p = T_{ON}/(T_{ON} + T_{OFF})$ . Observe that we assume that if a node has packets to be transmitted in its queue, it will remain active until the queue becomes empty.

Also we will consider the case in which *supernodes* appear in the network so that nodes in proximity of these supernodes can transfer to these nodes all the packets they have stored. The supernode will take the responsibility to deliver the packet to the destination and therefore, in the network perspective it is as if all the packets have been delivered.

Accordingly, the equation that regulates the size of the  $(i, j)$ -th queue at time slot  $t$ , which we denote as  $q_{i,j}(t)$  is given by the following equation:

$$q_{i,j}(t+1) = q_{i,j}(t) + p \cdot u(q_{i-1,j}(t)) + p \cdot u(q_{i,j+1}(t)) - \phi_{i,j}(t) \cdot [p \cdot q_{i,j}(t) \cdot (1 - u(q_{i,j}(t))) + q_{i,j}(t) \cdot u(q_{i,j}(t))] \quad (12)$$

where  $u(x)$  is the step function which is equal to 0 if  $x \leq 0$  and is equal to 1 if  $x > 0$ , and  $\phi_{i,j}(n)$  is a function which is equal to 1 if the supernode is within the radio coverage of the node in position  $(i, j)$  at the  $t$ -th time slot, and 0 otherwise.

Within this modeling framework we will study:

1. **The tradeoff between energy efficiency and delay.** In fact, if  $p$  is high then nodes spend most of their time in the active state. As a result, energy efficiency is low but we expect that delay decreases.

Also, second order effects should be considered. Indeed, if  $p$  is low, the nodes are idle most of the time, however, if a node has a packet to deliver, then it needs to remain active until the next hop in the path becomes active. In this state the node consumes energy which results in a decrease in energy efficiency.

2. **The performance increase which can be achieved by exploiting supernodes.** In fact, the presence of supernodes is expected to cause both an increase in energy efficiency and a decrease in delay.
3. **Identify the optimal mobility pattern for the supernodes.** We believe that the mobility pattern has a significant impact on performance. More specifically we expect that the faster are the movements of the supernode, the higher the positive impact on performance. We will try to derive guidelines for the design of supernodes mobility patterns.

### 3.2 Open issues and goals

Apart from the immediate goals concerning link fading, there are two important directions for our research activities in the near future. Both of these concern the issue of mobility. Either users are mobile and links are rearranged as time passes (the total number of users can also vary), or we have a “message ferrying” situation in which certain nodes can travel through the network and act as data carriers.

The (stochastic part of the) first kind of mobility can be modeled by a random reassignment of connections within the network, locally or globally, periodically or irregularly. The effect that this will have is not obvious a priori, since reshuffling the queues can move data packets both closer and farther from their destination. Also, the parameter space of the network grows, creating possibilities for more complex dynamics. With respect to the second kind of mobility, the *supernode* setup described in some detail in the previous section provides a rich framework for investigating the possibilities of message carriers providing connectivity in sparse or power-constrained networks.

Another interesting scenario is the queue-length-triggered network, in which nodes transmit data only when their buffer size exceeds a certain limit, in order to comply with power constraints. In this mode, the network exhibits a set of phenomena collectively known as self-organized criticality (SOC), with data propagating through the network in avalanche-like torrents and many important quantities following power-law distributions.

Finally, the ultimate goal of this JRA is to understand the combined effects of the aforementioned phenomena in the full parameter space of the network models. This will produce intuition for predicting the network behavior and designing protocols for opportunistic networking in a variety of conceivable situations.

## 4 TRANSPORT LAYER ISSUES IN OPPORTUNISTIC NETWORKS

*CONTRIBUTORS: CNIT-CT, KAU*

This JRA is the result of the collaboration among two institutions, namely CNIT-CT and KAU. It investigates on the benefits achievable through a packet aggregation technique and an enhanced MAC layer. Specifically, some interesting results obtained through simulations in a wireless mesh network have been obtained.

### 4.1 Research activities

Recently, Wireless Mesh Networks (WMNs) have attracted attention as a way to provide alternative Internet connectivity to rural areas, communities, and mobile users. In WMNs, wireless access points communicate with each other wirelessly, forming a true wireless mesh based access network of mesh relay nodes (MRNs).

WMNs can also provide a solution to the intermittent connectivity which is common in opportunistic networks, where devices are mobile and connections are not constant in time. As an example, mobile nodes in order to communicate with each other can connect to the wireless access points and use the WMN as a wireless backbone.

Due to the nodes' mobility and consequently the fast variation of the offered load, the wireless network shows very poor performance. The most common problems are: the high number of collisions, the PHY/MAC overhead, and the high delivery time.

The TCP protocol assumes packet losses as a congestion signal and consequently it reduces the transmission rate penalizing the overall throughput.

In such scenario, the network capacity can be significantly increased by opportunistically aggregating (or combining) several smaller packets into larger ones [14].

There are several benefits of packet aggregation in WMNs for TCP, the most important is that the overall number of MAC frames is reduced and collision risk between TCP flows are thus minimized. In this way it is possible to make the network robust to the variations of connected users.

When we consider a mesh network using an IEEE 802.11 technology, the gain obtained from data aggregation, without any modification to the MAC layer, becomes marginal because the inherent coordination problem of CSMA in multi-hop wireless networks dominates the performance.

Below we present preliminary results on the gains achievable through the use of packet aggregation as well as the use of an enhanced MAC layer. Future plans involve combining the enhanced MAC layer with packet aggregation to evaluate the combined effects of such approach.

### 4.2 Packet aggregation in wireless multi-hop networks

The basic idea of packet aggregation is to assemble several smaller packets into a larger one to save transmission time and reduce overhead and collision probability. The difference between packet aggregation and frame aggregation is that the latter operates at the MAC level and the former operates at the IP-level. The benefit of packet aggregation is that it can operate independent of the MAC/PHY standard, so it can be deployed with no need of adapting the chosen MAC/PHY layer [20].

The benefit of aggregating  $n$  packets into a single one can be quantified by the time saved at the MAC layer as:

$$t_s = t_o(n - 1) - \frac{8 * AH}{r} \quad (13)$$

where:

- $AH$  = size of aggregation header (in bytes)
- $t_o = DIFS + SIFS + averageChannelAccessDelay$  + the channel time for the PHY and MAC headers of one data frame and its MAC-level ACK frame ( $2 \cdot PHY + MAC + ACK$ ) (in seconds)

- $r$  = PHY layer rate (in bps)
- $n$  = number of packets (unaggregated).
- $t_s$  = saved time (in seconds) when aggregating  $n$  packets of same size.

It can be noticed that the benefit from aggregation increases with the number of (IP) packets that can be aggregated. In addition to the reduced overhead, the number of MAC layer frames will be reduced, which reduces contention and collision probabilities.

To have enough packets to aggregate, nodes usually introduce an artificial delay and queue packets [21]. When this additional delay expires and if the packet could not be aggregated, it is sent as it is. The right choice of such aggregation delay is an important design parameter, if the traffic load is low. Higher aggregation delays yields a higher aggregation rate, but also a higher end-to-end delay. With TCP under normal conditions, the effect of such delay on RTT is minimal.

Next we will present a detailed description of our packet aggregation algorithm.

#### 4.2.1 Proposed packet aggregation algorithm

In this section we present our algorithm for packet aggregation which is inspired by [22] because of its good performance with UDP traffic. The algorithm could be implemented either at the MAC layer or at the IP layer. However, for ease of deployment with current IEEE 802.11 hardware we choose to implement the algorithm at the IP layer.

Every mesh node marks incoming packets that potentially can be aggregated with a time-stamp and stores them in virtual “aggregation” queues. We use one virtual aggregation queue for each next hop. In addition, there is another virtual queue which holds all packets that should be sent un-aggregated (such as routing packets). Conceptually the virtual queues are seen by the upper layers as one Drop-Tail queue, where the current queue size is equal to the total number of packets in all virtual queues.

If a packet, when potentially can be aggregated, arrives at a mesh router and the virtual queue towards that next hop is empty an aggregation timer is set to timeout at  $packetArrivalTime + aggregationMaxDelay$ . There is one aggregation timer per virtual aggregation queue.

A dequeue event is triggered on either of three particular events: when a packet is enqueued, when the aggregation timer expires and when the MAC layer becomes idle. Whenever a dequeue event is triggered, four events can happen:

1. If there are aggregation queues with packets older than  $aggregationMaxDelay$  then the queue, aggregation or non-aggregation, with the oldest packet<sup>1</sup> is served first. This avoids starvation and packet reordering.
2. If there are no aggregation queues with packets older than  $aggregationMaxDelay$ , then the non-aggregation queue is served.
3. If there are no aggregation queues with packets older than  $aggregationMaxDelay$  and no packets in the non-aggregation queue, then the aggregation queue that contains packets of an aggregated size which is larger than  $MTU - aggregationHeader$  and has the oldest packet is served. This helps to improve flow fairness while still ensuring aggregation efficiency.
4. If neither of the above is fulfilled, then the aggregation queue stays idle until a new dequeue event happens.

All packets in an aggregation queue are aggregated up to  $MTU - aggregationHeader$ . If there is only one packet in a queue, the packet is sent as it is. If there are at least two packets, they are aggregated into a larger one and an aggregation header is added before the aggregated packet is passed to the MAC layer.

<sup>1</sup>For the aggregation queues the waiting time is subtracted with  $aggregationMaxDelay$ .

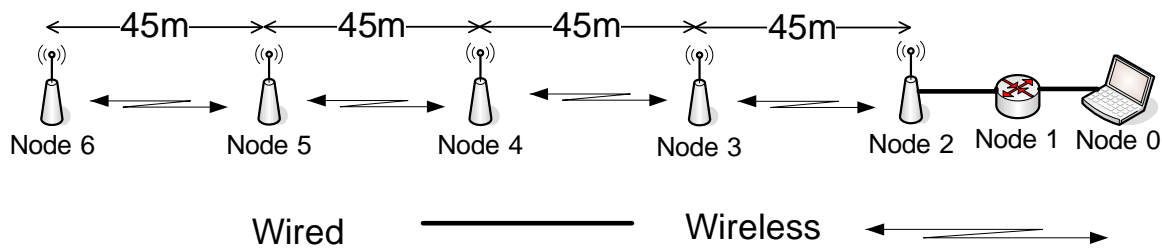


Figure 11: Simulation Topology.

If the MAC layer is idle and a new packet arrives that would make the resulting aggregated packet size larger than MTU or if an aggregation timer expires, the previous packets are aggregated and passed to the MAC layer and the aggregation timer is reset. When the dequeue event happens and the MAC layer is not idle the queue takes no action, as a dequeue event will happen as soon as the MAC layer becomes idle. If there are more packets in a queue than what would fit inside one MAC frame, then the queue will aggregate as many packets as fit inside the current MTU, while still respecting the packet order. The time stamp of the first packet that does not fit will set the new value for the aggregation timer. The aggregation header is an additional IP-header that adds an overhead of 20 bytes per aggregation packet.

The aggregation algorithm is controlled by *aggregationMaxDelay* which denotes the maximum forced delay for a single packet. When the network traffic is low, this parameter induces artificial delay, increases the number of packets in the queue and thereby increases the aggregation ratio. We choose 10 ms as it gives a good trade-off between aggregation efficiency and the additional delay introduced. If the network is highly loaded, queues will quickly contain enough packets to fill an entire MAC frame, which reduces the artificial delay introduced by aggregation. In that sense, aggregation automatically adapts to traffic load.

#### 4.2.2 Simulation results with aggregation

We conducted several simulations to investigate the impact of packet aggregation on TCP performance. We selected a typical WMN scenario where multiple flows traverse a common path towards an Internet gateway. To keep the analysis readable we modeled this as a string topology with 4 wireless links connected to 2 wired links (see Fig. 11).

We modeled all TCP flows as FTP-type file transfers between nodes 0 and 6, with infinite file sizes, and we varied the number of TCP flows according to Table 1. In simulations involving 4 or 20 flows, half of the flows are directed from node 0 to node 6<sup>2</sup>, while the other half of the flows are directed in the opposite direction. In that case, DATA and ACK packets from opposite FTP flows get aggregated in the same flow.

##### Aggregated goodput

Fig. 12 shows the aggregated goodput for 1, 4 and 20 flows. With aggregation enabled, goodput is higher than without aggregation and only slightly influenced by TCP packet size. The simulations showed a gain around 10 % when a large TCP packet size is used and around 50 % when a small TCP packet size is used.

##### TCP RTT

From Fig. 13, we can see that the TCP RTT was reduced when using a small TCP packet size and aggregation as compared to scenarios when no aggregation was used. This shows that even if we introduce artificial delay to be able to aggregate more, the TCP flows benefit from the reduced MAC contention.

<sup>2</sup>More precisely, in a FTP flow from node 0 to node 6, DATA packets go in that direction while ACKs go from node 6 to node 0.

Table 1: Simulation properties.

Property	Value
Simulator	NS 2.26 (extended with packet aggregation)
Nr of Flows	1, 4 and 20
Simulation time	910 s
Repetitions	25
Traffic time	Delayed by 30 s , Stopped after 908 s
TCP Sender	TCP/Sack1 (follows [23])
TCP Receiver	TCPSink/Sack1/DelAck (Selective and Delayed ACKs)
TCP Advertised window	5000
TCP Delayed ACK interval	100 ms
TCP ACK Size	40 Bytes
TCP/IP Header Size	40 Bytes (20 IP + 20 TCP)
TCP Packet Size	536 or 1460
TCP Traffic type	Infinite backlog (FTP)
Interface Queue	PriQueue or Aggregator
Queue Length	50 packets
aggregationMaxDelay	10 ms
aggregationHeader	20 Bytes
MAC/PHY	802.11a [24] (with extensions from [25] [26])
MAC MTU (Fragementation Threshold)	2304 bytes
PHY Basic rate	6 Mb
PHY Data rate	24 Mb
Sensing range	100 m
Transmission range	50 m
Wired Links	100 Mb/s, 1 ms delay, Droptail queue

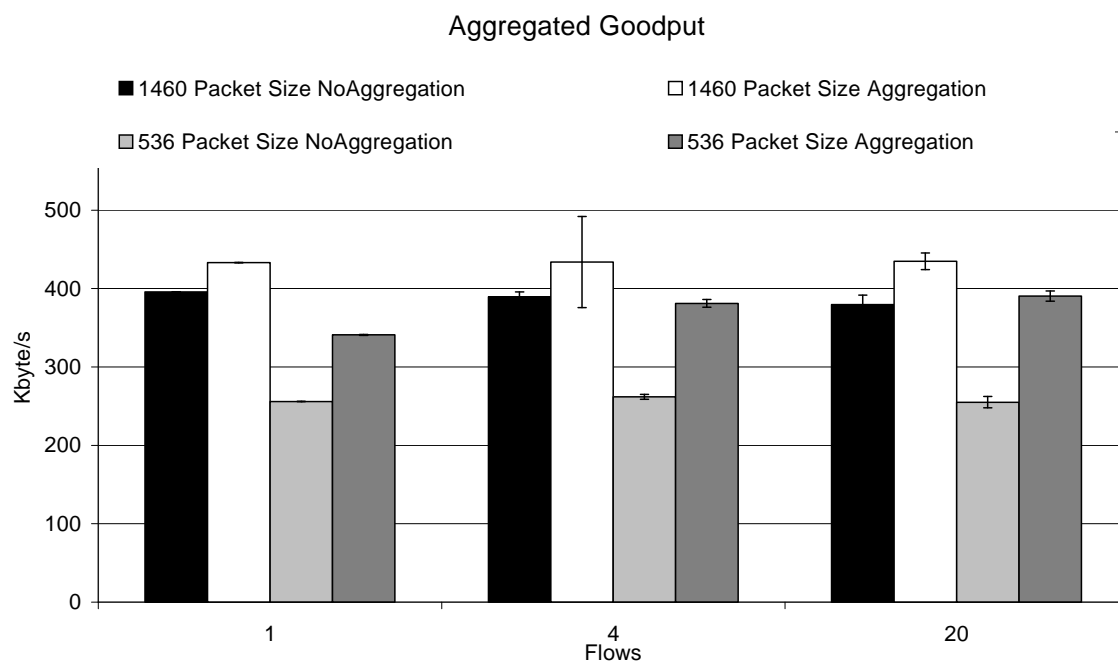


Figure 12: Aggregated Goodput.

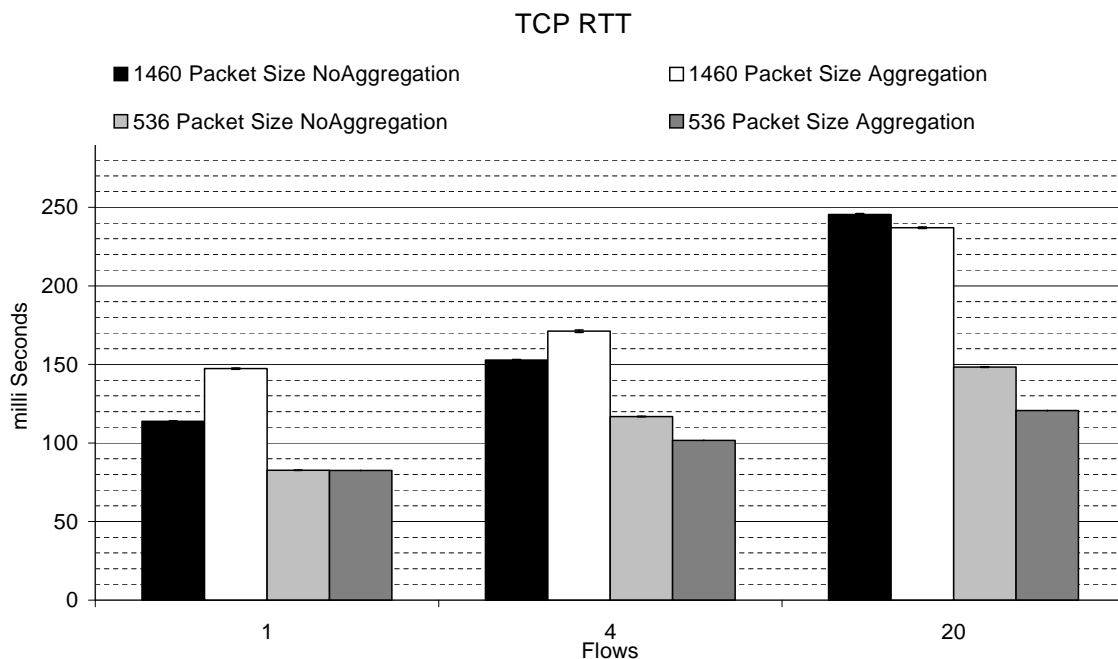


Figure 13: TCP round trip time.

In the scenarios where we use a large TCP packet size, the results are more contrasted. With 1 and 4 flows aggregation introduces extra delay in the TCP RTT. When using 20 flows, the TCP RTT decreases: this is due to a reduction of MAC layer contention followed from the aggregation of TCP ACKs.

### 4.3 Enhanced MAC layer

A key issue in order to exploit the advantage of packet aggregation is to improve the MAC layer so that the starvation and collision problems can be mitigated. Without any modification to the MAC layer, the gain obtained from data aggregation becomes marginal because the inherent coordination problem of CSMA in multi-hop wireless networks dominates the performance.

Several studies have demonstrated that when all transmitters are within the range of each other, CSMA protocols provide fair access opportunities to all flows. Unfortunately, in a multi-hop topology when transmitters are not in the range of each other, channel state information is necessarily incomplete because some transmitters cannot sense when other nodes are transmitting. This lack of information leads to poor performance, affecting the transmissions in different ways (i.e. starvation, unfairness), even if coordination enhancements such as RTS/CTS control packet exchanges, like in IEEE 802.11 DCF, are used [27]. In particular, a unfair throughput distribution arises in which a few flows capture all bandwidth while many other flows get very low or even zero throughput.

In these cases CSMA protocols show short- and long-term unfairness, high collisions probability and sometimes starvation. In fact, in similar scenarios, the possibility to acquire the channel becomes different among the terminals. As an example, we show what happens in the WMN shown in Fig. 14, when the IEEE 802.11 DCF is used.

Let us assume that node 4 has a packet to send and therefore it sends an RTS message. Its neighbor node 3, upon hearing this RTS, defer for a certain period of time. If node 5 is ready to answer, (i.e., it has not heard an RTS or CTS previously), it sends a CTS. Upon hearing the CTS, the neighbor of node 5 (i.e. node 6) remains idle for some time (i.e., do not send an RTS or respond with a CTS). Node 4 transmits a data frame to node 5, and node 5 acknowledges the successful reception with an ACK. As a result, during this period of time, the neighbors of node 4 and 5 remain idle.

In Fig. 14 we indicate with a cross (X) the transmissions which are blocked when the node 4 transmits

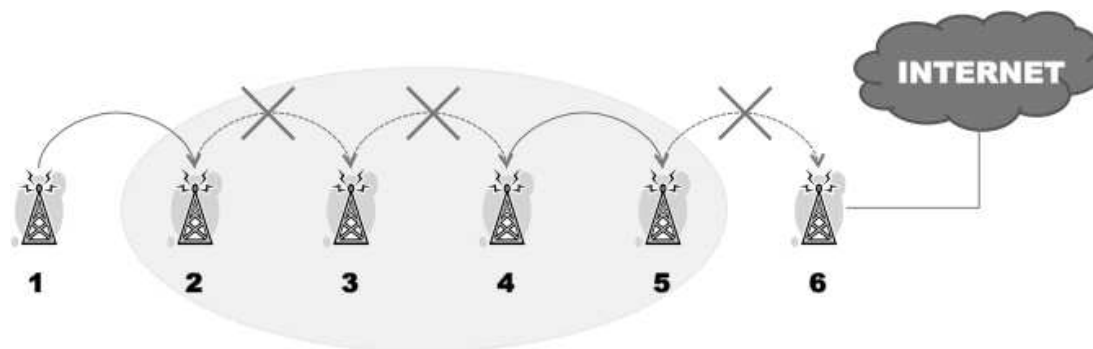


Figure 14: Chain topology.

to node 5.

Instead, node 2 does not sense any packet belonging to flow  $f_{4,5}$ . When node 2 wants to transmit a packet, it has to discover an available time-slot randomly, without any coordination with node 4, resulting in many attempts of node 2 without any response from node 3. Most of these random attempts occur in the middle of a transmission of flow  $f_{4,5}$ , and they result in some collisions at node 3 (see Fig. 15). Consequently, node 2 is forced to defer and to repeatedly double its contention window (CW), thus reducing the chances to start a new transmission in the next available time slot. As a consequence, its throughput approaches to zero (starvation).

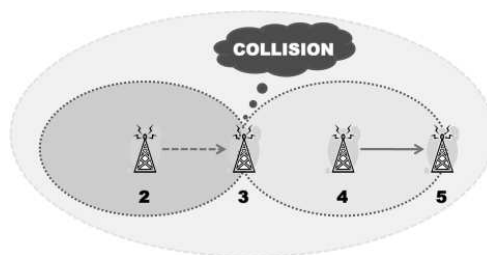


Figure 15: Example of packet collision at node 3.

In order to solve this problem, several approaches have been proposed. In the following section we describe our improvements to the MAC layer, when CSMA/CA is used, in order to solve the problem of lack of coordination between nodes in a multi-hop environment.

#### 4.3.1 Proposed enhancements to MAC layer

Here we propose a distributed modification of the IEEE 802.11 DCF, based on the tuning of the back-off mechanism using a feedback approach.

Specifically, we propose a distributed adaptive contention window-based scheme which works by using packet drop events at each node as a feedback to tune the IEEE 802.11 backoff mechanism of that node in order to reduce collision probability and to avoid starvation. Particularly, we refer to the chain topology shown in Fig. 14.

Our approach is similar to that in [28], where the authors propose a distributed modification to the IEEE 802.11 DCF standard to improve the efficiency of network coding techniques for TCP. We want to adapt these MAC modifications to also integrate the packet aggregation. In particular, the algorithm proposed in [28] adapts the backoff only in function of the node's interference range, instead our approach will tune the backoff depending on both the interference range and the packet size. The latter takes into account the effect of aggregation on the size of the packet forwarded.

The goal is to achieve a scheduling, shown in Fig. 16, which is collision-free and it is characterized by a high throughput and low collision probability.

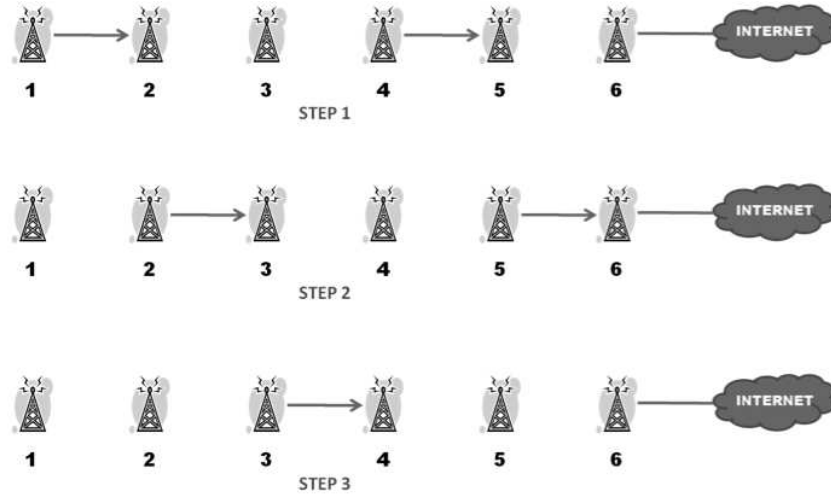


Figure 16: Collision-free scheduling.

The basic idea of the mechanism is the following. After a node has transmitted a packet, it defers its further transmissions until the transmitted packet crosses at least two hops. By doing so, the other packets transmitted by this node will not interfere with the transmission of the previous packet.

We allow the node to defer its transmissions by giving it a lower priority during the contention for the channel: as soon as a node transmits a packet and receives the corresponding ACK, it sets a large contention window, called  $CW_{large}$  (equal to a value we explain in the following), for a period of time equal to  $T_{sleep}$ . Once  $T_{sleep}$  expires, the node shrinks back its contention window to contend again for the channel.

We define  $T_{sleep}$  as:

$$T_{sleep} = (k + 1) \cdot T_S \quad (14)$$

where :

$$T_S = \overline{CW} + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK + DIFS \quad (15)$$

is the time needed for a successful transmission, and  $k$  is a parameter set in a different way for each node; hence, each node will have different  $T_{sleep}$ .

The value of  $k$  used by each node depends on its interference range and on the aggregation degree. We use a feedback from the events related to packet drops to compute  $k$  as explained in the following.

To calculate  $CW_{large}$  we refer to Fig. 14. Let us assume the interference range is equal to 1 for each node. According to the proposed mechanism, once node 1 receives an ACK from node 2, it has to defer its transmission during the transmissions of nodes 2 and 3. If both the transmission and the interference range are equal to 1, node 1 doesn't hear node 3 and consequently decreases its backoff. In order to avoid collisions between node 1 and node 3, we choose

$$CW_{large} = 2 \cdot T_S \quad (16)$$

so that the average backoff time is equal to  $\frac{CW_{large}}{2} = T_S$ .

Now, let us assume, for the above topology, that node 1 chooses the value of  $k$  as equal to 1. Consequently, node 1 sets its contention window to a large value,  $CW_{large}$ , after the transmission of the first packet, and shrinks it back to  $CW_{min}$  after a time equal to  $T_{sleep} = 2 \cdot T_S$ .

If the interference range is greater than 1 hop or the next nodes forward the original packet after aggregating other packets (note that in this way the value of  $T_S$  increases because the packet size increases as well), the next transmission of node 1 will likely collide with the packets which are being forwarded at the next hops.

If such collisions happen frequently (due to the incorrect estimate of  $k$ ), there would be several packets dropped because of the reached retry limit.

We use the value of dropped packets at that node as a feedback to improve node 1's estimate of  $k$ . In particular, when node 1's estimate approaches the correct value of  $k$ , the value of packets drop at that node reaches zero. Since the minimum value of  $k$  is 1, each node initially sets  $k = 1$  and for each packet drop, it increases  $k$  by 1. On the contrary, when no collision or drop event happens for a long interval of time, the node decreases  $k$  by 1.

Let us describe the mechanism through pseudo-codes. After a node has transmitted a packet, it sets a timer *ack\_timer*. If an ACK does not arrive during this time interval, the function *handle\_ack\_timer()* is invoked (see Fig. 17). The *retry\_counter* is incremented and compared to *retry\_limit*. If the *retry\_limit* is reached, the packet is dropped, *retry\_counter* is reset and  $k$  is increased in order to avoid other collisions.

---

### Function 1

```

handle_ack_timer()
  retry_counter++;
  if (retry_counter > retry_limit) then
    discard_pkt();
    retry_counter = 0;
    k++;
  else
    retransmit_data();
  end if

```

---

Figure 17: Pseudo-code of the procedure called when transmitter does not receive ACK.

When a transmission is performed correctly (in this case the node has received the ACK packet), the node calculates  $T_S$ , which is the time interval needed to perform a correct transmission, and then it sets the freeze timer equal to  $T_{sleep}$  (see Fig. 18). Then, the  $CW$  is set to  $CW_{large}$  and the backoff starts.

---

### Function 2

```

recv_ACK()
  if (pkt != NULL) AND (DST_ACK == Node_Address) then
    ack_timer.stop();
     $T_S = CW/2 + RTS + SIFS + CTS + SIFS + TxTime(pkt) + SIFS + ACK + DIFS$ ;
     $T_{sleep} = (k + 1) \cdot T_S$ ;
     $CW = CW_{large}$ ;
    extended_CW_timer.start( $T_{sleep}$ );
    backoff_timer.start( $CW$ );
  else
    discard_ACK();
  end if

```

---

Figure 18: Pseudo-code of the procedure called after an ACK reception.

After a duration of  $T_{sleep}$ , *extended\_CW\_timer* expires and *handle\_extended\_CW\_timer()* is called (see Fig. 19), which resets the  $CW$  to its minimum value.

#### 4.3.2 Simulation results with the enhanced MAC layer

In this section we consider the enhanced version of IEEE 802.11 and we comment the results obtained using the well-known ns-2 platform. For comparison, the results for the standard IEEE 802.11 DCF are also shown.

**Function 3**


---

*handle\_extended\_CW\_timer()*


---

```

CW = CWmin;
if (backoff_timer == ON) then
    backoff_timer.stop();
    backoff_timer.start(CW);
else
    backoff_timer.start(CW);
end if

```

---

Figure 19: Pseudo-code of the procedure called when *extended\_backoff\_timer()* expires.

Note that, as this JRA is recently started, these simulations do not feature packet aggregation yet; however we have considered different packet sizes in order to evaluate the algorithm behaviour when coupled with packet aggregation.

Each simulation was run for 100 seconds after a 5-seconds warm-up period. In the following we explain the results for the scenario represented in Fig. 14. A distance of 200 m separates each node from the previous and the following ones. Furthermore, we suppose both sensing range and transmission range equal to 250 m and we use for the physical layer the default values of the two-ray ground parameters. Other simulation parameters are summarized in Table 2.

Parameter	Value
Slot time $t_{slot}$	20 $\mu s$
SIFS	10 $\mu s$
DIFS	50 $\mu s$
RTS size	288 bits
CTS size	240 bits
DATA size CBR	250 to 1500 bytes
DATA size FTP	1000 bytes
ACK size	240 bits
$CW_{min}$	32
$CW_{large}$	$2 \cdot T_S$
Short Retry Limit	7
Long Retry Limit	4
Data rate	2 Mbit/s
Initial value of $k$	1

Table 2: Simulation parameters.

First, we have conducted some simulations assuming CBR traffic over UDP protocol in order to have only one-way traffic. Fig. 20 shows the throughput value vs. packet size, for the standard 802.11 MAC and the enhanced MAC. When the packet size is small, both algorithms behave in the same way, because the number of collisions is low. When the packet size increases, the throughput reaches a value of 500 kbps (which can be considered the maximum throughput for a 2 Mbit/s channel due to collisions, retransmissions and packet overhead) and the modified MAC outperforms the standard IEEE 802.11 protocol due to the reduced number of collisions.

We have then conducted simulations in order to show the behavior of our enhanced MAC when the TCP protocol is used. In this case we have considered a FTP traffic source. The results obtained are summarized in Table 3. They show that, even in this case, the modifications to the CSMA/CA lead to

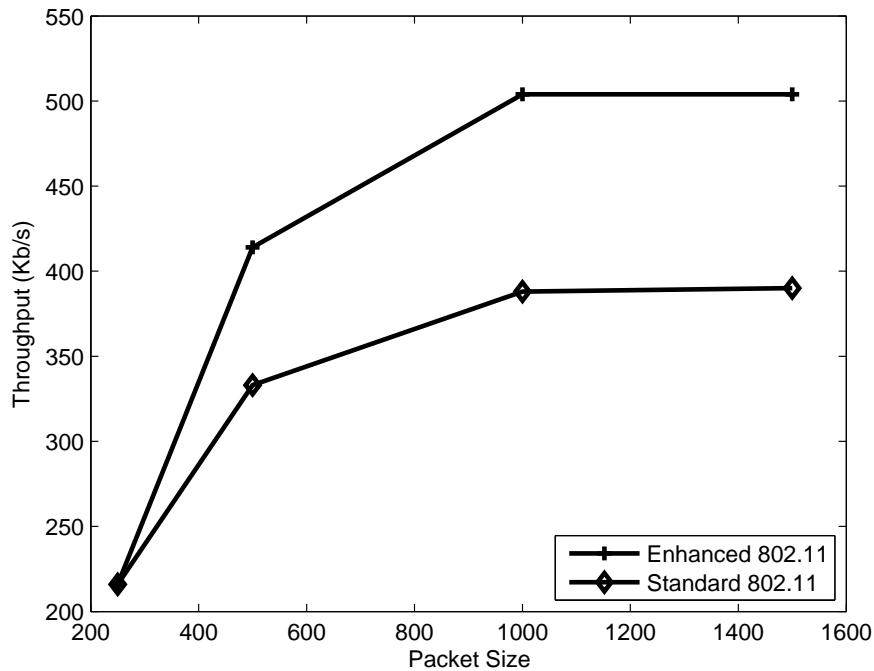


Figure 20: Comparison between Enhanced and Standard 802.11 MAC over CBR/UDP.

some improvements in terms of both throughput and delay as compared to the standard 802.11 MAC approach.

Metric	Adaptive 802.11 MAC	Standard 802.11 MAC
Throughput	48 Kbps	18 Kbps
Average Delay	0.2 s	0.33 s
Max Delay	7.51 s	9.84 s

Table 3: Comparison between Adaptive and Standard 802.11 MAC over TCP.

It is important to note that when TCP is used instead of UDP, we have two different flows, the first one from the source to the destination (DATA flow) and the other in the opposite direction (ACK flow). In this case, the modifications we have presented before do not achieve so high performances. This is due to the collisions caused by the ACK flows which are not taken into account in this first release of the proposed mechanism.

Performance can be increased thanks to both the data aggregation technique previously described in Section 4.2.1, and in particular by aggregating DATA and ACK packets together, and to the broadcast nature of the wireless channel. In fact, assuming packet aggregation, during each transmission, a node can transmit information helpful for both distincts flows.

#### 4.4 Future work

Future work involves combining the enhanced MAC layer with packet aggregation in order to exploit the advantage of the enhanced MAC over the aggregation technique when TCP flows are considered. We plan to mix together in the ns-2 platform the two proposed approaches and then evaluate the combined effects of such an approach through simulations.

## 5 PEER-TO-PEER TECHNIQUES IN OPPORTUNISTIC MESH NETWORKS

*CONTRIBUTORS: CNIT-CT, KAU*

The present JRA involves researchers belonging to CNIT-CT and KAU. The aim of this JRA was to investigate how some innovative and efficient P2P mesh networking protocols behave in dynamic and mobile environments. In the following we illustrate the results obtained.

### 5.1 Research activities

*Wireless mesh networks* are an emerging communication paradigm which recently received great attention [29, 30]. Experiences such as Seattle and Houston wireless witness the trend towards exploitation of realization of *wireless community networks*, which are becoming increasingly popular since the advent of cheap wireless technologies such as IEEE 802.11.

Given the *community* spirit of such networks, it is expected that users will be willing to share also non-communication resources, such as data, images, music, movies, disk quotas for distributed backup, etc. It is therefore likely that peer-to-peer applications will play a fundamental role in enriching the services offered by community networks.

The traditional architecture proposed for mesh networks considers a backbone composed of wireless routers (e.g. IEEE 802.11 or 802.16 wireless nodes) which provide long range connectivity to highly mobile wireless nodes. In the view of a community network, these peripheral nodes can be identified with peer devices which provide contents to be distributed to network nodes.

However, in order to support spreading of wireless community networks which rely on exploitation of wireless mesh networks, it is fundamental to provide a level of service availability and reliability comparable to the case of a wired network. Unfortunately, however, wireless links are characterized by additional sources of degradation like interference and fading and mobility of peer and router nodes makes the problem even more serious due to the need for dynamically updating the network status and resource location. Therefore, in order to cope with these impairments, opportunistic networking can be used. More specifically, a set of candidate wireless reference routers can be identified which takes care of performing resource search on behalf of mobile peer devices. This set can be dynamically changed according to where the peer node has moved and how the current status of the connecting link between a peer node and a wireless router changes in time. Also, network management and search procedures which allow to manage dynamic situations where peer nodes frequently join and leave the network should be considered. Another interesting feature which should be taken into account is related to the need for mapping the search procedure performed to look for available resources to the physical current location of network providing peers. This is useful in terms of increasing search efficiency in resource location reducing the distance between the overlay logical network and the physical one.

In this JRA we took into consideration two P2P mesh networking protocols, the Bamboo and the Georoy DHT, and performed a testing of their features coping with a dynamic and continuously changing environment. In the next sections we will illustrate the features of these two protocols and discuss the expected improvement that a combination of these two protocols can give.

### 5.2 Bamboo: handling churn over the Internet

Bamboo [31] is referred to as a third generation DHT, which improves previous systems by taking into account problems such as management traffic congestion in Pastry [32]. While Bamboo is based on the routing logics of Pastry, management of overlay structure is different in order to be more scalable in a dynamic environment.

To maintain the network structure, Bamboo keeps two sets of neighbor information in each node, leafset and routing table information. The leafset consists of successors and predecessors that are the numerically closest in the key space. While in the leafset two nodes may be neighbors in the overlay of a given node, they may be physically far away. When routing a query, it is ultimately forwarded to a node

which has the key in its leafset to ensure correct lookups. However, if only the leafset is used when doing lookups, a lookup complexity of  $O(\log(N))$  is achieved. To reduce the lookup performance, a routing table is used, which is populated with nodes that share a common prefix. Routing table lookups are then ordinary longest-prefix matches.

When data is stored in the DHT using the put command, the data is routed through the DHT to the node primarily responsible for storing the data. When the responsible node gets the data, it caches it within its leafset neighbors in each direction according to the number of desired replicas. For certain applications, the number of desired replicas can cause large demands for storage space (if each node has equal amounts of keys to store, every node needs to store that amount a number of times equal to the desired number of replicas). Therefore, for data storage updates, a node periodically picks a random node in its leafset and synchronizes the stored keys with it. The correspondent node calculates the set among its stored keys that should also be stored at the sender node, sending those keys to the sender, including the hash values of the data.

The major difference between Pastry and Bamboo is the way they handle management traffic. In Pastry, management is initiated when a network change is detected, while in Bamboo management traffic is periodic regardless of network status. While reactions to changes in the routing layer operate on very small timescale, reactions to changes in overlay structure are not so fast. However, the approach to use periodic updates has been shown to be beneficial during churn [31], since it does not cause management traffic bursts during congestion. Such traffic bursts can further increase packet loss probability; can lead to management messages being dropped and other overlay network disturbance.

In order for Bamboo to be able to serve requests and maintain a consistent network view among its nodes, it needs to perform overlay maintenance message exchanges between nodes. Periodic management traffic occurs in all layers of the Bamboo system. Neighbor ping is generated by every node in order to make sure that the node can still reach its one-hop neighbors in the overlay, and it is also used to maintain a RTT estimate used for retransmission timeout calculations. Such timer values are used to derive, if, for example, members left the overlay or messages need to be re-transmitted. However, re-transmitting too early will lead to too high number of packets. An accurate timeout value is crucial in order to predict if a packet is lost and needs to be resent along a different path in the overlay. Nodes also perform leafset update by periodically choosing a random node from its leafset, and execute a leafset push followed by a leafset pull. Both messages involve sending the complete leafset to the synchronizing node where the information is incorporated.

Bamboo considers that two nodes share the same level when one node contains the other node in its routing table. Therefore, the local routing table update is used to exchange the node information in that level. If a node gets information about other nodes that fit into the routing table, it probes the nodes to test reachability and to get a RTT estimate. If a node is reachable and fits into an empty field in the routing table, it is added. If the matching routing table entry is occupied, the node with the lowest latency is chosen. In standard configuration, Bamboo optimizes latency, although other schemes could be available, such as optimizing for uptime. It is important to note that an optimized routing table does not influence lookup correctness, but only lookup latency [33]. The Bamboo system has been evaluated through simulation and using testbeds such as the PlanetLab [34].

### 5.3 Georoy: a location-aware P2P algorithm

Although a careful construction of the overlay helps increasing the efficiency of P2P systems for wireless ad hoc networks, the combination of node mobility, lack of infrastructure and unreliable wireless links renders the traditional P2P proposals unsuitable for application in large ad hoc networks. Typically, it is assumed that a P2P network is formed by a few tenths of nodes (see [35–37]). Hence, the design of a *scalable* P2P system for wireless ad hoc networks is still an open problem.

The DHT algorithm discussed in this section is a variation of the Viceroy algorithm proposed in [38] based on the use of geographic information. For this reason, we call this algorithm *Georoy*. Another major difference between Viceroy and Georoy is that the former implements a flat peer-to-peer network

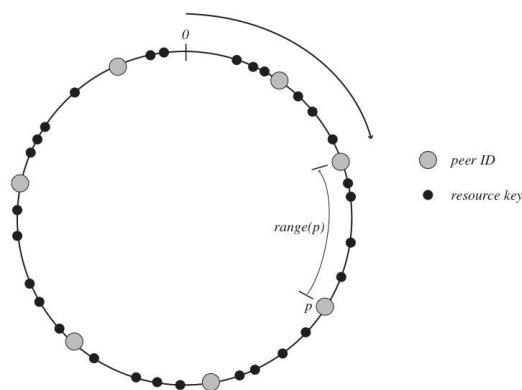


Figure 21: Viceroy's mapping of peer IDs and resource keys in the unit ring. Peer  $p$  manages all the resource keys in  $range(p)$ .

(i.e., peer nodes both provide content – resources – to the network and implement a distributed catalog for indexing), while the latter is based on a two-tier architecture: lower tier nodes (leaf peers) provide content, and upper tier nodes (super peers) implement a distributed catalog of available resources. In this context, leaf peers are mobile users, which make their resources available to the community, and super peers are the wireless routers (also called *wireless community nodes* in the following) composing the infrastructure of the wireless community mesh. The use of a DHT at the super peer level allows to implement a very efficient indexing of the available resources: every super peer maintains virtual links only to a constant number of neighbors (at most 7, independently of the system size), and queries about resource localization can be correctly answered by traversing at most  $O(\log n)$  links in the virtual overlay at the super peer level, where  $n$  is the number of super peers in the network. Furthermore, this design inherits from Viceroy important properties such as load balancing in the overlay network, and efficient maintenance of the distributed indexing mechanism in presence of super peer and leaf peer join/leaves.

### 5.3.1 The Viceroy algorithm

The algorithm used to implement the DHT abstraction at the super peer level is an adaptation of the Viceroy algorithm introduced in [38] to the wireless mesh network scenario. For this reason, we describe Viceroy before introducing our algorithm.

In Viceroy both peer IDs and resource keys are mapped by a hashing function into the same ID space (metric), i.e. the unit ring  $[0, 1]$ . Each key resides on the peer with the smallest ID larger than the key ID, i.e. the key range associated with peer  $p$  comprises all the resource keys with ID smaller than  $p$  and larger than the ID of the predecessor of  $p$  in the unit ring (see Figure 21). For simplicity, from now on we use notation  $p$  to denote both a generic peer and its ID in  $[0, 1]$ , and notation  $k$  to denote both a generic resource key  $k$  and its ID in  $[0, 1]$ . Note that two keys that are close in the ID space are located on peers which are also close in the same metric space.

Viceroy's overlay network uses both “short” and “long” range links between peers, which are established by combining the unit ring topology with an approximation of the butterfly network<sup>3</sup>. In order to emulate the butterfly network, each peer is assigned a certain level in the network, i.e. the identity of a node in the network is composed of the pair  $(p, l_p)$ , where  $p$  is the peer ID and  $l_p$  is the level of node  $p$  in the butterfly.

<sup>3</sup>The butterfly network on  $n$  nodes is a multi stage network with  $\log n$  stages, where a node at stage  $i$  is connected with a limited number of nodes at stages  $i - 1$  and  $i + 1$ . For a description of the butterfly network, see [39].

### 5.3.2 ID and level assignment

When joining the network for the first time, a SP  $p$  chooses uniformly, at random, a value in the  $[0, 1]$  interval, which represents its ID. Then,  $p$  randomly chooses its level  $l_p$  in the butterfly. Ideally, a peer should select its level by choosing uniformly at random a number in  $\{1, \dots, \log n\}$ , where  $n$  is the number of nodes currently forming the network. Since exactly computing  $n$  is virtually impossible in practice, the following procedure is used to compute an approximation of  $n$ . When joining the network, peer  $p$  first computes the distance  $d(p, succ(p))$  to its successor in the unit ring by invoking a LOOKUP( $p, p$ ) operation (see below for a description of the LOOKUP procedure); then, it estimates  $n$  as  $n_0 = 1/d(p, succ(p))$ , and selects a level by picking uniformly at random a number in  $\{1, \dots, \log n_0\}$ .

While the peer ID does not change during network lifetime, the peer level can change in order to maintain a balanced subdivision of nodes into stages when new peers join/leave the network. More in detail, a peer must recompute its level when its successor in the unit ring (and, consequently, its estimation of  $n$ ) changes.

### 5.3.3 Overlay construction

Viceroy's overlay network uses three types of directed links: *i*) *unit-ring links*, which connect a peer with its predecessor and its successor in the unit ring; *ii*) *level-ring links*, which are used to form a virtual bi-directional ring between the peers at the same level; and *butterfly links*, which are used to emulate a butterfly network. Butterfly links are composed of an *upward* and two *downward* links. The upward link connects peer  $p$  at level  $h > 1$  to the first  $(h - 1)$ -level peer after position  $p$  on the unit ring. The downward left link (the *short range* link) connects  $p$  to the first  $(h + 1)$ -level peer after position  $p$  on the unit ring; the downward right link (the *long range* link) connects  $p$  to the first  $(h + 1)$ -level peer after position  $p + 1/2^h$  on the unit ring. Summarizing, every peer in the Viceroy overlay network has at most 7 outgoing links: 2 unit-ring links, 2 level-ring links, and 3 butterfly links. The overlay topology is shown in Figure 22.

### 5.3.4 Routing

Routing in Viceroy is essentially performed by invoking a LOOKUP( $x, y$ ) function, where  $x$  is the requested key or peer ID (we recall that both resource keys and peer IDs are mapped in the same metric space), and  $y$  is the ID of the peer that invoked the function. The result of a LOOKUP( $x, y$ ) operation is the value associated with key  $x$ , or the ID of the peer that manages the key range to which  $x$  belongs (i.e. the peer with smaller ID larger than  $x$ ) if no value is associated with  $x$ . When the peer  $y$  at level  $l_y$  needs to retrieve key  $x$ , it initializes the current position to  $y$  and invokes the LOOKUP( $x, y$ ) function. The LOOKUP request is routed in the overlay, using the following three-phased process:

1. *up to the root*: starting from  $y$ , the request is recursively forwarded upward in the butterfly - each time updating the current position - using the upward link, until level 1 is reached;
2. *traverse the tree*: the request is forwarded downward in the butterfly, using either the short or the long range link depending on whether  $x$  is at distance smaller than  $1/2^h$  from the current position or not;
3. *traverse the ring*: finally, when the current peer has no downward links or it overshoots the target  $x$ , the request is forwarded using the level-ring and/or unit-ring links, until the peer  $s$  that manages the key range to which  $x$  belongs is found. Node  $s$  then returns the answer to the LOOKUP operation to  $y$ .

### 5.3.5 Overlay maintenance

A peer  $y$  in the overlay maintains the following information: (1) the ID on the unit ring (for simplicity,  $y$ ); (2) the current level  $l_y$ ; (3) the connections on the unit ring, *pred<sub>y</sub>* and *succ<sub>y</sub>*; (4) the connections on the

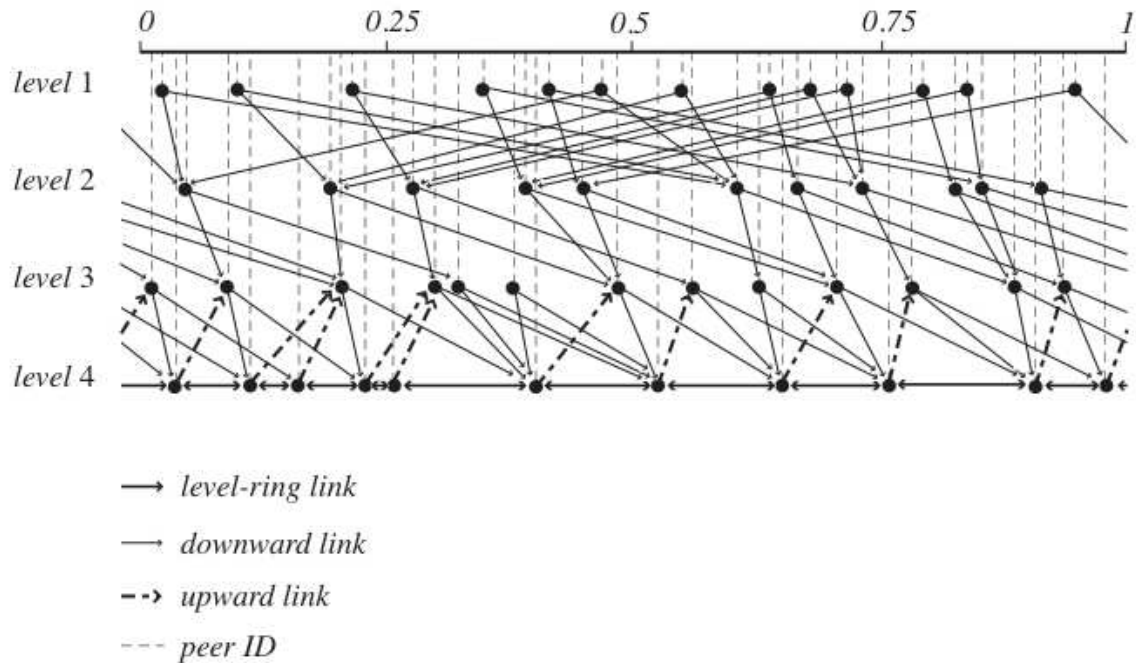


Figure 22: Viceroy's overlay network. For clarity, unit-ring links are not shown, and level-ring and upward links are shown only at level 4.

level ring,  $pred_y^l$  and  $succ_y^l$ ; (5) the upward butterfly connection,  $succ_y^{l-1}$ ; and (6) the downward butterfly connections,  $short_y^{l+1}$  and  $long_y^{l+1}$ .

When a new peer  $y$  joins the network, it first selects its ID as described above. By invoking  $LOOKUP(y, y)$ <sup>4</sup>, node  $y$  finds its successor  $succ_y$  in the ring, and establishes a connection to it. By exchanging information with  $succ_y$ , peer  $y$  knows the ID of its predecessor  $pred_y$  in the ring, and establishes a connection to it. Both  $succ_y$  and  $pred_y$  update their predecessor and successor link, respectively, in order to reconfigure the correct links in the unit ring. Then,  $succ_y$  transfers to  $y$  all the key-value pairs whose key is between  $pred_y$  and  $y$ . This completes the unit ring update. After that, peer  $y$  selects the current level  $l_y$  in the butterfly as described above. Then, it finds its successor  $succ_y^l$  and predecessor  $pred_y^l$  in the level ring by single-stepping on the unit ring, and establishes connections with them. The predecessor and successor links in the level ring of peers  $succ_y^l$  and  $pred_y^l$  are updated accordingly. Finally, node  $y$  establishes the butterfly links by finding  $succ_y^{l-1}$  and  $short_y^{l+1}$  (this can be done by single stepping on the unit ring), and  $long_y^{l+1}$  (this can be done by invoking  $LOOKUP(y + 1/2^{l_y}, y)$ , and then single stepping on the unit ring).

When peer  $y$  leaves the network, it has to remove all its established connections, notifying all neighbors in the overlay to update their links; then,  $y$  transfers its content to its successor in the unit ring.

When the current level changes (we recall that this is possible if  $succ_y$  changes), peer  $y$  has to update its level ring connections and butterfly connections, notifying the neighbors when necessary.

### 5.3.6 Viceroy's properties.

The following properties of Viceroy have been proved in [38], under the assumption that peer IDs and resource keys are distributed independently and uniformly at random in  $[0, 1]$ :

<sup>4</sup>Given the assumption of uniform ID distribution, the probability of two peers having the same ID is close to 0.

- *dilation*: If  $n$  peers are present in the network, then a LOOKUP operation is successfully completed by traversing at most  $O(\log n)$  links w.h.p.<sup>5</sup>
- *congestion*: Let the *load* of a peer be the probability that it is involved in a LOOKUP operation on a random value generated at a random starting point, and let the *congestion* of the network be the maximum of the peer loads. If  $n$  nodes are present in the network, the expected load for any peer is  $O(\log n/n)$ , and the congestion is  $O((\log^2 n)/n)$  w.h.p.
- *node degree*: If  $n$  peers are present in the network, then the out-degree of each node is at most 7, the expected in-degree is  $O(1)$ , and the largest in-degree of a peer is  $O(\log n)$  w.h.p.

### 5.3.7 Georoy

In this sub-section we show how to adapt Viceroy to a wireless mesh network scenario. First, we observe that, differently from Viceroy, in Georoy we assume a hierarchy of peers. Network members, which represent the lower tier of the hierarchy, are denoted as leaf peers (LPs) and provide content; the higher tier of the hierarchy is, instead, composed of super peers (SPs) which implement a distributed index of the available resources. Hence, in Georoy we assume that identities (i.e., a peer ID in the  $[0,1]$  interval, and a level in the butterfly) are assigned only to super peers. Also, a resource ID is assigned to each resource made available in the network; the resource ID is mapped into  $[0,1]$  by an appropriate hash function. In Georoy, the answer to a lookup operation on a certain resource  $k$  contains the location of the resource (e.g., the addresses of the leaf peer which holds  $k$  and its Home SP) in case  $k$  is available. We distinguish between two types of unavailability of the requested resource: *temporary unavailability* (i.e., the resource is available at some LP,  $u$ , in the network, but  $u$  is currently disconnected from the network), and *permanent unavailability* (i.e., none of the LPs has the requested resource). The answer to the lookup operation on  $k$  could be the address of  $p^{(H)}(u)$  in case of temporary unavailability, and the ID of the super peer that manages the key range to which the key associated with  $k$  belongs in case of permanent unavailability.

In what follows, we assume that peers<sup>6</sup> are distributed in a square deployment region of side  $s$ , for some constant  $s > 0$ , i.e. peers are located in<sup>7</sup>  $R = [0, s]^2$ . Furthermore, we assume that peers (i.e., wireless community nodes) are aware of their position in  $R$ .

Our goal is to define a mechanism to assign peer IDs that preserves geographical proximity, i.e. two peers which are geographically close should be assigned close IDs in the unit ring. Preserving proximity is fundamental to achieve a close correspondence between the virtual overlay and the physical network topology.

In the Georoy algorithm, peer IDs are computed as follows. Let  $(x, y)$  denote the coordinates of peer  $p$  in  $R$ . We define a mapping function  $\mathcal{M}$  that maps a point in  $R$  into  $[0, 1]$  as follows:

$$\mathcal{M}(x, y) = \begin{cases} \frac{x\Delta}{s^2} + \lfloor \frac{y}{\Delta} \rfloor \cdot \frac{\Delta}{s} & \text{if } \lfloor \frac{y}{\Delta} \rfloor \text{ is even} \\ \frac{(s-x)\Delta}{s^2} + \lfloor \frac{y}{\Delta} \rfloor \cdot \frac{\Delta}{s} & \text{if } \lfloor \frac{y}{\Delta} \rfloor \text{ is odd} \end{cases}, \quad (17)$$

where  $\Delta$  is an arbitrary constant with  $0 < \Delta < s$ .

The intuition behind our mapping function is depicted in Figure 23: the deployment region  $R$  is divided into  $s/\Delta$  sub-regions of equal area, which are defined in terms of the  $y$  coordinate. All the nodes in the same sub-region are mapped into the same segment of the unit ring, where the position of the node within the segment is determined by its  $x$  coordinate. In order to preserve proximity, the order of nodes in a segment is reversed alternately (see Figure 23).

<sup>5</sup>W.h.p. means with probability at least  $1 - c/n$ , for some constant  $c > 0$ .

<sup>6</sup>From now on in this Section, the term ‘peer’ is used instead of the term ‘super peer’.

<sup>7</sup>The use of a right open interval is needed to simplify the definition of the ID mapping function below, and it has no practical consequences.

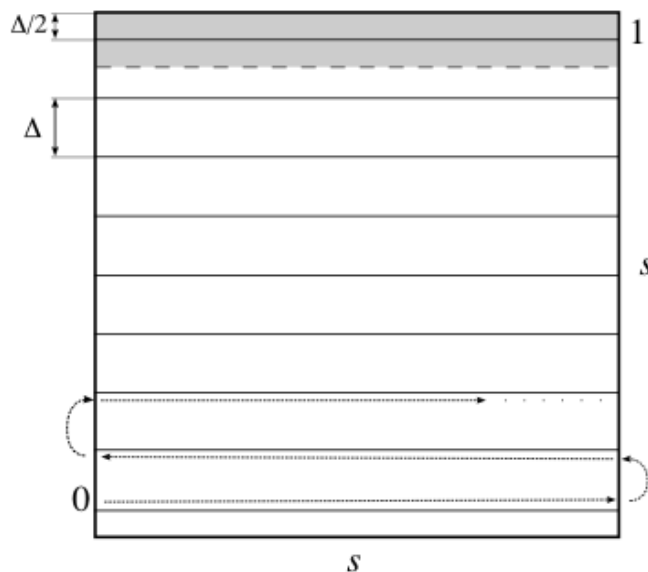


Figure 23: The mapping  $\mathcal{M} : R$  is divided into sub-regions (shaded area), and nodes in the same sub-region are mapped into the same segment of the unit ring. The order of nodes in a segment is reversed alternately to preserve proximity.

Before ending this subsection, we observe that the above defined mapping can be applied to any DHT approach which maps resource and node IDs in the  $[0,1]$  interval, such as Chord [40] and Koorde [41].

### 5.3.8 Mobility management procedures

In this section we describe the procedures required to integrate the Georoy DHT algorithm in a realistic system where any node can suddenly show up, move or disappear.

More in deep, the procedures being defined are:

- Node joining/leaving used when the generic LP  $u$  node periodically passes from the *active state* to the inactive one and viceversa.
- Resource insertion procedures required to update the distributed resource catalog when a LP decides to share a new resource, or not to share a certain resource anymore
- Information Retrieval procedures describing how the retrieval of resources is addressed
- peer node handoff from a wireless router (i.e. super peer) formerly responsible to another

More details on the definition of these procedures can be found in [42].

## 5.4 Ongoing activity and future work

The joint activity has been aimed at identifying strengths and weaknesses of the two above discussed protocols. More specifically, while the Bamboo does not require the definition of a two layer architecture and thus results simpler than Viceroy, the latter can give good improvement in terms of delivery delay due to the strict fitting between the logical and the physical network search. However, many aspects are currently under investigation like reliability in case of link variability and the scalability. To cope with

these aspects we are currently developing ns-2.28 code for the two algorithms to perform an exhaustive simulative analysis of the protocols in various network conditions and depending also on peer density.

In the next period we will complete the simulator development and perform an exhaustive simulation campaign to compare algorithms' performance. This could be used as the basis for the definition of a new P2P protocol effectively designed to behave in opportunistic mesh scenarios.

## 6 OPPORTUNISTIC CONNECTIVITY: THE IMPACT OF NODES MOBILITY

*CONTRIBUTORS: CNIT-BO, RWTH*

The present JRA involves researchers belonging to two institutions, namely CNIT-University of Bologna and RWTH Aachen University, and was originally introduced in June 2008. The aim was to introduce and analyze novel scenarios for wireless ad-hoc and sensor networks, with emphasis on considering more classical connectivity and MAC issues while jointly accounting for the opportunistic behavior of the network. After about one year of collaboration which included both remote meetings and a 3 months exchange of a PhD student from Bologna to Aachen, the activity has resulted in two joint papers, one accepted at ICT-Mobile Summit, 10-12 June 2009, Santander, Spain, and the other submitted to Globecom, 30 Nov - 4 Dec 2009, Honolulu, USA. In the following, an extensive report of the results obtained is given.

### 6.1 Research activities

Wireless sensor networks (WSNs) are a promising technology for environmental monitoring, offering potentially high spatial and temporal resolution using inexpensive networked embedded devices. Traditionally fixed gateways connected at all times to the sensor network have been used to gather measurement results and report them to the backend systems reachable through the Internet [43]. However, such an approach requires rather dense deployment of sensor nodes (or a large number of gateways) with commonly used radio technologies. This increases the cost of a WSN solution significantly, and creates remarkable routing overhead. A highly interesting alternative would be to use *opportunistic* or *delay tolerant* communication techniques, exploiting the mobility of nearby nodes (such as vehicles or pedestrians) to ferry the data to a harvesting point. Such an approach potentially enables much sparser WSN deployments, with density of nodes only dependent on the desired spatial resolution of the sensing task.

In this work we focus on the use of vehicles to carry measurement readings from a potentially disconnected collection of sensor nodes to gateways for further processing in an opportunistic manner. Especially vehicles used in the public transport sector would be ideal for such an application. Such a fleet of vehicles owned and administered by a single entity with highly predictable traffic patterns makes for a natural candidate in applying opportunistic communications for data harvesting in the real world. As further motivation for exploiting the mobility of buses, we highlight that: (i) high route predictability and periodic schedule make it possible to a-priori evaluate performance with higher accuracy; (ii) the power-law behavior of some metrics of interest such as inter-contact time (i.e., the time duration between two contiguous network contacts), pointed out as detrimental with respect to performance in recent literature (see Section 6.2), is mitigated by high periodicity of buses' movement: as a consequence, large performance deviations can be avoided.

Besides defining new metrics that are specific to our scenario and presenting an alternative point of view on the analysis of inter-contact time, we especially study the delay performance and packet delivery rate using both measured vehicular mobility traces from the Seattle bus network as well as synthetic vehicular mobility traces. This approach enables comparison of the effectiveness of using public transport only for harvesting against use of all the vehicles (latter having obvious deployment difficulties). We also compare the results against those obtained using classical mobility models for ad hoc networking, demonstrating that latter leads to significantly different conclusions. Finally, even making the comparison between real traces and synthetic models as fair as possible by proper tuning of input parameters (e.g., number of vehicles, vehicle speed, domain size, etc...), we conclude that the use of actual vehicular traces is critical for evaluating performance of such systems as the proposed one.

The rest of the present chapter is structured as follows. In Section 6.2 we give an overview of the related work in the literature. We then describe the specific scenario we study in detail together with the used simulation environment in Section 6.3, and give definitions of the metrics used for performance evaluation in Section 6.4. Then the results are given in Section 6.5 and the implications discussed at

length in Section 6.6. Finally, we conclude the chapter in Section 6.7.

## 6.2 Related works

The opportunistic networking concept was originally proposed as a solution for providing Internet access where regular coverage is not costworthy [44] and it has been recently attracting great attention. The idea of taking advantage of sporadic contacts between nodes (perhaps even belonging to different networks) has made it possible to enhance the capacity of wireless networks. However, since this increased capacity usually comes at the cost of an increased average delay, intense study has been devoted to the paradigm of Delay Tolerant Networks (DTNs) [45–47], where communication between node pairs are enabled even if there is no direct route connecting them.

While Fall [45] reports practical motivations for studying DTNs and few case studies (among which terrestrial mobile networks employing buses mobility), Jain *et al.* [46] address the routing issues raised in a multi-hop DTN where nodes are of the same type, share the same mobility characteristics and have constrained buffering capabilities. While we share with the previous cited works the motivations for considering the opportunistic/delay tolerant paradigm, our scenario foresees three different types of nodes interacting, namely, mobile nodes (vehicles), stationary sensors and a fixed fusion center.

As a key metric for understanding time-evolving connectivity in mobile networks, we consider the inter-contact time distribution. Several works (see, e.g., [48–50]) report the analysis of inter-contact time in social context, ranging from human mobility to vehicular traffic. Hui *et al.* [48] show through real life data how human inter-contact time is governed by power-law behavior and discusses its impact on opportunistic forwarding techniques. In [49], authors prove that the power-law behavior holds until a characteristic time after which an exponential cut-off takes over. Also, the close relation between distributions of inter-contact time of a node pair and return time of a single node to a specific location is clarified in [49].

Finally, while we focus on the aggregate statistics of inter-contact time, Conan *et al.* [50] discuss the relevance of both pairwise and aggregate inter-contact time analysis.

## 6.3 Scenario description and simulation setup

### 6.3.1 Scenario description

Our scenario consists of a domain as large as needed to accommodate an urban downtown area and its vicinity. In an arbitrary subdomain  $\mathcal{D}$ , we assume a number  $N_s$  of stationary sensor nodes are deployed on a regular grid in such a way that each one is few kilometers away from its neighbors. Every node, equipped with one or more sensors and a transceiver, is in charge of monitoring a specific region which competes to its covering pattern and of producing data samples (that we generically call “*data units*” hereafter) at a constant rate. No synchronization is assumed among different sensors. Each generated data unit shall be stored in an internal sensor node buffer.

Along with stationary sensor nodes, we consider the presence of  $N_v$  mobile nodes representing urban vehicular traffic, generically including both public transportation vehicles and private cars. Mobile nodes also have wireless communication as well as storing capabilities.

Here we employ a simple disc model for quickly generating connections among nodes. Such a model has been extensively used in ad hoc networks connectivity literature (see, e.g., [51]) despite of its limited realism. However, even though it does not account for obstructions such as those encountered in an urban scenario, our results are more sensitive to the mobility pattern of vehicles rather than channel model.

Hence, whenever a vehicle happens to be in the range of a sensor (i.e., at a distance less than the transmission range,  $TR$ ), they can start communicating. We assume that at the beginning of contact period between any vehicle-sensor pair all data units present in the sensor’s buffer are transferred to the vehicle’s memory, appended to the previous content. Right after this operation, sensor’s buffer is erased. In case of multiple contacts between a single sensor and several vehicles, the latter will download its data units in order to increase the chances that at least one will approach the destination.

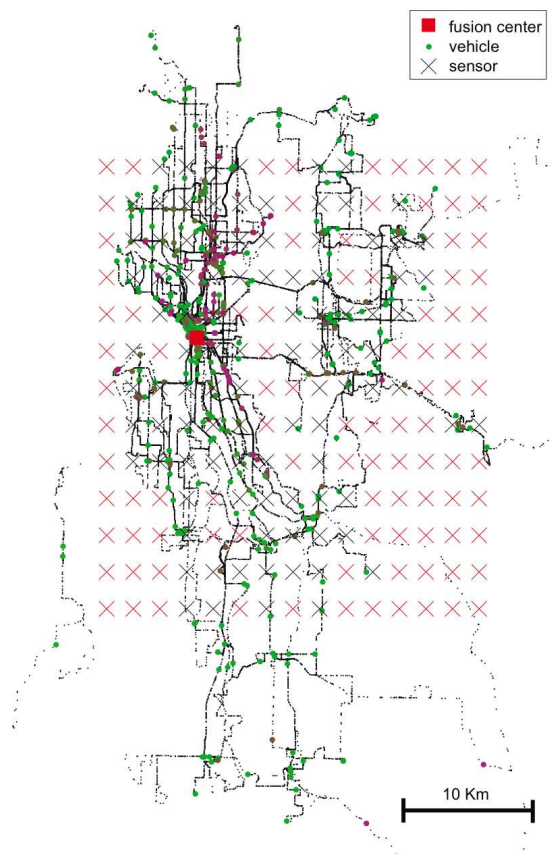


Figure 24: Scenario representation with mobile nodes (circles) belonging to the King's County (WA) public transportation fleet, sensors (crosses) placed on a regular grid and fusion center (red square) located in downtown Seattle.

Finally, we assume there exists a fusion center placed in a fixed location where all data units shall be eventually delivered for processing. To this end, every time a vehicle gets in touch with the fusion center it uploads all collected data units to it and frees its memory. At the collection point all duplicates (i.e., data units originated by the same sensor at the same time instant) are discarded. Hence, the life of a single data unit begins when it is generated by a sensor. Then, it is first stored into sensor's buffer until pick-up time, when it is transmitted to a vehicle and stored in the bus memory. It remains there until the vehicle reaches the fusion center, where it is eventually delivered. In this fashion, each data unit will be delivered after a random delay.

A graphic representation of this scenario may be found in Fig. 24 for the specific case where mobile nodes (circles) are buses belonging to the King's County (WA) public transportation fleet, sensors (crosses) are placed on a regular grid and the fusion center (red square) is in downtown Seattle.

### 6.3.2 Simulation setup

In order to test the described scenario, we designed a discrete-time simulator environment which takes as an input a mobility trace description from an external file. In principle, any kind of data may be used, such as real life mobility patterns, traces generated by a synthetic model and outputs of micro-mobility simulator environments. Since our aim is to emphasize the role of various types of mobility to network performance, we considered three different simulation setups, each one featuring a different mobility source.

*Real-life mobility traces.* We examined the mobility patterns of the public transportation of King County, WA. In particular the bus fleet traces of Seattle area are publicly available in [52]. The measurements report the position of each bus at random time intervals (with mean approximately equal to 1

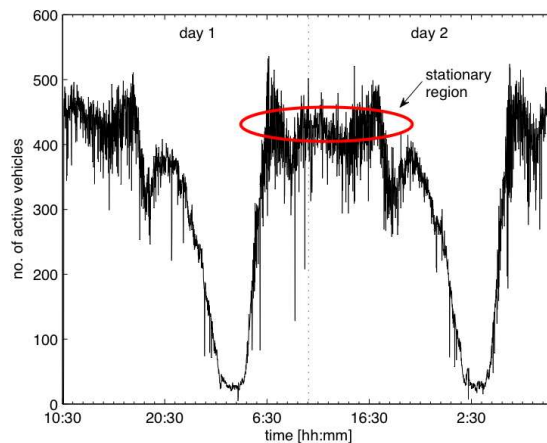


Figure 25: number of active buses as a function of time over a 48 hours period.

measurement per minute) over several days. More than a thousand vehicles are tracked, even though not all of them are active at the same time (i.e., buses may stop service, be still for several hours, etc...). We can formally state that a bus is active in a one minute time slot if it sends at least a position update in that slot. As an example, we report in Fig. 25 the number of active buses for a 2 days period. It is easy to note the time-of-the-day periodicity due to the alternation of early morning rushes and nightly service stops, which makes this kind of mobility highly predictable. For the sake of comparison to other mobility models, we highlighted a 10 hours time frame where the number of active vehicles can be considered stationary with mean 376, which ranges from 8:30 am to 6:30 pm. In the stationary region buses run at the average speed of 22 km/h.

*Random Waypoint.* This choice was suggested by the ease to generate mobility traces out of a synthetic mathematical model. Moreover, Random Waypoint is among the simplest, yet most studied classical mobility models. Its input parameters are tuned so as to resemble the real traces during the stationary time as much as possible. In particular, since there is no roadmap involved here, the domain area is chosen equal to the total area of viable roads of the King County region (i.e., we excluded lakes, sea, forests, etc...).

*VanetMobiSim.* VanetMobiSim [53] is a vehicular mobility simulator featuring realistic automotive motion models at both macroscopic and microscopic levels. At macroscopic level, it can import maps from the US Census Bureau TIGER database [54]. Also, it supports multi-lane roads, separate directional flows, differentiated speed constraints and traffic signs at intersections. At microscopic level, VanetMobiSim provides realistic car-to-car and car-to-infrastructure interaction. Its accuracy was tested against CORSIM [55]. Once again, we selected the King County roadmap and generated a random traffic flow with the same average number of vehicles as in the stationary period of real traces case.

## 6.4 Performance metrics

Here we introduce a selection of performance metrics related to the described scenario which are most critical to the application requirements. We define each one of them in a way that is specific to our network. We shall then discuss their relevance in the following sections.

### 6.4.1 Inter-contact time

Referring to a discretized time axis, we define *contact events* as those events when an arbitrary sensor-vehicle<sup>8</sup> pair  $p = (s_{ID}, v_{ID})$  are at a distance less than  $TR$  apart at time  $T_k$  and they were at distance greater

<sup>8</sup>Unlike recent literature, we study contact events between mobile and stationary nodes. However, our findings do not lose generality because of the close relation between the statistics of inter-contact time in mobile-to-mobile nodes contact analysis

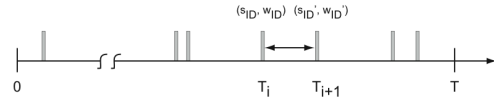


Figure 26: Representation of contact events as a marked random process, where each event is marked with the pair  $(s_{ID}, v_{ID})$ .

than  $TR$  at time  $T_{k-1}$ . We model the time distribution of contact events by means of the marked random process

$$\mathcal{C} = \{(T_i, m(T_i))\}, \quad T_1 \leq T_i \leq T_{N(T)}, \quad (18)$$

where  $T_i$  is the time of the  $i$ th contact event,

$$m(T_i) \in \{(s_{ID}, v_{ID}), 1 \leq s_{ID} \leq N_s, 1 \leq v_{ID} \leq N_v\} \quad (19)$$

is a marker,  $T$  is the simulation time and  $N(T)$  the overall number of contacts on  $[0, T]$  (see Fig. 26). This quite general definition is flexible enough to let us focus, for example, on the subprocess

$$\mathcal{C}_{h,w} = \{(T_i, (h, w))\}, \quad T_1^{(h,w)} \leq T_i \leq T_{N_{h,w}^{(h,w)}(T)}, \quad (20)$$

which models the contact events between sensor and vehicle whose IDs are  $h$  and  $w$ , respectively, by simply selecting the corresponding mark. Otherwise, we may consider the contact events between a specific sensor  $h$  and any vehicle by picking the subprocess

$$\mathcal{C}_{h,-} = \{(T_i, (h, -))\}, \quad T_1^{(h,-)} \leq T_i \leq T_{N_{h,-}^{(h,-)}(T)}, \quad (21)$$

etc... (note that  $\mathcal{C}_{-,-} \equiv \mathcal{C}$ ).

An inter-contact time between two contact events of the process  $\mathcal{C}_{h,w}$  is the duration of the time interval at whose two endpoints the vehicle  $w$  enters in the range of the sensor  $h$ . Instead, an inter-contact time between two contact events of the process, e.g.,  $\mathcal{C}_{h,-}$  is the duration of the time interval between two successive contacts of sensor  $h$  with some vehicles. In order for our measurements not to be biased on some specific sensor/vehicle, we consider the aggregate statistics of inter-contact time, i.e.,  $\forall h$  and/or  $\forall w$ .

Now we briefly discuss the importance of considering each one of the following cases and we compute the empirical aggregate Complementary Cumulative Distribution Function (CCDF) of inter-contact time:

1.  $\mathcal{C}_{-,-} \equiv \mathcal{C}$ . This process records the time instants of the contacts between any vehicle and any sensor. The CCDF is defined, for a given  $t, 0 \leq t \leq T$ , as

$$\hat{F}_i^{(-,-)}(t, T) = \frac{1}{N(T)} \sum_{n=1}^{N(T)-1} 1(T_{n+1} - T_n > t), \quad (22)$$

where  $1(x)$  is the indicator function which returns 1 if  $x$  holds and 0 otherwise. These inter-contact times are primarily related to overall data volumes produced and to the total protocol overhead.

2.  $\mathcal{C}_{h,-}, \forall h$ . The CCDF is defined as

$$\hat{F}_i^{(h,-)}(t, T) = \frac{\sum_{h=1}^{N_h} \sum_{n=1}^{N_{h,-}(T)-1} 1(T_{n+1}^{(h,-)} - T_n^{(h,-)} > t)}{\sum_{h=1}^{N_h} N_{h,-}(T)}. \quad (23)$$

Since this distribution is that of contact events of a given sensor, it can be used to reason about, for example, energy consumption rate of the sensor or the dimensioning of the buffers for storing sensor readings waiting to be transported.

---

and that of return time of a single mobile node to a specific location, pointed out in [49].

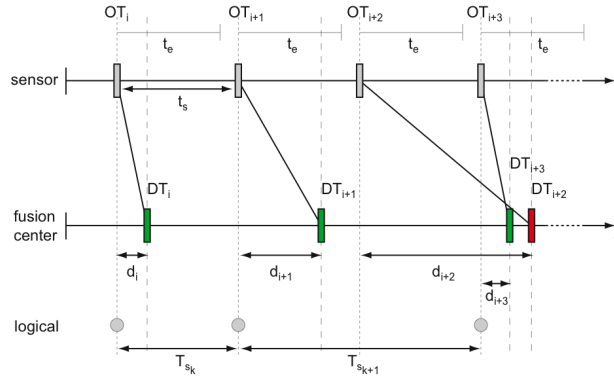


Figure 27: Sketch of generation and delivery of data units: each one of them is collected at the originating sensor and transmitted to the fusion center after a random delay. In time critical applications all data units with delay greater than some expiring time  $t_s$  are rejected by the fusion center.

3.  $\mathcal{C}_{-,w}, \forall w$ . The CCDF is defined as

$$\hat{F}_i^{(-,\cdot)}(t, T) = \frac{\sum_{w=1}^{N_w} \sum_{n=1}^{N_{-,w}(T)-1} \mathbf{1}(T_{n+1}^{(-,w)} - T_n^{(-,w)} > t)}{\sum_{w=1}^{N_w} N_{-,w}(T)}. \quad (24)$$

This is the counterpart of the previous definition, but this time for vehicles instead of sensors. Thus the corresponding distribution is related to the amount of data and communication activity of any given vehicle.

4.  $\mathcal{C}_{h,w}, \forall h, w$ . The CCDF is defined as

$$\hat{F}_i^{(\cdot,\cdot)}(t, T) = \frac{\sum_{h,w>h} \sum_{n=1}^{N_{h,w}(T)-1} \mathbf{1}(T_{n+1}^{(h,w)} - T_n^{(h,w)} > t)}{\sum_{h=1}^{N_s} \sum_{h<w \leq N_v} N_{h,w}(T)}. \quad (25)$$

The resulting distribution is that of contact times between a specific sensor and a specific vehicle. Thus it mainly relates to recurring behavior in mobility, and can be used to estimate the time a given fixed vehicle takes to make contact for a given sensor. This is also how inter-contact times are usually defined in the literature.

#### 6.4.2 Other performance metrics

Before introducing next performance metrics, we report a time sketch of data units collection mechanism in Figure 27. By considering an arbitrary sensor, a new data unit is generated every  $t = t_s$ . Thus, this time can also be regarded as the sampling time (or inter-sensing time) of the sensor. In particular, the  $i$ th data unit is originated at time  $OT_i$  and delivered at time  $DT_i$  with a delay  $d_i = DT_i - OT_i$  caused by (i) the time waited for being collected by a vehicle and (ii) the time spent traveling. Moreover, considering those monitoring applications imposing a maximum tolerable delay, we set an expiration time,  $t_e$ , such that if  $d_i > t_e$  the  $i$ th data unit is discarded by fusion center. However, it will be explicitly remarked when such a constraint is used.

*Overall delay.* With respect to sensor  $s$ , we denote by  $d_i^s = DT_i^s - OT_i^s$  the delay of the  $i$ th data unit generated by sensor  $s$ . Once again we collect the aggregate statistics by considering all data units originated by the entire set of sensors  $\mathcal{S}$ , as long as they successfully reach destination (i.e., with finite delay). If we now let  $N_r(T) = \sum_{s \in \mathcal{S}} N_r^s(T)$  be the total number of data units received by the fusion center on  $[0, T]$ , the aggregate CCDF of the delay is defined as

$$\hat{F}_d(t, T) = \frac{1}{N_r(T)} \sum_{s \in \mathcal{S}} \sum_{n=1}^{N_r^s(T)} \mathbf{1}(d_n^s > t). \quad (26)$$

*Sensor delivery ratio.* Each data unit stored in the fusion center buffer comes from a specific sensor node, and hence from a specific location. The sensor delivery ratio function,  $\hat{f}_r(s, T)$ , measures the normalized contribution of each sensor to the overall amount of data units at the fusion center. It can be written as

$$\hat{f}_r(s, T) = \frac{N_r^s(T)}{N_r(T)}. \quad (27)$$

*Average sensor throughput.* Once we have collected the delay values  $d_i^s$ ,  $1 \leq i \leq N_r^s(T)$ , relative to sensor  $s$ , we can come up with an estimate of the amount of data units that sensor  $s$  can deliver over time by taking the inverse of the average delay, i.e.,

$$\bar{R}(s, T) = \frac{1}{\mathbb{E}_i[d_i^s]} = \frac{1}{\bar{d}^s}. \quad (28)$$

*Monitoring efficiency.* Here we assume there is time-critical monitoring application running on the sensor network. For this reason, a minimum sampling frequency shall be guaranteed. Here we also set the constraint on the maximum tolerable delay,  $t_e$ , after which a data unit expires. Recall that sensors sample their nearby environment at evenly spaced time intervals, so their sampling frequency is  $f_s = 1/t_s$ . The fusion center must be able to reconstruct the underlying space-time process by means of data units received by the various sensors (i.e., locations). However, the generic data unit originated at sensor  $s$  will expire (i.e., will be discarded) if  $d_i^s > t_e$ . As a consequence, the fusion center cannot base its estimation on the complete set of uniform samples generated by sensor  $s$ , rather, it will have to deal with a reduced set of non evenly distributed samples. We call *feasible sampling frequency* of the sensor  $s$ ,  $f_s^{(F)}(s, t_e, T)$ , the average sampling frequency that the processing center has at its disposal for the location of  $s$  on  $[0, T]$ . More formally, we have

$$f_s^{(F)}(s, t_e, T) = \frac{\sum_{n=1}^{N_r^s(T)} 1(d_i^s < t_e)}{T}. \quad (29)$$

Now assume some target sampling frequency,  $f_s^*$ , is required by the application. We then define monitoring efficiency,  $\eta$ , as

$$\eta(f_s^*, t_e, T) = \frac{\sum_{s \in \mathcal{S}} 1\left(f_s^{(F)}(s, t_e, T) \geq f_s^*\right)}{\sum_{s \in \mathcal{S}} 1}, \quad (30)$$

that is, the fraction of sensors for which the network can actually sustain the target sampling frequency.

## 6.5 Numerical results

We shall now give a concise overview of the results obtained in terms of the mobility traces and performance metrics discussed above. More detailed discussion of the implications of these results is given in the following section.

The duration of simulation is in all cases  $T = 600$  min, while time precision is 1 second for Random Waypoint and VanetMobiSim simulations and 1 minute for real traces simulations. The subdomain  $\mathcal{D}$  where sensors are deployed has area  $A_{\mathcal{D}} = 1114$  Km<sup>2</sup>. As for their number, we set  $N_S = 180$ , they are deployed on a regular grid with vertical and horizontal spacing equal to 3.05 km and 2.18 km, respectively. Two different values are taken for transmission ranges, namely  $TR = 600, 1200$  m. With such a choice, the values of the average number of sensors in contact with an arbitrary vehicle crossing  $\mathcal{D}$  are  $N_{\text{con}} = 0.18, 0.73$ , respectively. Each sensor produces a data unit every minute. The location of fusion center is downtown Seattle for the roadmap case (i.e., real traces and VanetMobiSim traces) and simply the center of the scenario in the roadmap-less case (i.e., Random Waypoint traces).

The statistics that we collect are taken on the “cleaned” set of received data units, i.e., after discarding those marked as multiple copies.

In the analysis of those events lasting longer than the measurement period, we adopted the Kaplan-Meier Estimator together with the algorithm proposed in [56] for dealing with the censorship issue. In Figure 28 the main performance metrics are summarized. While Figures 28(a,b,c) address quite general

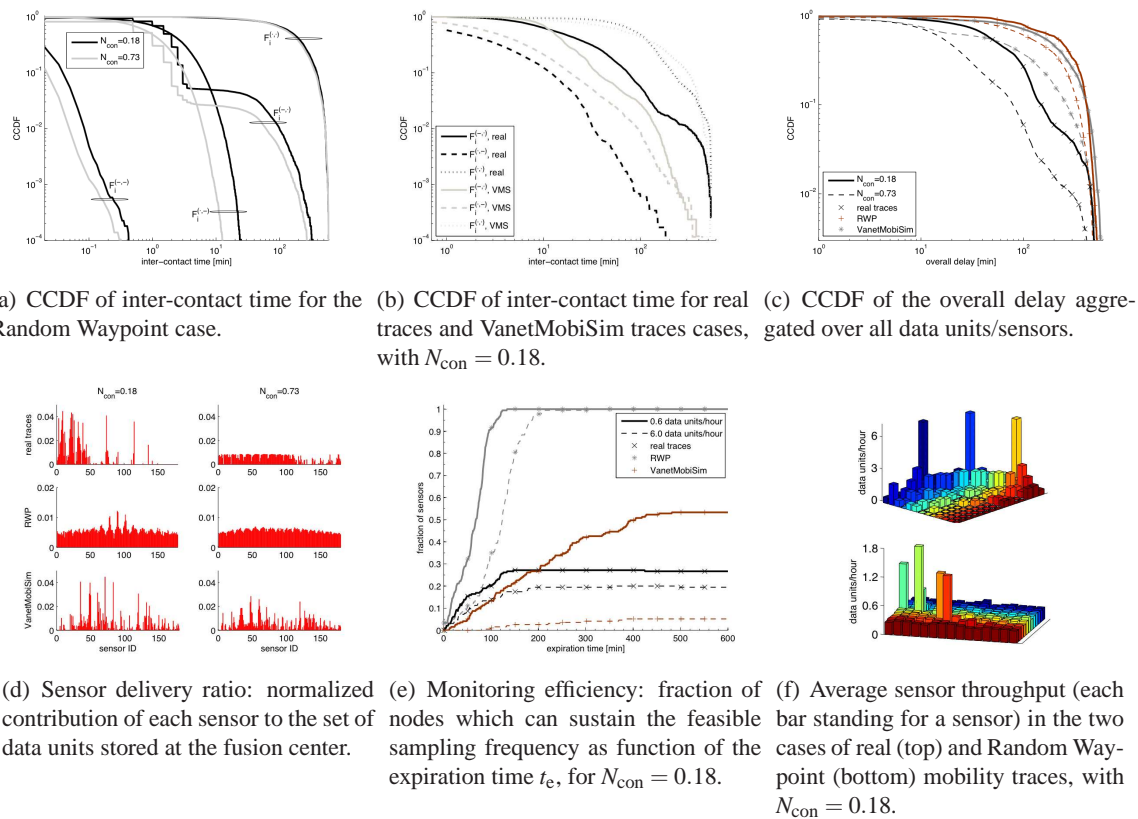


Figure 28: Collection of all performance metrics.

purpose metrics (i.e., they are not specific to a type of application), Figures 28(d,e,f) are more oriented towards an environmental monitoring application. The different inter-contact time distributions are given in Figures 28(a) (random waypoint) and 28(b) (vehicular mobility). There clearly is a significant difference between the distributions, aggregate ones exhibiting the commonly observed power-law behavior (see, for example, [56]) with more specific ones tending towards exponential behavior.

Figure 28(c) provides another view to the data, namely the CCDF of overall delay,  $\hat{F}_d(t, T)$ , further highlighting the differences induced by the mobility traces used.

In Figure 28(d) the delivery ratio,  $\hat{f}_r(s, T)$ , for different sensors are shown. As can be expected for random waypoint mobility the results are much more uniform than for vehicular mobility constrained by the road network. The performance for time-critical data in terms of the target sampling frequency  $f_S^{(F)}$  and the expiration time  $t_e$  is illustrated in Figure 28(e). Finally, in Figure 28(f) we show the achieved average sensor throughput for the sensor field for both the real mobility traces and for random waypoint mobility. We again see the significant difference in homogeneity over space, highlighting the different nature of the mobility patterns.

## 6.6 Discussion

We now aim at discussing how critical is the choice of a realistic mobility model with respect to the performance metrics introduced.

To begin with the more general purpose metrics (Figures 28(a) and 28(b)),  $\hat{F}_i^{(\cdot, \cdot)}$ , that is, the CCDF of inter-contact time of a specific sensor/vehicle pair seems to be the heaviest-tailed, while, as expected, the CCDF of inter-contact time between any vehicle with any sensor is the fastest decaying one. Note that  $\hat{F}_i^{(\cdot, \cdot)}$  (i.e., the CCDF of inter-contact time between a specific vehicle with any sensor) exhibits a “double waterfall” behavior: the first part of the curve refers to when a vehicle crosses the entire grid of sensors and gets in contact with each one at evenly spaced time instants, depending on its speed and trajectory;

the second part of the curve instead is dominated by rarer events, that is, when a vehicle's path touches only marginally the sensor field. Even though this may appear as a result of random waypoint mobility, the same behavior is observable for the real traces case (Figure 28(b)). While in regular mobile networks performance is mainly governed by  $\hat{F}_i^{(\cdot, \cdot)}$  (i.e., the CCDF of inter-contact time between a specific node pair), in our opportunistic scenario only sensor nodes are data sources whereas vehicles are just carriers. Hence, the primary parameter to look at is the interaction between a specific sensor and any vehicles, represented by  $\hat{F}_i^{(\cdot, \cdot)}$ .

Delay performance (Figure 28(c)) shows no substantial difference between the VanetMobiSim and random waypoint cases. However, they both give a too pessimistic view of delay of data units. The real traces curve highlights a faster decay as well as some improvement when increasing the transmission range. In fact, this helps reducing the time since data units generation to their collection. However, the traveling time, that is the dominating delay component, cannot be substantially decreased in this fashion.

As far as environmental monitoring applications are concerned, the plot of Figure 28(d) gives an overview of how homogeneously the target area can be monitored in the different configurations. It clearly appears how the simplistic random waypoint model leads to an overly optimistic result. When exploiting regular traffic instead, some sensors are better served than others because of their more accessible locations (e.g., close to main highways), resulting in a rather inhomogeneous sampling. The worst case is when we consider buses mobility: its regularity makes unreachable at any time those sensors not located on some bus route. In the latter case sensor deployment should be planned in a route-aware fashion. Increasing the transmission range may be to some extent beneficial, but may also result in buffering issues and energy waste.

From Figure 28(e) it appears once more how random waypoint mobility leads to an overestimation of performance: about 90% of sensors can get 0.6 data units/hour across with delay less than 100 min, while in the two other cases no more than 50% of sensors satisfy the same requirement even relaxing any delay constraint because of unreachability. Moreover, buses mobility tends to favor few "well-located" sensors, outperforming regular traffic for tight delay constraints. On the other hand, the more randomized nature of regular traffic makes it possible to collect data from a wider set of sensors provided that the application is more tolerant to delay.

The same can be observed in terms of per-sensor throughput from Figure 28(f) (notice the scale difference).

## 6.7 Conclusions and future work

In this chapter we have addressed the performance of an opportunistic network where vehicular mobility is exploited for collecting sensor readings to be delivered to a fusion center. Specifically, we examined the case where buses are employed as carriers and compared it to the one where regular car traffic is used. As further comparison, we also measured performance when the mobility pattern is generated out of a mathematical model.

We focused on several metrics both conventional and novel, which have been numerically evaluated and discussed. The outcome has highlighted how different mobility sources can lead to significantly different results (event if they are conveniently tuned), making the use of proper vehicular mobility critical to performance evaluation.

As future work, we intend to delve into the role of fusion center location as well as to take into consideration the limited buffering capabilities of sensor nodes.

## 7 HETEROGENEOUS AND OPPORTUNISTIC WIRELESS MESH NETWORKS

*CONTRIBUTORS: PUT, IST-TUL, KAU, CNIT-TO, ISMB*

Opportunistic networks require intermittent connectivity with fixed infrastructures to enable performing upload (e.g. of collected data – WSN) and download (e.g. access to the Internet) operations. Nodes forming an opportunistic network may use heterogeneous radio interfaces for both communicating with other nodes and access points.

One of the possible solutions in providing connectivity to opportunistic networks is recurring to Wireless Mesh Networks (WMNs), self-organised networks with self-configuring, self-optimising and self-healing characteristics that provide a stable (slowly varying over time) and robust topology.

In this section we describe two JRAs on opportunistic wireless mesh networks. The first one is the result of a collaboration among three institutions, namely PUT, IST-TUL and KAU, and focuses on the analysis of heterogeneous (802.11 and 802.16) multi-radio and multi-channel WMNs. It exploits concepts and strategies that may increase their performance, taking into consideration opportunistic aspects of WMNs as well as connectivity with opportunistic networks. Such multi-radio multi-channel WLAN can have significantly higher capacity than single channel WLAN meshes.

The second one, is the result of a collaboration between CNIT-TO and ISMB. It has the objective of investigating the performance of several routing protocols for ad hoc and mesh networks in a scenario where mobile nodes move along a constrained path and opportunistically connect to infrastructure mesh points.

### 7.1 Capacity analysis of opportunistic WMNs using collision domains and multi-radio WMNs

*CONTRIBUTORS: PUT, IST-TUL, KAU*

#### 7.1.1 Collision domains in 802.16 WMN

The concept of collision domains was applied to WMNs capacity calculation for the first time in [57]. The method is based on the fact that the existence of gateways in WMNs introduces hot spots in the network that act as bottlenecks. Identifying the bottleneck collision domains allows computing exactly the minimum and maximum data rates available for each node for a given network topology and link layer protocol. The concept was further developed by Aoun and Boutaba in [58], by considering fairness to ensure proper operation of WMNs.

In our research we apply the collision domains concept to the 802.16 based WMNs, as specified in the IEEE 802.16 standard [59]. Therefore, we need to analyze the MAC protocol carefully. Following [57] we define the collision domain of the  $i$ -th link as a set of links formed by the  $i$ -th link and all other links that have to be inactive for the  $i$ -th link to transmit successfully. For clarity, let us consider a simple chain of  $n = 8$  nodes receiving and forwarding traffic from the gateway to the other nodes. We define the collision domain for link  $k = 3$  (between nodes 3 and 4) of the 802.16 based WMN as presented in Fig. 2 in [60].

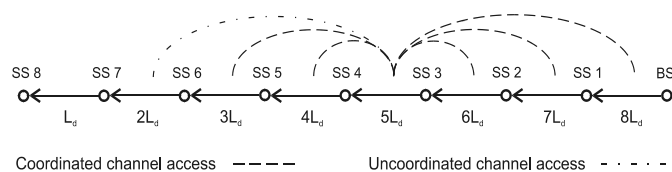


Figure 29: Simple chain topology of 802.16 WMN with  $n=8$  nodes.

This definition of the collision domain allow us to compute the traffic to be forwarded within the collision domain. Assuming that each node in the chain downloads from the gateway the traffic of  $L_d$  [b/s] (this assumption is made for clarity only and the method can be easily extended to an arbitrary load of each node), a link which is closer to the gateway has to carry more traffic, e.g. link 0 (gateway -> node 1) has to carry the load of  $8L_d$  [b/s] while link 5 has to carry  $3L_d$  [b/s]. Since each collision domain has to be able to forward the total load of its links, the collision domain for link 3 defined in the previous example forwards  $35L_d$  [b/s] (nominal load). If the bandwidth (the capacity of the MAC layer) for each link in the collision domain is constant and equal to  $B$ , the throughput  $L_d$  available to each node in the chain is limited to  $L_d < B/35$  (for the considered collision domain). In order to find the collision domain which is limiting for the network (so called bottleneck collision domain) we have to identify the collision domain and its load for every link in the network and find the minimum throughput  $L_d$  [57]. However, due to spatial separation of links in the collision domain simultaneous transmissions are possible and should be deducted from the total load of the collision domain. The effective load of the collision domain #3 is now equal to  $26L_d$  [b/s] and the throughput available to each node in the chain is limited to  $L_d < B/26$ .

### **7.1.1.1 Numerical results**

A MATLAB® script was used to evaluate the method and we considered a chain topology similar to that presented above. The script identifies the collision domain for every link and calculates the available throughput. If all nodes in the chain download the same traffic from the gateway the collision domain #3 (node 3 -> node 4) is the most congested and forms a bottleneck for  $n \geq 3$ . If the traffic is unidirectional (downlink or uplink) the throughput decreases to 0.1 (per node) for  $n = 4$  and becomes as low as 0.03 for  $n = 10$  (see Fig. 30.a). For bidirectional asymmetric traffic with  $L_u = 0.1 L_d$  (a value typical for ADSL links) the same throughput of 0.03 is reached for  $n = 9$ . If the traffic is symmetric ( $L_u = L_d$ ), a similar throughput is obtained for  $n = 6$ .

802.16 OFDM PHY layer used in the mesh mode features an adaptive coding and modulation scheme [59]. Using a modified version of the method presented earlier we can easily show the impact of the available bandwidth, which can vary from link to link, on the overall performance of the mesh network. Doubling the bandwidth on the links 0 to 4 increases the throughput almost linearly (Fig. 30.b), however, modifying subsequent links does not increase it any more. Similarly, reducing the bandwidth on the links 0 to 3 by two decreases the throughput and it does not change when the other links are modified. More important is the fact that the substantial change of the bandwidth impacts the location of the bottleneck collision domain. This can be observed by increasing the bandwidth of some links to  $4B$ . Therefore we can conclude that the performance of the 802.16 WMN depends heavily on the bandwidth available on the links closer to the gateway, while the bandwidth of the other links is less important.

### ***7.1.2 Multi-radio WMNs***

One of the research topics under development deals with the optimal management of radio resources in multi-radio WMNs, in particular in the load aware optimisation of channel assignment and scheduling. We propose a unified radio agnostic approach for the efficient formation and maintenance of connectivity of flows between mesh nodes which takes into consideration the particularities of WMNs. It is a link layer protocol, described in [61], which presents the abstraction of a single channel to higher layers, and supports multiple radio-interfaces and radio-channels. It is a distributed strategy and does not require synchronization between nodes. Any type of routing (single- or multi-path) is supported. It enables the efficient management of the available radio resources of MPs (radio-channels, carriers and transmission power) and makes the WMNs flexible, by the opportunistic exploitation of node resources, with the objective of increasing network connectivity and reducing interference as much as possible.

The approach exploits a set of existing orthogonal channels and the nodes' capability of supporting multiple radio-channels. As an example, consider a mesh node with 2 radios, Figure 31, which operates

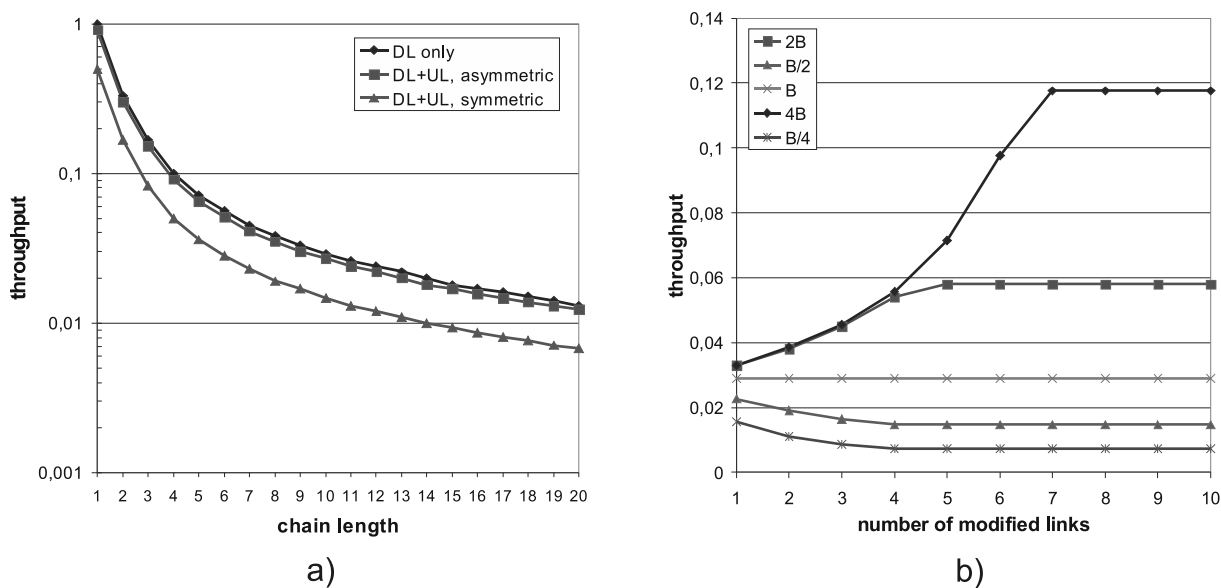


Figure 30: a) Throughput vs. number of active nodes, b) Effect of adaptive coding and modulation – throughput vs. number of affected links in the chain.

simultaneously on 2 different channels/carriers. One of the radios is considered a stable radio, having allocated a carrier for a large amount of time. The other radio is so called dynamic, because it dynamically switches between any of the remaining channels. Each node advertises its stable-channel as its receiving channel. Instead, to forward data to a neighbour node, the dynamic-radio is switched to the stable-channel of the neighbour node. This approach addresses an important trade-off between maximising connectivity and minimising interference. On one hand, connectivity is guaranteed with the dynamic-radio which allows the communication with any neighbouring node; on the other hand, the allocation of stable-channels can be strategically done in order to minimise the interference.

As a future work, we will study the following aspects. Regarding the selection of stable-channels, we will take into consideration the particularity of the received and forwarded traffic, the number of neighbour nodes which forward the traffic, the proximity to the gateway nodes, and so on. In particular, an algorithm, based on a collision domain concept, is under development.

Moreover, the estimation of theoretical bounds of capacity of WMNs is under study. This will be done under the assumption of absolute fairness, where is supposed that mesh nodes have equal offered load.

Finally, a set of simulations using the OPNET Modeler platform and an implementation in the KAU mesh testbed have been planned.

### 7.1.2.1 Channel scheduling in multi-radio WMNs

#### Channel scheduling strategies

Multi-radio multi-channel wireless mesh networks utilize multiple radios and channels simultaneously. In non-static multi-radio/multi-channel wireless mesh networks architectures such as Net-X [62] mesh nodes need to switch channels in order to communicate with different neighbors. According to [63] the channel switching time on current hardware is between  $200 \mu s$  and  $20 ms$ , which causes a very high overhead for per-packet switched channels. Therefore, multi-radio multi-channel mesh network platforms such as Net-X use per-channel queues, which are serviced for a longer time span. A scheduler then has to decide when and for how long a channel is serviced.

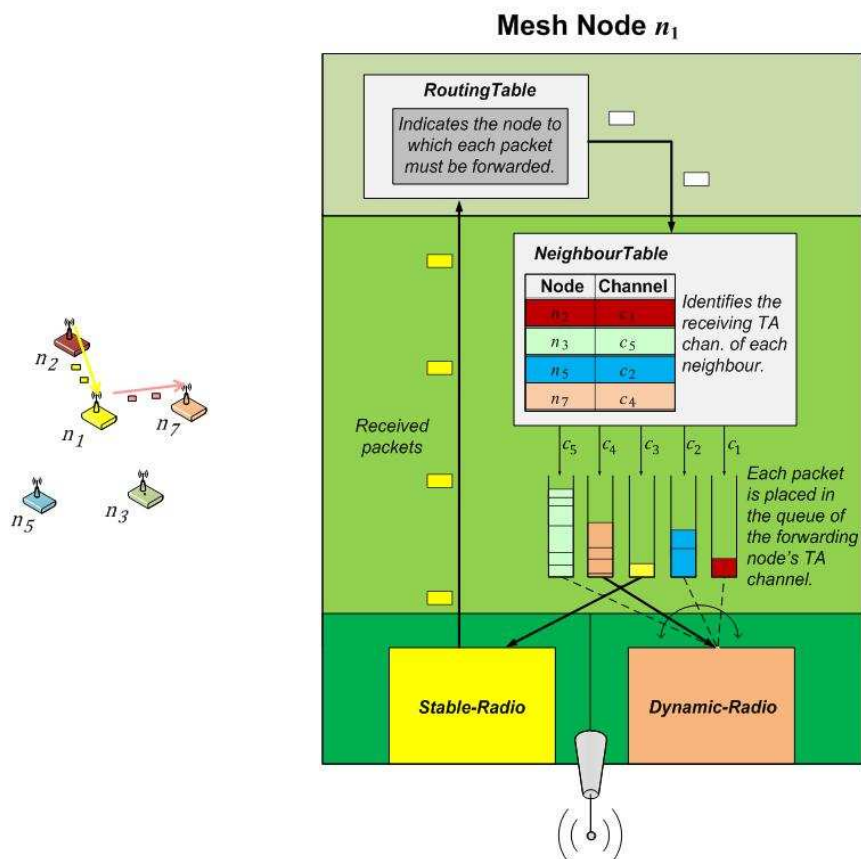


Figure 31: Example of operation of a mesh node equipped with two radios.

Different channel scheduling strategies are possible. The most simple one, which is implemented in Net-X, is Round Robin scheduling. Hereby the channels which have packets to send are serviced in a Round Robin fashion. Another option would be to make the channel scheduler load-aware. Using this strategy channels with an higher average queue-length are serviced more often. However, none of the two strategies takes explicitly into account the requirements of real-time traffic such as VoIP. The latency and the jitter of VoIP traffic should be kept low in order to provide a satisfactory end-user quality. Thus we propose a QoS-aware channel scheduler which explicitly takes into account the requirements of VoIP traffic. The goals of our scheduler are thus to minimize the delay and waiting time for delay sensitive traffic while at the same time providing reasonable throughput for delay-insensitive traffic at reasonable switching cost. It does so by producing a channel switch pattern which reduces jitter and gives enough channel time to the background traffic. We have implemented and evaluated the algorithm in the KAUMesh testbed. Preliminary results show that with our algorithm we can reduce the delay and jitter of VoIP flows while still providing good performance for background TCP-traffic.

### Implications on capacity

In mesh networks, due to the limited number of available channels, channel interference can not be completely eliminated. Therefore, an important problem to consider is the channel allocation in Multi-Channel Multi-Interface Meshes. Our objective is to mitigate the effect of channel interference so that we could maximize the end-to-end throughput of multihop wireless network communication. As channel assignment is an inherently NP-hard problem, we would like to evaluate several simple distributed channel

scheduling algorithms such as random or Round Robin or an algorithm that takes QoS requirements into account.

In this section, we consider wireless mesh networks with static channel assignment where every node has  $N$  half duplex wireless interfaces, which are assigned to the same set of  $N$  orthogonal channels. For direct communication, two nodes need to be within communication range of each other. All the interfaces of nodes have the same transmission range  $T$  and carrier sensing range  $C$ . Our MCMI network has the same topology as single channel single interface network and we assume that a node can communicate with its neighbors using all the available interfaces simultaneously. As an example, node  $v$  can receive data packets from node  $u$  on channel  $Ch_0$ , then node  $v$  relays those packets to node  $w$  on channel  $Ch_1$ . If we assume that the channels are orthogonal, node  $v$  also may send packets to node  $w$  on channel  $Ch_0$  and  $Ch_1$  concurrently. We denote  $uv(c_i)$  as communication link on channel  $c_i$  between node  $u$  and node  $v$ . In MCMI networks two links  $u_1v_1(c_1)$  and  $u_2v_2(c_2)$  can interfere with each other if the following conditions apply:

$$c_1 == c_2 \text{ AND } (d(u_1, u_2) \leq C \text{ OR } d(u_1, v_2) \leq C \text{ OR } d(v_1, u_2) \leq C \text{ OR } d(v_1, v_2) \leq C)$$

A simple strategy can be the following: if a node  $v$  detects that channel  $Ch_0$  is busy, other idle channels such as channel  $Ch_1$  could be used for data transmission.

With multiple channels and multiple interfaces in place, we can minimize the channel interference along a multihop path by minimizing the number of communication links using the same channel within carrier sensing range of each other. However, the number of available channels are limited, thus the challenge is scheduling a channel for a communication link from the set of limited channels to minimize the channel interference. As in our model all the interfaces are equal, all share the same routing table. At the source nodes, packets are generated and can be sent out on a randomly selected channel. When an intermediate node along the path receives a data packet on an incoming channel  $Ch_i$ , it needs to select the outgoing channel  $Ch_o$  to forward the packet. We will now briefly look into four channel scheduling algorithms for selecting the outgoing channel at the intermediate nodes along the path for communication in MCMI networks: *Random*, *Round Robin* and *Round Robin+*. For those scheduling approaches, we will quantify the capacity in a chain topology using the concept of collision domains.

In the *Random* Channel Scheduling algorithm, the node which receives the packet on  $Ch_i$  forwards the packet to the next hop on a random channel  $RAN(Ch_i)$ . This algorithm tries to utilize all available channels by giving them equal probability to be selected for forwarding data packets. Due to absence of coordination, transmissions within vicinity of each other can be assigned the same channel with some probability, leading to potential high interference.

The *Round Robin* Channel Scheduling algorithm tries to minimize the intraflow channel interference so that channels are evenly distributed along the flow. Here, the node which receives the packet on  $Ch_i$  forwards the packet to the next node on channel  $Ch_{(i+1) \bmod N}$ . As a result, the Round Robin algorithm ensures that two communication links of a path which use the same channel will be  $(N + 1)$  hops away from each other. As a result, the Round Robin channel scheduling algorithm can effectively minimize the channel interference along a multihop path. However, a node having multiple communication flows within its vicinity, might encounter much more channel interference than other nodes. As a result, some channels might be congested as many packets might arrive at that node on the same channel causing collisions. In addition, all those packets will be forwarded on the same next channel, even further increasing the congestion.

The *Round Robin+* Channel Scheduling algorithm tries to improve the Round Robin scheduler by assessing the queue size of the outgoing channel in the scheduling process. If the channel selected for forwarding packets has a non-empty queue, we can infer that the load on the forwarding channel is high (otherwise the channel queue would be empty or contain just the current packet). In that case, we could use other available channels for forwarding packets, i.e. we can select the channel which has the smallest amount of packets in the queue.

As a summary, the random channel scheduling algorithm assigns channel randomly, which does not systematically try to resolve the intraflow channel interference nor the interflow channel interference. However, due to the random assignment of channels it does reduce the channel interference. Round

Round Robin channel scheduling algorithm rotates channel usage along the path; its main objective is to resolve the intraflow interference. If we assume the average hop length of the multihop path to be  $L$ , if  $N > L$  then Round Robin channel scheduling algorithm can eliminate the intraflow channel interference. For longer paths  $L > N$ , Round Robin algorithm can eliminate the intraflow interference if  $N \geq (M + 2)$ . Assuming  $C = 2T$ , with 6 channels available, the Round Robin algorithm can eliminate the intraflow interference. However, Round Robin algorithm does not consider the interflow channel interference. Thus, in dense random networks, where the interflow channel interference occurs, Round Robin algorithm may not perform so well. Round Robin+ algorithm is as good as Round Robin algorithm in resolving the intraflow interference. In addition to Round Robin algorithm, Round Robin+ algorithm indirectly considers the interflow channel interference by choosing the channel with least number of packets in the queue to forward the data packets, which corresponds with the channel experiencing the least interference locally. With the ability to reduce the interflow channel interference, Round Robin+ algorithm has the best performance among the proposed algorithms. More details on the achievable capacity in a linear chain topology can be found in [64].

### 7.1.3 Future work

The simple chain topology and the unrealistic interference model considered in the capacity analysis of opportunistic WMNs does not allow for any generalization of the results. We plan to adapt the method to the arbitrary topology of the 802.16 WMN. Since the collision domains are defined based on the interference among links, this can be done only after implementation of the more realistic propagation models in our collision domain identification algorithm. The other extensions, like multi-channel and multi-radio terminals will be included as well. Finally, the mesh mode is very likely to be replaced by the relay solution in the new version of the 802.16 standard. Therefore, we will consider the application of the collision domain concept to the new mode as well as to the forthcoming IEEE 802.11s standard. The integration and evaluation of the channel assignment strategies developed in IST-TUL into the KAUMesh testbed is planned as future work.

## 7.2 Wireless Mesh Networks with Mobility Support

*CONTRIBUTORS: CNIT-TO, ISMB*

Wireless mesh networks are often built by many fixed nodes and few mobile nodes: they are robust systems that can be conveniently integrated with the existing infrastructure and offer high bit-rate services. *Mesh networks with mobility support* do not represent a new kind of network, but can be seen as an important *trade-off* between static wireless networks and traditional ad-hoc networks; the terms *with mobility support* are used to emphasize the fact that mobility in the network is limited to only a few nodes [65]. In this case the routing protocol efficiency, its responsiveness, and its capability to quickly identify loop-free routes are all fundamental qualities to guarantee the connectivity among nodes and provide the required performance.

We therefore investigate the performance of several routing protocols for ad hoc and mesh networks in a very specific setting: one or more mobile nodes (e.g., cars, buses or streetcars) that move along a constrained path and opportunistically connect to infrastructure mesh points (hereinafter called roadside mesh points). In such a scenario, uninterrupted connectivity is nominally guaranteed, but a rapidly-reacting routing protocol is needed to handle sudden link quality drops due to mobility, channel fading or shrinking capacity due to more than one node travelling on the same section of the path. We are therefore interested in verifying which routing protocols are capable of providing sustained end-to-end throughput as well as low latency/jitter (e.g., suitable for multimedia streaming to/from the mobile node), as the the vehicle(s) move along the path and opportunistically connect to the best point of access.

### 7.2.1 Reference Scenario

Our reference scenario is shown in Figure 32. In particular, we assume that a vehicle (node 8) equipped with transmission devices travel along a 1-km road stretch, as in the case of buses, streetcars, low-speed leisure trains, etc. Our objective is to devise a solution to cope with handovers between different roadside mesh points. The vehicle travels at a constant speed of 30 km/h. A continuous UDP stream (either in uplink or in downlink) is arranged between the vehicle and the roadside mesh point 0.

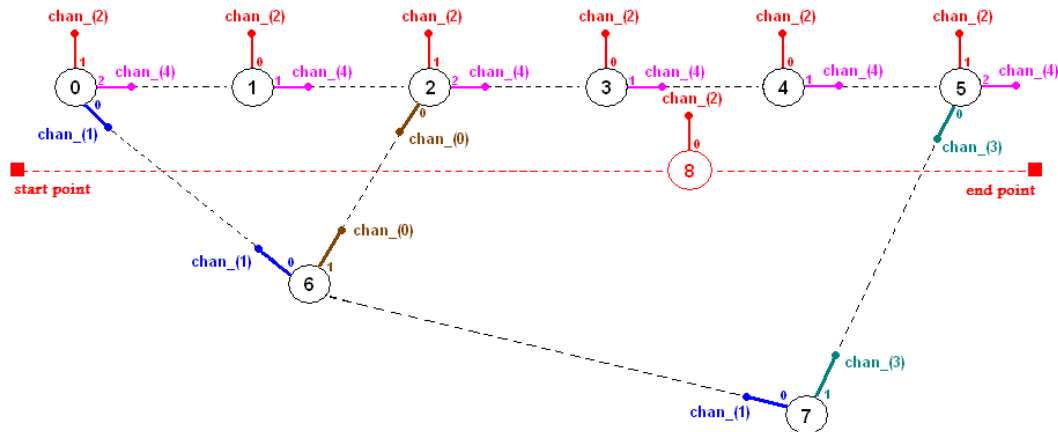


Figure 32: Reference network scenario.

All nodes use the IEEE 802.11 technology; the data rate on the links between the vehicle and the roadside nodes is 1 Mbps, while it is set to 11 Mbps on the links between roadside mesh points. Adjacent roadside mesh points use different frequency channels.

### 7.2.2 Comparing different routing protocols

In order to identify a good candidate for traffic routing in WMNs with mobility support, we compare four schemes, each of which represents a different approach to routing: (i) AODV [66], a reactive protocol which inspired the Hybrid Wireless Mesh Protocol specified in the IEEE 802.11s Draft Standard [67], (ii) OLSR [68] and OLSR-ETX, well-known proactive routing protocols, (iii) GPSR [69], a geographical routing protocol, and (iv) BATMAN [70], which has been recently proposed to handle traffic routing in multi-hop mesh networks.

Below, we briefly recall the main features of the routing protocols under study. Then, we identify a protocol in the BATMAN behavior and we change the scheme in order to solve it. Throughout this section, we present some simulation results, which allows us to assess the performance of the various schemes in opportunistic mesh networks.

**AODV** is a well-known protocol, which builds routes on demand by flooding the network with route request messages. As a route request hits the intended destination or a node that is aware of a route toward the destination, the request is not further forwarded and a reply message is sent back to the source. While traveling through the network, route request and reply messages create, respectively, reverse paths (pointing to the source) and forward paths (pointing to the destination). Sequence numbers and route error messages are used to handle link failures and avoid loops.

**OLSR** adopts an optimized *link-state* scheme to diffuse topology information while keeping the message overhead low. The key idea is that link-state information is generated and flooded in the network only by selected nodes, called Multi Point Relays (MPR). Any source-destination route is bidirectional and includes only MPRs as relay nodes. Furthermore, an MPR node may chose to report only links between itself and the node which selected it as next hop. Hence, unlike the classic link-state algorithm, partial link state information is distributed in the network.

OLSR can be extended to include the Expected Transmission Count (ETX) metric [71], so that routes with higher packet delivery ratio are selected. ETX of a link is indeed computed as the expected number of transmissions required for successfully transmitting a packet over that link.

**GPSR** exploits the correspondence between geographic position and connectivity in a wireless network, using the positions of the nodes to make packet forwarding decisions. It uses a *greedy* approach to forward packets to nodes that are progressively closer to the destination. In regions of the network where a greedy path does not exist, because the only path requires that packets move temporarily farther away from the destination, GPSR forwards traffic in *perimeter* mode. In this case, a packet traverses successively closer faces of a planar subgraph of the full radio network connectivity graph until reaching a node closer to the destination where the greedy forwarding resumes.

**BATMAN** is a proactive routing protocol, which has been recently submitted to the IETF. The strategy of this scheme is to determine for each destination in the mesh network one single-hop neighbor that can be used as best next hop to communicate with the destination node. To learn about the best next-hop for each destination, all nodes periodically broadcast originator messages (OGMs) to their neighbors; each OGM contains an *originator address*, a *sending node address* and a *unique sequence number*. When a neighbor node receives an OGM, it changes the sending address to its own address and rebroadcast the message in the network according to specific forwarding rules. In order to find the best route towards a certain destination, BATMAN counts the number of OGMs originated by that node and received within a sliding window. Then, BATMAN selects as next hop the neighbor from which it has received the highest number of OGMs within the sliding window (packet count metric), i.e., the link toward the destination with the best quality level. In this way a node does not maintain the full route to a destination but every node along the path only knows information about the next-hop to use.

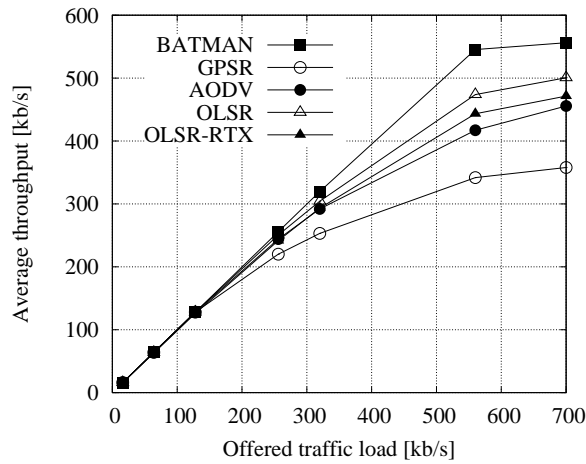
We compare the above protocols by using ns2 simulations, in the network scenario described above. Both the cases of uplink (from the vehicle to node 0) and downlink (from node 0 to the vehicle) traffic are considered. In all considered protocols, the control messages used to assess the connectivity with neighboring nodes (i.e., Hello messages, in AODV and OLSR, Beacon messages in GPSR, and of OGMs in BATMAN) are periodically transmitted with interval time equal to 1 s.

Figure 33 presents the average throughput obtained at the application layer as the bit rate of the traffic varies, in uplink (left plot) and downlink (right plot). We notice that BATMAN outperforms all other schemes when uplink traffic is considered, while it provides the worst results when downlink traffic is supported.

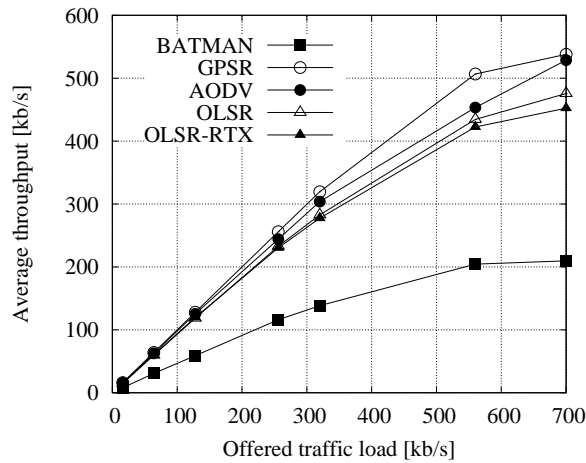
To understand the contradictory results obtained through BATMAN, we investigated the protocol behavior and identified the cause of the performance decrease in download mode.

Recall that BATMAN counts the number of OGMs within a sliding window to choose the *best next-hop* node towards a given destination. In the sliding window, all OGMs have the same weight and thus older and newer messages have all the same importance. We have found that in mesh networks with mobility support, such window management may cause temporary routing loops and, thus, packet dropping and high delivery delays.

As an example, consider the network in Fig. 34(a), where nodes 1, 2 and 3 are fixed and node 4 is mobile (circles with dashed lines represent the nodes' radio range). Using a *window size* of 128 packets and an OGM interval time of 1 s, as proposed in [70], the BATMAN routing tables at the four nodes are as shown in Table 4. Observe that node 1 does not receive OGMs from node 4 and, thus, it uses node 2 to reach node 4; also, all routes report the maximum value of the packet count metric. Now, let node 4 start moving toward node 3 at  $t = 200$  and reach the new configuration shown in Fig. 34(b) at time  $t = 220$ . The routing tables at the four nodes at  $t = 220$  s are shown in Table 5. Note that, within a time out, set by default to twice the OGM interval, nodes purge a neighbor from which OGMs are not received any longer. Thus, nodes 2 and 4 realize that their link has failed and update their routing tables. On the contrary, node 3 starts receiving OGMs directly from node 4 and updates its stored information accordingly, however, since BATMAN does not discriminate between recent and older events, node 3 maintains node 2 as next hop to reach 4. It follows that node 4 can successfully send data to any other node, instead in the opposite direction losses may arise. Suppose that node 1 wants to send a data packet to node 4. The packet is



(a) Uplink traffic



(b) Downlink traffic

Figure 33: Protocols comparison in terms of average throughput, as the offered traffic load varies.

delivered to node 2, which forwards the packet to node 3. Node 3 checks its routing table and re-sends the packet back to node 2; the packet is then bounced between node 2 and node 3 until the *TTL* expires and the packet is finally dropped.

Based on the above observations, we modify the sliding window management so as to sensibly reduce the problem, without altering the spirit of BATMAN or increasing its complexity. Our simple solution consists in changing the way OGMs are counted in the sliding window so that newer OGMs, representing fresh information, are weighted more than older OGMs.

The weights, however, must be chosen in such a way to avoid fluctuations in the routing table. Indeed, if the freshest OGMs are weighted too much relatively to the previous ones, it may occur that the reception of OGMs from different neighbors with similar link quality leads to continuous change of the best next-hop towards a destination. Through extensive simulations, we found that in WMNs with mobility support, a good choice for the values of the weights vector is as follows

$$w(i) = \max\left(1, \left\lfloor \frac{i \cdot S}{e^i} \right\rfloor\right) \quad i = 1, \dots, S \quad (31)$$

where  $w(i)$  is the  $i$ -th element of the weight vector and  $S$  is the sliding window size (namely, 128);  $w(0)$  and  $w(S)$  are associated to the freshest and oldest OGM, respectively.

We name this modified version of the protocol as smart window (sw-) BATMAN. By looking at

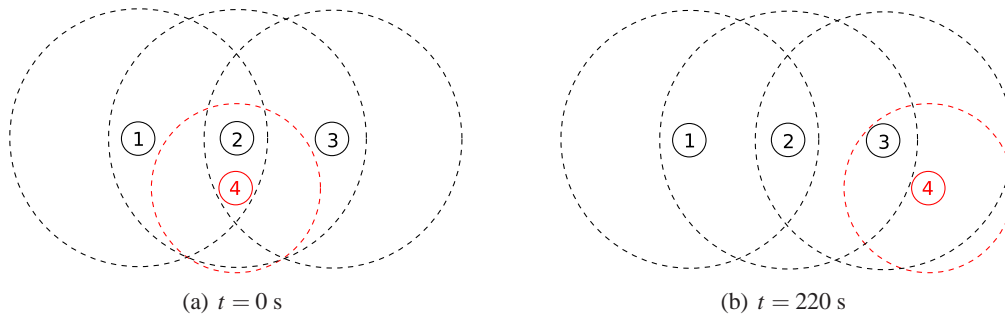


Figure 34: Network topology at different time instants.

Table 4: Routing tables at time  $t = 0$  s.

Node 1		
Dest.	Next hop	Metric
2	2	128
3	2	128
4	2	128

Node 2		
Dest.	Next hop	Metric
1	1	128
3	3	128
4	4	128

Node 3		
Dest.	Next hop	Metric
1	2	128
2	2	128
4	2	128

Node 4		
Dest.	Next hop	Metric
1	2	128
2	2	128
3	2	128

Fig. 35 and comparing these results (obtained in the same simulation scenario and settings as before) with the curves in Fig. 33, we now notice that sw-BATMAN provides the best results both in upload and download. We therefore select sw-BATMAN as the best candidate for implementation in our testbed.

### 7.2.3 Future work

As for wireless mesh networks with mobility support, future research will focus on the development of a testbed in a vehicular setting, where mesh nodes implement the smart window version of BATMAN. The goal will be to show through an experimental performance assessment that mesh networks can successfully support highly demanding applications in a realistic urban environment.

Table 5: Routing tables at time  $t = 220$  s.

Node 1		
Dest.	Next hop	Metric
2	2	128
3	2	128
4	2	125

Node 2		
Dest.	Next hop	Metric
1	1	128
3	3	128
4	3	11

Node 3		
Dest.	Next hop	Metric
1	2	128
2	2	128
4	2	117

Node 4		
Dest.	Next hop	Metric
1	3	7
2	3	7
3	3	7

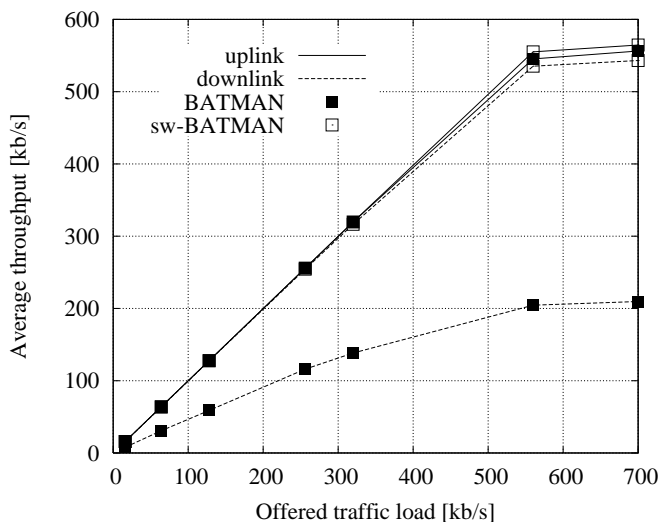


Figure 35: Comparison between BATMAN and sw-BATMAN in terms of average throughput, as the offered traffic load varies. Both the upload and download cases are presented.

## 8 RESOURCE DESCRIPTION LANGUAGE

*CONTRIBUTORS: IST-TUL, CNIT-BO*

This JRA is the result of collaboration among IST-TUL and CNIT-BO. It is aimed at defining the Resource Description Language and how it can be used in the JRAs inside WPR11 and extensible to other WPs within the project.

### 8.1 Research activities

Communication networks tend to consist of a large number of communicating nodes with heterogeneous characteristics, which try to make communication as efficient as possible [72]. Cross-layering of information is a popular method to expose information of the lower layers to the higher layers and vice versa in order to optimise the efficiency of the resources [73]. Also, each node needs to collect information about neighbouring nodes, as the nodes have to negotiate available resources to communicate on, and also available services [74]. There is a lot of similarity between the management of resources in heterogeneous networks and the management of resources for meta-computing systems [75]. There are many resource management algorithms, which all try to optimize for a certain scenario, however, the main goal of these algorithms is the same, namely to manage the available resources in the node. Given the heterogeneous nature of modern networks, interoperability issues need to be kept in mind when designing new algorithms.

A unified approach for the description of resources, called the Resource Description Language (RDL) is defined. The RDL is a specification for the description of resources, which can then be used by various management algorithms. Within the World Wide Web Consortium (W3C), there is a working group with a seemingly similar task, named Resource Description Framework (RDF) [76]. The RDF is a language for representing information about content in the World Wide Web (WWW). The RDF inserts meta-data into a document, e.g., a webpage. The meta-data is used to describe the actual data the document contains in order to be more easily processed by machines. The RDL concept is not about meta-data, but about describing objects that do not exist as content, such as temperature sensors, queues, etc. The RDL is input to management algorithms and as such there is a need to separate the definition of the resources from the functionality of the algorithms that use them. This way, the algorithms can more easily support heterogeneity of resources, which is beneficial for opportunistic networking, where each node might try to use whatever resources are available at any given time.

The RDL can be used to describe a wide range of resources, such as node-, network-, service-, and community-resources. An algorithm using the RDL does not have to limit itself to one group, but can easily combine information from the various groups in its decision making progress. The main goal for the RDL is to define a re-usable resource description specification for different algorithms, which can only be reached if the resources are described in a standard form. The RDL, for instance, can be used for cross-layering information from the physical layer, so that actions can be triggered from events happening at the lowest layer, such as a fading of the wireless channel in case of a wireless link. Another example is the use of the RDL to describe the resources in a Wireless Sensor Network (WSN). The RDL standard will not be closed, but rather will always adapt to new evolutions in the field.

### 8.2 Concept

The RDL is designed to provide a flexible resource description framework, capable of not only describing the resources per se, but also of contextualizing them as capabilities or requirements, as well as specifying any additional constraints and limitations on their functionality and the policies that govern their use. The resources in the RDL are ordered hierarchically in three layers: communities, services and scalar resources.

The RDL can operate within multiple layers, describing scalar resources, services, and communities. Scalar resources represent the most fundamental aspects of the described objects, i.e., describing what the

objects are and characterizing them both qualitatively and quantitatively. These scalar resources describe such things as node hardware capabilities, as well as any specific metrics or state information provided by the network layers. Services, on the other hand, describe coherent packages of scalar resources that export some sort of functionality. For example, whereas a network router can describe its queues and protocol capabilities as scalar resources, it can export its gateway capabilities as a service, allowing its peers to route packets through it. Finally, communities build upon the concept of services by attaching usage policies and agreements. This enables groups of nodes to set the terms by which they may cooperate to achieve common goals.

Resources, whether they are scalar, services or community ones, belong to nodes in the network. Scalar resources are, most likely, found in nodes in the network, describing what resources the node has. Service resources, as well as community ones can be available either in a single node, a group of nodes or an entire network. The RDL defines two scopes in order to define where the resources exist, or where they can be used. These scopes are the node-scope, where the scope indicates a single node, or a network-scope, where it indicated a group of nodes or a network. At the node level, individual nodes export their resources, advertising them internally or to the network, at the network level, these resources can be presented in an aggregate form, describing entire networks of nodes. These networks, in turn, can take any form, from simple single-hop node clusters to larger multi-hop neighbourhoods, or even entire networks. In fact, the RDL can be used to describe the resources within communities of non-contiguous nodes, building on the concept of logical neighbourhoods. Once these network resources have been collected and described within an appropriate scope, additional information can be cooperatively extrapolated from it, allowing the RDL to perform such tasks as clear channel assessment, and opportunistic spectrum allocation.

### 8.3 Scenarios

In this section a few generic use case scenarios are provided as means to better exemplify how the RDL can be used in several contexts.

#### 8.3.1 *Cross-layering*

Cross-layering is the sharing of information between 2 or more layers of the protocol/network stack in a node. Currently, for every piece of information to be shared, special APIs need to be implemented for every attribute. To be fair, special fixed constructs could be created in order to jointly access some attributes, but this alternative is not very flexible as the construct has to be known at design-time. Since the RDL splits the resources from the functionalities, the resources can be easily shared by all layers. A RDL hook can be implemented at the different layers of the network stack, which can periodically scan the RDL for changes. Another option would be to implement a simple trigger API to alert the hooks at the other layers of changes. This allows information on resources to be shared among all layers much more easily, than in current situations.

Fast Mobile IPv6 (FMIPv6) [77] is an extension on standard Mobile IPv6 [78], where some improvements in latency are obtained by making use of cross-layering. In FMIPv6, the node can send a solicitation request to initiate a handover when detecting new access points (APs). The detecting of the APs is done at the link layer, which has to alert the network layer to send such a request. If the RDL is used, the hooks implemented at the link layer would update the RDL information with the new resource (new AP) and either wait for the hook at the network layer to poll the information or send the RDL trigger. Other algorithms could make use of this sharing of information between layers at the same time, without the need of introducing special APIs for every algorithm.

#### 8.3.2 *Modular middleware solution*

Network resource management is a key issue when developing modular middleware solutions for embedded networked systems and wireless sensor networks, as an intelligently crafted middleware solution

could use this information to automatically adapt applications based on a node's capabilities or the context of its neighbourhood. Additionally, the RDL can be used to describe application level constraints, ensuring that heterogeneous applications are only run within environments where sufficient resources are available to support them.

Using these mechanisms, not only can the RDL play an important role in describing network resources to users, but also within the middleware itself, as input to automatic decision making algorithms. From the user's perspective, the RDL can be used to provide a powerful insight into the networks built-in capabilities and resources, allowing them to create better informed policies. From within the middleware itself, the additional information provided by the RDL can be used to enable automated application optimizations, or even to decide when it would be wise to migrate a particular application or functionality to another, more capable, node. Under these circumstances, the RDL can be a key element in enabling opportunistic resource sharing and creating more powerful, context-aware, embedded applications.

### 8.3.3 *Community*

In the IST-WIP project [79], a community is defined as "a community is a group of entities which cooperate to reach a common goal." The idea behind the community concept is that people can offer services or use services from other members in their community. Examples of communities are: supporters of the same football-club or people who live in the same neighbourhood. The FON network [80] is a community where members share their internet connection by allowing members to use their Wireless Local Area Network (WLAN) gateway. In this example, the constraint for anyone to use the WLAN is that they have to be a member of the FON community. These services and constraints can be easily published with the RDL. By using a standard format, like the RDL, the specifications can be understood by various communities. It would also allow a member to define what to offer to whom in a standard manner.

## 8.4 **RDL within WPR.11 JRAs**

The RDL can be used in the JRA "Opportunistic localization and tracking", where the nodes that have GPS capabilities, or other means to know their physical location, store that information in the RDL. The nodes that do not have location information, can build their localization information from the RDL from vicinity nodes. The nodes do not need to have the same location mechanism to determine their position, but can work together as long as the RDL is used for the exchange of information.

Congestion control is a key-issue in transport protocols, especially when Quality of Service (QoS) is requested. The RDL can efficiently collect congestion information from the node and its vicinity, which is useful for the transport mechanisms to be developed.

RDL and routing, have a lot of similarities. Routing tables are based on information on local nodes, so to make a decision where to send the data to in the next hop. The RDL could be used to enhance the routing decision, by taking the state, localization information and the quality of the links into account, which can be found in the RDL.

In peer-to-peer techniques, the RDL can be a key-instrument for announcing and discovering peer-nodes, -services and -capabilities. It can also be seen as a way to do data-aggregation in a WSN, as proposed by [81] through the use of resource vectors.

In opportunistic spectrum access, you need to collect the information on the available radio interfaces from the nodes in the vicinity. Each node needs to always have its RDL updated with the available spectrum. Based on the RDL, nodes can negotiate a free channel within the vicinity. Such a mechanism could be used to reduce the interference in the network, because possible interfering nodes are kept updated on busy and available spectrum. This improves the overall link quality and throughput, by reducing the number of retransmissions.

## 8.5 Format specification

Given the heterogeneous nature of modern networks, interoperability issues come into play. As such, the RDL was built upon standard formats with clear and unambiguous specifications, assuring that different nodes, with different implementations of the RDL, can still successfully manage their resources cooperatively. Furthermore, given the wide array of platforms that can benefit from the use of the RDL and the diverse requirements that they have, two distinct data representation formats have been designed within the RDL, one built on XML and another more compact binary representation, based on Key-Length-Value (KLV) data. Whereas the XML based format is suited for most applications, enabling the use of a vast library of XML based tools and APIs that have already been developed, the compact binary representation could be used in networked embedded systems and wireless sensor networks where the limited network resources and processing power, as well as the need for energy efficiency advise against the use of XML. While the full description of the specification of either of these formats goes beyond the scope of this document, two examples, illustrating the use of each, are provided below in Figs. 36 and 37.

```
<!DOCTYPE rdl PUBLIC "-//IT//DTD RDL 1.0//EN" "http://localhost/rdl.dtd">
<rdl>
<capability>
<scope type="node">
<name>AP10.26</name>
<description>The AP hanging on the wall on the 10th floor</description>
<ipv4address>192.168.8.254</ipv4address>
</scope>
<scalar id="1" type="IEEE802.11g-transceiver">
<description>The Tx/Rx in the access point 10.26</description>
<constraint>
<constrained-value key="channel" operator="greater-equal" value="1" />
<constrained-value key="channel" operator="less-equal" value="13"/>
</constraint>
<constraint>
<constrained-value key="txpower" operator="greater-equal" value="0" />
<constrained-value key="txpower" operator="less-equal" value="20"/>
</constraint>
</scalar>
<service id="2" type="accesspoint" scope="1">
<name>AP10.26</name>
<description>Access point on the 10th floor.</description>
<used-resource id="1"/>
<constraint>
<constrained-value key="SSID" value="eduroam" />
<constrained-value key="WPA2" value="1" />
</constraint>
<constrained-value key="SSID" value="eduroam-guest" />
</service>
</capability>
</rdl>
```

Figure 36: Example of the XML RDL data representation.

## 8.6 Conclusions

The main goal for the RDL JRA is to define a re-usable resource description specification for different algorithms, which can only be reached if the resources are described in a standard form. By separating the actual resources from the functionalities behind the algorithms that use them, new systems can take advantage of a pool of available resources. Within this JRA, the following contributions have been made, towards this goal:

- The specification of the RDL, describing its architecture and the two data representation formats.
- A first reference implementation, developed in the Java programming language.

```
<capability> = {  
  <scope> = {  
    <scope-type> = <node>;  
    <node-address> = 0;  
  };  
  <scalar> = {  
    <scalar-type> = <acceleration-sensor>;  
    <constraint> = {  
      <constrained-value> = {  
        <constraint-key> = <sample-frequency>;  
        <constraint-operator> = <greater-equal>;  
        <constraint-value> = 1; // Hz  
      };  
      <constrained-value> = {  
        <constraint-key> = <sample-frequency>;  
        <constraint-operator> = <less-equal>;  
        <constraint-value> = 1000; // Hz  
      };  
    };  
    <constrained-value> = {  
      <constraint-key> = <error-margin>;  
      <constraint-operator> = <greater-equal>;  
      <constraint-value> = 1000; // PPM  
    };  
  };  
};
```

Figure 37: Example of the KLV RDL data representation (in a human readable format).

## 9 EXPERIMENTAL ACTIVITIES

*CONTRIBUTORS: ISMB, KAU, IST-TUL, CNRS-LAAS*

This JRA is the result of a collaboration among three institutions, namely ISMB, KAU and IST-TUL. It is focused on experimental activities related to two opportunistic scenarios: MESH networks and Wireless Sensor Networks. Specifically, it will be described how the platforms developed by the JRA's partners can be used to carry out experimental tests inside both WPR11 and the others WPRs of NEWCOM++.

### 9.1 Experimental scenarios

#### 9.1.1 Mesh

When comparing Mesh networks and Wireless Sensor Networks, under the perspective of opportunistic scenarios, the differences between them can be highlighted in terms of basic features which define corresponding specific behaviours.

For example, mesh networks provide an opportunistic scenario with the following main features unavailable for WSNs:

- Broadband connectivity;
- Support of protocols enabling complex routing and forwarding policies and advanced services (including reservation and QoS management).

For this purpose mesh networks have been included in the JRA on Experimental Activities: the above mentioned features allow to study two main experimental scenarios which abstract future practical scenarios which seem likely to occur.

- Mesh networks as opportunistic nets (Figure 38). This configuration of WiFi nodes is conceptually quite close to the use proposed for WSN in Section 9.1.2. However, in this case, the scenario is different in terms of available features (bandwidth and flexibility). So the kind of things which are going to be studied is not optimized towards energy consumption and scalability (number of nodes), but rather focused on different aspects such as: software-defined radio and channel configuration, mobility management and consequent protocol reaction to mobility.

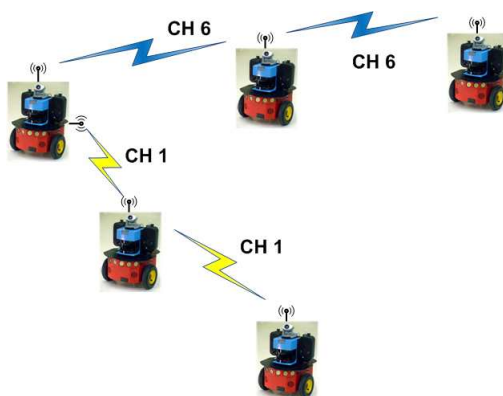


Figure 38: Mesh networks as opportunistic nets.

The corresponding practical scenario should be that of smart mobile nodes spanning from vehicular node to networks exploiting personal smart devices and supporting additional services (such as positioning).

- Mesh networks as backbone wireless networks (Figure 39). In this case the partners refer to a scenario where the broadband connectivity and rich signaling protocol stack are used to connect opportunistic mesh islands. The practical scenario is that of a core network able to face the transmission of large amount of data collected over an opportunistic network. In this case a logical pipe either need not to be continuously active (it would imply bandwidth waste) or, by converse, cannot be always on (in case mobility is involved, the channel should undergo re-configuration). For this purpose a mesh (MPLS) network could support the connection of an opportunistic network island to a remote server - for instance: the benefits would include the opportunity of setting up a broadband channel able to preserve QoS to all the stations already connected and, on the other hand, the ability to cope with a huge asynchrons, delay-tolerant and seldom data.

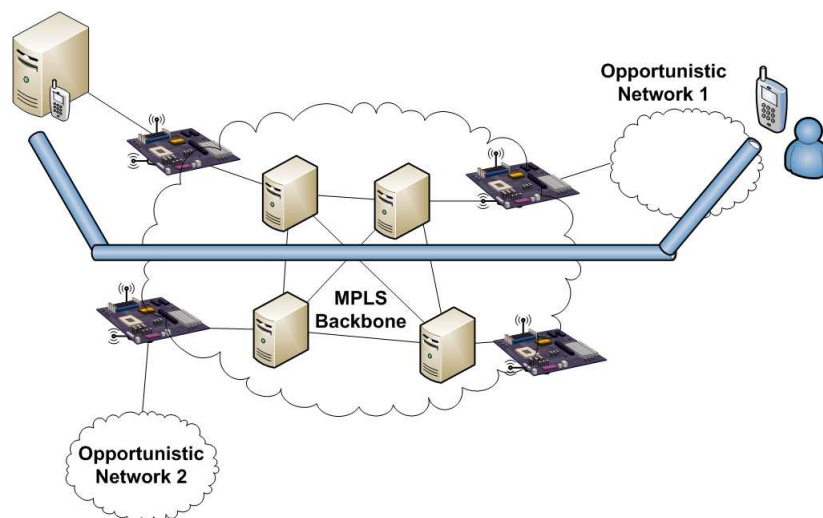


Figure 39: Mesh networks as backbone wireless networks.

This scenario also foresees a study on the interaction between the mesh core networks and the opportunistic network (in practice the experimental study of WSN-WiFi network interaction).

Correspondingly two different sets of measures and tests have to be considered in order to investigate the two scenarios.

#### Mesh opportunistic nodes

- Software-defined radio and opportunistic channel configuration (interaction between dynamic channel allocation and routing, evaluation of channel switching overhead, spectrum usage...);
- Mobility management (i.e. how to react to nodes' mobility and how to exploit position for georouting);
- Protocol reaction to seldom connectivity (i.e. setup time for mesh routing protocols in case of mobility).

#### Mesh network as backbone wireless network

- Comparison between mesh protocols (e.g. OLSR) and MPLS for opportunistic backhauling;
- Quality of service and bandwidth management for the two approaches;
- Rerouting performance for the two approaches;

- Bandwidth efficiency for MPLS (signaling overhead) and for other approaches (QoS impact);
- Setup time for MPLS channel and for mesh network routing.

### 9.1.2 WSN

A set of experiments is to be conducted by the participating partner institutions. These experiments should abide by similar models and produce comparable results. Definition of reference scenarios and metrics of interest is of critical importance in order to achieve this goal.

#### 9.1.2.1 Reference scenarios

Two example scenarios are proposed. These represent typical situations in opportunistic WSNs but, of course, are not a comprehensive set.

**Linear** A group of nodes is organized in a linear topology (Figure 40). A carrier node moves in a horizontal trajectory to the group, and should ideally collect information from all of its nodes.



Figure 40: Linear Pattern Scenario.

**Grid** A group of nodes is organized in a grid pattern (Figure 41). A carrier node that goes through the group will be in range of several nodes, which could lead to a higher radio load and impact the network performance.

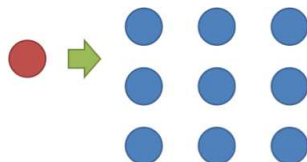


Figure 41: Grid Pattern Scenario.

#### 9.1.2.2 Evaluation metrics

The metrics of interest can be divided into four different sets, corresponding to the different phases on an opportunistic exchange. Out of these, a compact subset should be selected for inclusion in future experiments.

##### Idle

- Energy consumption in idle listening, since this is the most frequent state.

**Detection**

- *Detection delay*: the delay between the point in time when an opportunistic contact could ideally be initiated (the point at which the nodes come in range of each other) and the point in time when the receiver detects the sender. This second point can be related either with the instant when the received signal reaches a given threshold (defined as a function of a combination of RSSI and LQI) or with the instant when the receiver's detection mechanism actually detects the contact in low-power listening (LPL) configurations, whichever comes last.

**Transmission**

- Throughput.
- Packet loss ratio/pattern.

**Post-transfer**

- *Shutdown delay*: the amount of time between transmission of the last successful packet and sender radio shutdown.

**Summative**

- Long-term energy consumption.
- Network lifetime.

**9.1.2.3 Routing algorithm evaluation**

In addition to the agreed characterization of opportunistic contacts, IST-TUL will be conducting some experiments in opportunistic routing for internal use. For this purpose, the same reference scenarios will be used, plus:

- A modified linear topology scenario in which the reader moves in a trajectory perpendicular to the group.
- A new scenario in which a message has to go through multiple carriers to reach its destination, with carriers moving in a periodic trajectory.

In these scenarios, IST-TUL intends to evaluate the following network-wide metrics:

- *Message delivery ratio*: the number of delivered packets divided by the number of sent packages.
- *Message latency*: the time between message generation at the source and message delivery at the destination.
- *Path length*: the number of hops through which a message travels.
- *Buffer needs*: the amount of buffer space required in the nodes.
- *Transmissions per message*: the total number of times a message is transmitted during its lifetime.
- *Reliability*: the number of messages delivered within a defined deadline.

## 9.2 Testbeds

In this Section we describe four testbeds. The first two testbeds are related to mesh networks and have been deployed at ISMB and Karlstad University. The other two testbeds are related to WSNs and have been deployed at ISMB and IST-TUL.

### 9.2.1 ISMB mesh testbed

The Wireless Mesh Testbed deployed at ISMB is composed by three different layers that could be used alone or together in order to test complex scenarios. The first layer (**Backbone Layer**) is made up of eight MPLS capable nodes. The boards used for this layer are Mikrotik RouterBOARD 433AH (Atheros AR7161 680MHz, 128 MB DDR RAM, 64 MB NAND, 3 mini-pci slots). The software running on these boards is RouterOS, which supports a large number of MPLS related features (i.e. Label Switching, Label Distribution Protocol, Virtual Private LAN Service, Reservation Protocol-Traffic Engineering). The main purpose of this layer is to test QoS and bandwidth reservation for opportunistic traffic within the scenario of a wireless backbone network. The main features exploited for this purpose are MPLS over WiFi and the mapping between IP schedulers and WMM/11e.

In wired networks, MPLS has given a satisfying answer to the problem of QoS requirements, in fact it has been adopted in the main international backbones. When we move to the wireless medium, the problem of finding a solution for quality of service in wireless multi-hop networks becomes much more difficult. The use of MPLS on wireless scenarios is complicated because of various causes: the medium is shared and the radio link is not controlled by one station; moreover the radio link quality and the available bandwidth may vary in time. On the other hand, every alternative solution to provide wireless QoS, especially in multi-hop networks, collides with the lacking of control about allocated resources (that is the impossibility of doing Traffic Engineering) and a scarce awareness of the actually available bandwidth. In this sense, emerging protocols like OLSR implement solutions for at most load balancing. MPLS represents a valuable solution, provided that some important problems are solved, i.e.:

- Implementation of reservation mechanisms with strict priority for MPLS flows also at MAC level (necessary to solve access contentions);
- Study of all the factors that can affect the capacity of a wireless medium (number of contending nodes, packet size, signal-to-noise ratio on the link) and study of an adaptation of technology to these factors;
- Rating of these parameters, in order to optimize bandwidth reservation.

The second layer (**Mesh Layer**) is composed by eight Mesh boards. The boards used for this layer are ADI Pronghorn Metro SBC (Intel IXP425, 533 MHz, 64 MB SDRAM - 16 MB Flash memory, 4 mini-pci slots). The main purpose of this layer is to test novel (e.g. 802.11s, OLSR...) and opportunistic (e.g. ExOR) routing protocols – also together with the Backbone layer in order to understand how to fully exploit the two different layers. The software running on these boards is OpenWRT, which is an open source project aimed to create a free embedded linux operating system for network devices based on different architectures (x86, ARM, MIPS, Atheros SoC...). Using OpenWRT it is possible to cross compile a large number of packages, allowing user to easily customize their embedded systems. The typical configuration of OpenWRT for mesh networking consists of: Busybox (tiny versions of many common UNIX utilities into a single small executable), Dropbear (small SSH server and client), different wireless drivers (MadWiFi, Ath5k, Ath9k...), different routing protocols (OLSR, 802.11s) and testing tools (iperf, D-ITG...). Every board has a variable number (from 1 to 4) of wireless interfaces. A wireless interface could be an 802.11a/b/g Atheros based miniPCI or a 802.11a/b/g/n Ubiquity miniPCI. All this interfaces are connected with a 2 dBi omnidirectional antenna. The network topology could be dynamically arranged working on WiFi channels.

The third layer (**Mobility Layer**) is composed by three mobile robots (Pioneer P3DX) and personal devices (handheld devices, webcams...). Every robots is equipped with a ADI Pronghorn SBC (Intel IXP425, 533 MHz, 64 MB SDRAM - 16 MB Flash memory, 3 mini-pci slots) running OpenWRT. The robots are equipped with software for indoor localization (Positioning by WiFi enhanced with other positioning techniques). This localization can be used to exploit positioning for robot deployment and geo-routing. Robots are also able to transmit images captured by their internal video camera. In addition, a wireless webcam could be added in every point in order to test video streaming capabilities. The purpose of this layer is to test how to opportunistically connect mesh island through mobile nodes (i.e. mobile robots). The experimental scenarios for this layer mainly deals with *mobile* data collectors (i.e. store-carry-forward approach for delay tolerant networks) and *position* aware robots coordinated in order to create temporary bridges between mesh islands.

### 9.2.2 KAUMesh

KAUMesh is an experimental Wireless Broadband Mesh Network based on 802.11a/b/g WLAN based devices (mesh nodes) that has been deployed at the Karlstad University Campus. KAUMesh comprises currently 20 mesh nodes, which are permanently installed to cover large areas inside the new Engineering Building. Standard 802.11b/g based clients can be used to connect to the mesh as they would to any given normal WLAN access point. In that sense, a mesh network appears as a large virtual access point to the mesh clients, which only need a W-LAN card to connect. The mesh network is deployed of two main connected regions, each one with multiple gateways to the Internet, as well as of one smaller opportunistic island, which has no gateway and, consequently, need to communicate with the rest of the network opportunistically, through, for example, mobile nodes carrying packets exploiting the mobility of users roaming around.

Mesh Backhaul connectivity is established through mesh nodes, which wirelessly relay packets among each other and towards the gateway mesh nodes, that connect the mesh to the Internet. The number of interfaces used for backhaul access to create the mesh physically limits the number of transmissions and receptions that can take place simultaneously. By increasing the number of interfaces, the upper bound on the capacity of the network can be increased. Due to several limitations, the number of physical interfaces cannot be increased arbitrarily and the number of wireless channels will be the limiting factor. Multiple channels are needed for multiple interfaces to transmit simultaneously. Depending on the country, there can be up to 14 channels available in 802.11b/g and 13 channels in 802.11a.

KAUMesh is a multi-radio, multi-channel mesh solution, which runs on a high-performance network processor platform based on Linux OS. As such, each mesh node has multiple physical radio interfaces. One interface is used for transparent client access, using typically 802.11b/g mode. The mesh backhaul connectivity is established using multiple radio interfaces per mesh node. One of these interfaces is called 'fixed interface' and is assigned for relatively long intervals of time to a 'fixed channel'. This fixed channel is picked initially at random and automatically adapts over time according to the number of users on a given channel to balance traffic. The second interface at each node is called 'switchable interface' and can be switched dynamically and opportunistically between any of the remaining channels according to traffic demand. The fixed channel of a node is selected using a fixed channel assignment and selection protocol. All data sent to a node must be over the fixed channel, because the node is guaranteed to always listen to the fixed channel. A node may send data to a neighbor using the fixed interface if both nodes use a common fixed channel; otherwise, the switchable interface is tuned to the neighbor's fixed channel, and data is sent out over the switchable interface. In that sense, the mechanisms do not require coordination channels. KAUMesh features also the 802.11e based extensions for enabling service differentiation at the MAC layer in the backhaul interface. Finally, different schedulers are available which decide which channels to serve next for the switchable interface (round-robin and hierarchical multi-level QoS aware).

For monitoring purpose, each mesh node has a wired Ethernet connection to a managed network in order to send monitoring data to a monitoring server, located in the fixed network. It serves also for sending configuration information to the mesh nodes as well as for downloading kernel versions to the

mesh nodes. KAUMesh implements an In-Mesh monitoring solution which allows to monitor one-way delay, packet loss, and jitter for VoIP flows. For example, having information on the used codec, allows to estimate the Mean Opinion Score (MOS) for VoIP using e.g. the R-Model within each mesh node. Other statistics are available such as the bandwidth used by TCP flows, and different routing metrics such as WCETT, link qualities, etc. A centralized monitoring server allows to monitor traffic and operational status. Access to the testbed is possible by remote login both for individual nodes and the mesh monitoring server which qualifies the testbed for performing remote experiments. As the mesh nodes run Linux OS and MADWiFi compatible cards, is also possible to change the routing protocol, the channel assignment, or modify parameters at the MAC layer.

### 9.2.3 ISMB WSN testbed

The ISMB test-bed consists of a flexible setup, meaning that the more general WSN can be constituted by an undefined number of nodes, each one playing a specific role as described by the three typologies of nodes introduced below. Additional nodes can be inserted in the network when desired as well as existing nodes can be removed from it. The minimal setup only requires a source node and a destination node. The three typologies of nodes are:

- source nodes: original sources of network traffic, they generate data packets, each one destined to a specific destination node;
- destination nodes: each of them only waits for the reception of data packets specifically addressed to itself;
- mule nodes: their task is to temporary store data packets originated by some source node and opportunistically received, to carry them for a while, and finally to deliver them to the destination node identified as the final recipient.

In order to make the realization and actual deployment of the test-bed concretely doable, only a typology of nodes is endowed by mobility capability: mule nodes, provided that at least one representative of theirs belongs to the WSN, or, alternatively, source nodes. On the contrary, destination nodes, which are connected to a PC via USB serial cable, are always static. In case data mules take part in the opportunistic WSN, source nodes and destination nodes are out of common radio coverage regions. A complete description from the hardware and the software points of view is reported in following sections.

#### 9.2.3.1 Hardware platforms

The ISMB test-bed has been realized by means of commercial-of-the-shelf sensor nodes belonging to TelosB Crossbow platform. Imote2 platform nodes, analogously commercialized by Crossbow, can be easily adopted too, since they share the same radio transceiver as TelosB nodes, the CC2420. Nonetheless, even if we have a few Imote2 nodes at disposal, up to now we have utilized TelosB nodes only. The main technical difficulties in realizing the test-bed concerns the mobility capability. The problem has been solved by using a Lego Mindstorms NXT robot, which can be programmed to travel according to the desired mobility scheme. The additional hardware elements taking part in the ISMB test-bed are one or more pcs, which destination nodes are physically connected to, where data packets received by destination nodes are forwarded and definitively stored for post processing purposes.

#### 9.2.3.2 Software applications

The firmware applications flashed on board nodes are written in nesC language using the Tinyos development tool (currently in the Tinyos 1.x version but partially already ported under the more recent Tinyos 2.1). From the application perspective the network functions as follows. Source nodes, at least one representative is necessary, generate data packets univocally identified by the source node, the final recipient

(chosen in the set of destination nodes) and an appropriate sequence number to distinguish different packets for the same recipient. A data packet originated by a source node can be either directly delivered to the final recipient, if the two nodes are in the same radio coverage area, or transmitted to one or multiple data mules that can hear the source node at that moment. The source node periodically retransmits the same data packet until an acknowledgement, confirming that the packet has been successfully transferred, is received from some other node (mule or final destination). Mule nodes, when present in the network, temporarily collect data packets originated by any source node and addressed to any destination node, according to their currently available storage resources. They will deliver them to the final destination when a contact is realized with it, provided that it is actually found and that the information is not too stale. The interaction between source node and mule node is similar to the one between mule node and destination node, that is, an acknowledgement is necessary to let the sender stop retransmitting a data packet. The role of a destination node is simply to receive data packets destined to itself and to confirm the successful reception with an acknowledgement, while discarding any other received packet.

The application scheme described so far leaves several open issues about the behavior of mules in storing, carrying and forwarding data packets. To this purpose, it is possible to define, implement and evaluate optimized opportunistic protocols which could dynamically manage the interaction between a couple sender-receiver in time-varying channel conditions in order to maximize metrics like data goodput and/or throughput, and at the same time taking into consideration information staleness.

#### 9.2.4 Tagus-SensorNet

The Tagus-SensorNet (Figure 42) is the experimental Wireless Sensor Network (WSN) test-bed that has been deployed at the Taguspark campus of the Instituto Superior Tecnico - Technical University of Lisbon (IST-TUL). The Tagus-SensorNet is currently made up of 29 nodes, permanently installed within the IST-TUL Taguspark main building, as well as 3 mobile nodes and several additional nodes that may be used for general application development and debugging.



Figure 42: IST-Taguspark main building map and Tagus-SensorNet Topology.

The network is composed of three main connected regions (towards the middle and left side of the map), each with its own sink node, as well as of two smaller opportunistic islands (on the right side of the map), which have no sinks and, consequently, communicate with the rest of the network opportunistically, through Mobile Ubiquitous LAN Extensions (MULEs). The following sub-sections explain the Tagus-SensorNet hardware and software platforms, as well as the network's organization, in further detail.

##### 9.2.4.1 Hardware platforms

The Tagus-SensorNet has been primarily built using commercial-of-the-shelf sensor platforms comprising, today, Crossbow MICAz and MICA2 sensor nodes, MTS300/310 and MDA300 sensor boards,

MIB510, MIB600, and SPB400 Stargate Gateways. While several custom hardware solutions have already been developed, these currently take the form of specialized add-on daughter boards that can be coupled with the Crossbow motes to enhance their capabilities, adding such capabilities as ultra-sound localization, vibration and soil moisture measurement, and a LCD/keypad interface. Recently, Sun Microsystems SunSPOTs were acquired and will be used in expanding the Tagus-SensorNet with additional sensor platforms, thus further promoting heterogeneity within the network. Plans do exist to include a custom designed, internally manufactured, sensor platform, the MoteIST Mote, as well as Crossbow IRIS motes, and specialized gas detection and speech reproduction boards.

#### **9.2.4.2 Software applications**

Given the heterogeneous nature of the Tagus-SensorNet, a specialized software framework has been developed to allow multiple concurrent applications to coexist within the network and operate in an integrated fashion. This framework is used as a set of common libraries and APIs that are shared by all applications within the network, providing the following services:

- **Convergecast Data Collection:** This service allows embedded applications to send data towards a sink, ultimately delivering it to the corresponding computer application within the graphical management console. Additionally, the system also allows fixed routes to be configured into the network in run-time, allowing users to change the networks routing topology on the fly. This is used to collect data from the sensor network, for example sensory data, application state information, or debug information.
- **Unicast Data Delivery:** Unlike the convergecast data collection service, the unicast data delivery service forwards data from a sink to any specified destination node within the network. This allows the application running on the management console to communicate with individual nodes, conveying orders or new run-time parameters.
- **Time Synchronization:** This service uses the FTSP time synchronization protocol to create a network-wide clock that can be used by any of the applications.
- **Power Management:** The power management service uses the synchronization service to turn the motes' radios on and off synchronously across the network, effectively duty-cycling its usage. This extends the sensor network's life-time ten-fold.
- **Centralized Graphical Management Console:** The centralized graphical management console consolidates each application's graphical interface into a single integrated computer application. This console is also capable of communicating with the sensor network remotely, over the Internet, allowing network users to manage all of the network's applications from outside the University Campus.
- **Local User Interface Panel:** Whereas the centralized Graphical Management Console provides a convenient way for users to interact with the network using a computer, a local user interface panel allows users to interact with individual sensor nodes more directly through an LCD panel and a keypad.

On top of this common framework, the following applications have been developed:

- **Environmental Interaction:** The Environmental Interaction Application is a generic data retrieval, processing and actuation system that allows users to configure what data is collected within the network, how it should be processed, which decisions should be made based on the results, and how these decisions should be acted upon.

- **VibSense:** The VibSense Application is a vibration monitoring system that allows the user to either collect raw vibration measurements or receive periodic peak and RMS aggregate values. While the raw data may be further analyzed to extract such information as the structures vibration modes, the required bandwidth limits its use to one or two sensor nodes within the network. Peak and RMS values, on the other hand, use far less network resources and, as such, can be collected from many more sensor nodes at once, while still providing essential structural health data.
- **Temperature Gradient Map:** This application uses a distributed algorithm to determine the temperature gradient map within a room. Since nodes collaborate to calculate the room temperature gradient, the system offers a more efficient solution than simply collecting the data from each individual node.
- **Remote Monitoring and Control:** This application enables remote access to sensors in one or more wireless sensor networks, through machines or devices that aren't directly connected to them. This way, users may use the Internet, for example, to know the available battery capacity on any given node, or to manually initiate local actuation. This service is composed of a Gateway that gets the queries from the user over the Internet and translates them into native network messages that convey the query directly to the desired sensor node.

### 9.3 Emulation of opportunistic networks

Network emulation is commonly used to evaluate and examine the behavior and performance of applications and transport layer protocols. A major advantage of emulation is that a wide variety of real implementations of the applications/protocols to be studied can be used, which is not the case for simulations. In this joint research activity we examine how network emulation of opportunistic networks can be supported. Our work is based on the KauNet network emulation system. Below we first describe this system and then discuss how it can be extended to support emulation of opportunistic networks.

#### 9.3.1 The KauNet emulation system

KauNet enables emulation-based experiments with a high degree of control and reproducibility of the emulated conditions. An overview of the KauNet emulation system is provided in this section. An illustration of the system is provided in Fig. 43.

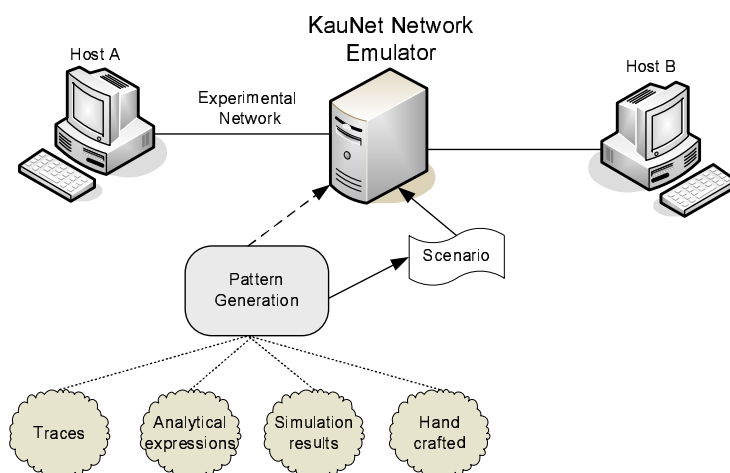


Figure 43: Overview of the KauNet Emulation System.

The design of KauNet is centered around a number of pattern-handling extensions to the well known Dummynet emulator [82], together with user-space programs for pattern creation and management. The use of Dummynet as a starting point provides a stable codebase that has been in wide-spread use for

several years, as well as the integration with the `ipfw` program for emulation setup and management. Dummynet has the ability to lose packets, and to apply bandwidth restrictions and delays to the packets thus emulating the desired link or network conditions. KauNet extends these abilities by also including the ability to introduce bit-errors and packet reordering. Furthermore, KauNet allows deterministic packet losses in addition to the probabilistic losses provided by Dummynet. In fact, KauNet allows bit-errors, packet losses, packet reordering and delay and bandwidth changes to be exactly and reproducibly controlled on a per-millisecond or per-packet basis with the use of patterns.

Compressed patterns are created ahead of time and then inserted into the kernel space under the control of KauNet. During emulation these patterns are then played to control the emulated behavior. For emulations controlled on a per-millisecond basis KauNet operates in time-driven mode and for emulations controlled on a per-packet basis in data-driven mode. In time-driven mode, the index for the pattern is advanced once per millisecond regardless of whether or not any data is transferred. For time-driven bit-errors the amount of movement by the index is coupled to the bandwidth restriction used in the same pipe. This restriction indicates how many bits to process per millisecond. Other kinds of time-driven patterns do not require a bandwidth restriction to be set since they work only on a millisecond level and individual bits do not need to be accounted for. For data-driven patterns, the index move forward one step for each packet, except for bit-error patterns where the index is increased according to the number of bits in the packet.

With regards to emulating multiple connections or links, KauNet inherits the Dummynet firewall rule capabilities that are specified with `ipfw`. These capabilities make it easy to create multiple emulated flows, emulating for example the effective bandwidths and delays for a set of nodes in a mobile network. While Dummynet allows emulated conditions to be changed during the run of the emulation, these changes must be done using command line tools (typically under the control of a script). Using regular Dummynet, it is thus very difficult to exactly control or reproduce an experimental run. As discussed above, KauNet provides enhanced functionality in this area through the use of patterns. However, the use of patterns and command line control can be effectively combined in KauNet. Even though the fine-grained behavior of the emulation is under the control of the patterns, higher-level dynamic events can still be incorporated by using the command line tools to dynamically switch between multiple patterns.

As illustrated in Fig. 43, the KauNet system is flexible with regards to the origin of the patterns, which can be created from collected traces, previous simulations or analytical expressions. In order to simplify creation and management of patterns, a command line tool has been developed to create and manage patterns. The `patt_gen` tool can generate patterns according to several parametrized distributions. It is also capable of importing uncompressed pattern descriptions from simple text files. These text files can be generated by arbitrarily complex models, off-line simulators or trace collection equipment.

One pattern is necessary for each of the controllable parameters, i.e. bit-errors, packet losses, packet reordering, bandwidth changes and delay changes. Since many experiments require simultaneous control of several parameters, multiple patterns need to be managed. A satellite link may for example introduce specific delay characteristics as well as specific bit error or packet loss patterns. To simplify the management of patterns, KauNet supports the grouping of multiple patterns into so called scenarios. A scenario is a combination of several patterns with some additional meta information. Scenarios are also created and managed using the `patt_gen` utility.

As compared to other popular network emulators, such as Dummynet [82], Netem [83] or Nist-Net [84], KauNets pattern-based approach provides a more fine grained and flexible control over the emulated network. A more in depth discussion on the specifics of KauNet in relation to other emulators can be found in [85].

### 9.3.2 *Future work: emulation of opportunistic networks*

Opportunistic networks pose challenges to network emulation as the traditional pattern of end-to-end connectivity is no longer applicable. This implies that it may not be possible to emulate the entire network as a single entity present between application or higher layer protocol end-points. Instead the applications

and higher layer protocols will be present at several places in the network between the original sender and receiver.

Emulation for this communication pattern can be achieved by using several instances of KauNet (or other similar network emulators). Simple synchronization between the nodes may be obtained through the use of `ntp` and `at`. If more advanced synchronization is needed, SWINE [86] could be used for the off-line generation of logically synchronized scenarios for each node.

In KauNet, a limited form of opportunistic connectivity could be modeled using delay or bandwidth limitations. However, applications and higher layer protocols in opportunistic networks typically have to react to the changing connectivity which would be hard to model with this approach. As an improved way of emulating opportunistic connectivity, the extension of KauNet with so called trigger patterns is currently being proposed. The idea with trigger patterns is that a trigger to the application is generated at points specified by the trigger pattern. Through the use of scenarios these triggers could be coupled to other changes in emulated network behavior. This would give the opportunity for applications to react to changes in the emulated network. As a general representation an integer could be used as a trigger allowing different interpretations for different scenarios. A key issue to solve is how the triggers should be integrated with the application.

## 9.4 Contribution to WPR11 JRAs and other WPRs

### 9.4.1 Mesh

Taking into account the features offered by Mesh testbeds (QoS for wireless networks, Novel and opportunistic routing protocols, Node mobility, Radio channel assignment and scheduling...), they could be useful to perform experimental tests on theoretical studies carried out within other WPR.11 JRAs. In particular, the most suitable JRAs for this purpose are:

- *Opportunistic localization and tracking*: how to exploit node's position in order to improve routing (georouting)
- *Opportunistic connectivity: impact of nodes' mobility*: setup and rerouting time for mesh routing protocols in case of mobility
- *Capacity estimation in opportunistic mesh networks*: bandwidth efficiency for MPLS (signaling overhead) and for other approaches (QoS impact)
- *Opportunistic Spectrum Access*: evaluation of channel switching overhead, spectrum usage...

The features offered by these testbeds could be used to perform joint experimental activities within our own JRA:

- *WSN interaction with opportunistic mesh nets*: Comparison between mesh protocols and MPLS for opportunistic backhauling
- *Opportunistic channel selection interaction with opportunistic mesh routing*: interaction between dynamic channel allocation and routing

These testbeds could be also used to validate protocols and result coming from other WPRs. In this section, we briefly outline potential experimental research which can be performed with the help of mesh testbeds. As a result, researchers could use the testbed to validate theoretical and/or simulation results in a real world setup. In a similar way, the testbed can be used to calibrate simulation models against (such as propagation models). This section is structured along the different WPs in Newcom++.

**WPR.1** This WP is focusing on Modeling, calibration, and validation of multi-dispersive, multi-link channels. As a result, KAUMesh provides a good opportunity to conduct measurements which can be used to calibrate link and channel models against realistic interference effects under different combinations of frequencies, channel bandwidth, distances between nodes...

**WPR.2** This WP is focusing on Feedback and Resolution of the Channel State. In mesh testbeds, based on measurements, feedback error could be quantified at e.g. MAC layer taking into account also collisions of hidden nodes.

**WPR.3** This WP is focusing on Adaptive coding/modulation for the wireless channel. An interesting evaluation possibility that mesh testbeds offer is the possibility to select different modulation schemas for different links and channels, combined with different carrier sense thresholds and sending power. Therefore, there is the possibility to evaluate the impact of changing modulation schemas in a multi-channel environment.

**WPR.4** This WP is focusing on Iterative Receivers for Wireless Communications. Due to their nature, potential research issues could only be evaluated in KAUMesh, if we deploy software reconfigurable radios.

**WPR.5** This WP is focusing on Coding for multi-hop networks. KAUMesh is a multihop multi-channel multi-radio indoor testbed. As such it is perfectly suited to study the effect of network coding in this kind of environment. It allows to develop and evaluate novel schemas to optimize e.g. multicast delivery in such networks using network coding. Also, it would be good candidate to develop and evaluate channel assignment algorithms that take into account the opportunity of coded overhearing, etc. So in summary, KAUMesh provides ample opportunity for WPR.5 to validate the schemas, including erasure coding, developed in a real experimental setup.

**WPR.6** This WP is focusing on Relaying and cooperation in networks. Due to its multihop nature, it requires cooperation among nodes. Therefore, it is very well suited to evaluate the performance of novel estimate-and-forward protocols that are based on joint source-channel coding or even joint source-network coding focussing on higher layer and cross-layer aspects. In Virtual Antenna Arrays, it would be very interesting to develop and evaluate novel schemas that make use of superposition coding either for downlink or uplink. In addition, KAUMesh is a very interesting candidate to study the interaction between network layer protocol and cross-layer optimisation techniques. Currently, KAUMesh supports and implements both AODV and OLST routing and includes plug-ins for channel assignment into the routing protocol. In a similar way, it is straightforward with KAUMesh to use additional routing information for the purposes of organising cooperative relaying at the link layer.

**WPR.7** This WP is focusing on Joint Source and Channel Coding/Decoding. We tested already with KAUMesh effective VoIP and Video streaming over multi-channel multi-radio meshed network from within the mesh to destination located in the Internet. Therefore, the incorporation of UEP for streaming application seems feasible to evaluate using KAUMesh. In what congestion concerns, there is also the possibility to incorporate congestion aware mechanisms in the joint source/channel coding and take this into account in the route selection to e.g. maximize user utility for all flows within the network. As a result, KAUMesh is a good opportunity to evaluate novel video delivery mechanisms in a multihop environment.

**WPR.8** This WP is focusing on Scheduling and adaptive radio resource assignment. Due to its self-organizing nature and the possibilities to serve multiple channels and frequency bands in parallel,

it would be interesting to develop and evaluate novel scheduling mechanisms for multi-channel mesh networks using KAUMesh as experimental platform.

**WPR.9** This WP is focusing on Joint RRM and Flexible use of radio spectrum. In KAUMesh, radios can tune to any of the frequency bands available in 802.11 a/b/g band dynamically with minimum switching overhead (around 1 microsec). In addition, radios can be tuned to 5, 10, 20 and 40 MHz. Modulation schemas can be changed as well as sending power and carrier sense threshold. This opens up the opportunity to test and evaluate novel RRM mechanisms and study the effect of e.g. different channel bandwidth on interference and performance. When it comes to channel assignment, currently we use a simple strategy to balance the number of neighbors on a channel. However, more advanced mechanisms can be designed, implemented and evaluated through KAUMesh based on e.g. Game Theoretical Approaches. Such approaches can be combined with cognition capabilities so that the MAC layer can learn better adaptation strategies over time (e.g. which frequency to use, what adaptations of parameters of PHY layer to select, etc.).

**WPR.10** This WP is focusing on Network Theory. Due to its multihop and multi-channel nature, KAUMesh can serve as a vehicle to evaluate different methods for achieving throughput and delay performance limits in a multihop environment.

**WPR.11** This WP is focusing on Opportunistic networks. As mesh testbeds can be deployed as two different not-connected islands, they can be used to evaluate delay tolerant networking protocols utilizing peoples mobility or scheduled links (such as a wireless node placed on an elevator to opportunistically connect those two islands) to disseminate data. In addition, mesh testbeds can serve as a mechanisms to evaluate opportunistic routing protocols (such as Ex-OR) in a multi-channel setting.

#### 9.4.2 WSN

In WSN test-beds, as the ones provided by IST-TUL and ISMB, it is quite difficult to run experiments that could be interesting for other partners. Since WSNs are closely related with the application environment it is not possible to set-up a network configuration generic enough to cope with the specific requirements of each interested partner. Therefore, it is almost impossible to reproduce exactly the setups required by these experiments, as these would likely entail a huge development effort not compatible with the manpower available in this JRA. Hence, this JRA proposes the creation of a website where test-beds measurements are available and can be retrieved by other partners, to drive analytical models or simulators instead of using some other generic source models for obtaining results that otherwise would be less related with real-world test cases.

## 10 CONCLUSIONS

In this deliverable we have presented an overview of the most consolidated Joint Research Activities (JRAs) inside the WPR11 of NEWCOM++. Actually, eleven JRAs have been planned during the first year, however in this deliverable we have considered only the JRAs which have already obtained results (T0+18). In this view, 8 JRAs have been identified for inclusion in the deliverable.

These are:

- Opportunistic Localization and Tracking (Participating partners: CNIT-PD, CNRS-LAAS, ISMB, RWTH);
- Mathematical Modeling of Intermittent Behavior in Opportunistic Networks (Participating partners: NKUA/IASA, CNIT-CT);
- Transport Layer Issues in Opportunistic Networks (Participating partners: CNIT-CT, KAU, CNRS-LAAS);
- Peer-to-peer Techniques in Opportunistic Networks (Participating partners: CNIT-CT, KAU);
- Opportunistic Connectivity: the Impact of Nodes Mobility (Participating partners: CNIT-BO, RWTH);
- Heterogeneous and Opportunistic Wireless Mesh Networks (Participating partners: PUT, IST-TUL, KAU, CNIT-TO, ISMB);
- Resource Description Language (Participating partners: IST-TUL, CNIT-BO);
- Experimental Activities (Participating partners: ISMB, KAU, IST-TUL);

For each JRA, we have discussed the results obtained so far, together with the already published publications and the expected ones. Moreover, with exclusion of "Resource Description Language" which can be considered concluded, we have highlighted the future work planned in the short- and mid-term.

## REFERENCES

- [1] A. Savvides, H. Park, and M. B. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mobile Network Applications*, vol. 8, no. 4, pp. 443–451, August 2003.
- [2] ———, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proc. of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA'02)*, Atlanta, GA, USA, September 2002, pp. 112–121.
- [3] N. Patwari, R. O'Dea, and Y. Wang, "Relative location in wireless networks," in *Proc. of the IEEE VTS 53rd Vehicular Technology Conference (VTC 2001 Spring)*, Rhodes, Greece, May 2001, pp. 1149–1153.
- [4] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005.
- [5] A. Savvides, M. Srivastava, L. Girod, and D. Estrin, *Localization in sensor networks*. Norwell, MA, USA: Kluwer Academic Publishers, 2004, pp. 327–349.
- [6] R. Huang and G. V. Zaruba, "Static path planning for mobile beacons to localize sensor networks," in *Proc. of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2007)*, White Plains, NY, USA, March 2007, pp. 323–330.
- [7] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proc. of the 2002 USENIX Annual Technical Conference*, San Francisco, CA, USA, August 2002, pp. 317–327.
- [8] H. S. Cobb, "Gps pseudolites: theory, design, and applications," PhD Thesis, Stanford University, September 1997.
- [9] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, January 1992.
- [10] Ekahau. [Online]. Available: <http://www.ekahau.com>
- [11] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, June 2000.
- [12] S. Roumeliotis and G. Bekey, "Collective localization: a distributed kalman filter approach to localization of groups of mobile robots," in *Proc. of the 2000 IEEE International Conference on Robotics and Automation (ICRA '00)*, vol. 3, San Francisco, CA, USA, April 2000, pp. 2958–2965.
- [13] A. Howard, M. Matarić, and G. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and System (IROS 2002)*, Lausanne, Switzerland, October 2002, pp. 434–439.
- [14] "IEEE Std 802.15.1 - 2005 IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. - Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, 2005.
- [15] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, 12 2007.

- [16] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor networks," in *Proc. of IEEE INFOCOM*, Anchorage, AK, USA, April 2001, pp. 1655–1663.
- [17] G. Kang, T. Pérennou, and M. Diaz, "Barycentric location estimation for indoors localization in opportunistic wireless networks," in *Proc. of the Second International Conference on Future Generation Communication and Networking (FGCN 2008)*, Sanya, China, December 2008, pp. 220–225.
- [18] —, "An opportunistic indoors positioning scheme based on estimated positions," in *Proc. of the IEEE Symposium on Computers and Communications (ISCC'09)*, Sousse, Tunisia, July 2009.
- [19] F. Zorzi and A. Zanella, "Opportunistic localization: modeling and analysis," in *Proc. of the IEEE 69th Vehicular Technology Conference (VTC Spring 2009)*, Barcelona, Spain, April 2009.
- [20] D. Kliazovich and F. Granelli, "On packet concatenation with QoS support for wireless local area networks," in *Proceedings of the IEEE International Conference on Communications (ICC 2005)*, vol. 2, Seoul, South Korea, May 2005, pp. 1395–1399.
- [21] A. Jain, M. Gruteser, M. Neufeld, and D. Grunwald, "Benefits of Packet Aggregation in Ad-Hoc Wireless Network," Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, CU-CS-960-03, August 2003. [Online]. Available: <http://www.cs.colorado.edu/departement/publications/reports/docs/CU-CS-960-03.pdf>
- [22] M. Castro, P. Dely, J. Karlsson, and A. Kessler, "Capacity Increase for Voice over IP Traffic through Packet Aggregation in Wireless Multihop Mesh Networks," vol. 2, Jeju-Island, South Korea, December 2007.
- [23] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "RFC 218 - TCP Selective Acknowledgment Options," 1996.
- [24] *IEEE Standard 802.11a-1999: Wireless LAN MAC and PHY specifications - High-speed physical layer in the 5GHz band*, IEEE Std., 2000.
- [25] W. Xiuchao, "Simulate 802.11b channel within ns2," Communication and Internet Research Lab, School of Computing, National University of Singapore, Tech. Rep., 2004.
- [26] P. Barsocchi, G. Oliveri, and F. Potorti, "Frame error model in rural Wi-Fi networks," in *Proceedings of the Fifth International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops WiOpt 2007*, Limassol, Cyprus, April 2007, pp. 1–6.
- [27] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a Media Access Protocol for Wireless LAN's." in *Proceedings of SIGCOMM '94*, London, UK, September 1994.
- [28] P. S. David and A. Kumar, "Network coding for tcp throughput enhancement over a multi-hop wireless network," in *Proceedings of the 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE 2008)*, Bangalore, India, January 2008.
- [29] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Computer Networks Journal (Elsevier)*, vol. 47, no. 4, Mar. 2005.
- [30] R. Bruno, M. Conti, and E. Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, Mar. 2005.
- [31] S. Rhea and D. Geels and T. Roscoe and J. Kubiawicz, "Handling Churn in a "DHT"," in *Proc. of USENIX*, June 2004. [Online]. Available: {<http://citeseer.ist.psu.edu/648942.html>}

- [32] A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” in *Proc. of ACM/IFIP Middleware*, Heidelberg, Germany, November 2001.
- [33] S. Rhea and B. Godfrey and B. Karp and J. Kubiatowicz and S. Ratnasamy and S. Shenker and I. Stoica and H. Yu, “OpenDHT: a public DHT service and its uses,” in *Proc. of ACM SIGCOMM*, 2005, pp. 99–100.
- [34] B. Chun and D. Culler and T. Roscoe and A. Bavier and L. Peterson and M. Wawrzoniak and M. Bowman, “PlanetLab: An Overlay Testbed for Broad-Coverage Services,” in *Proc. of ACM SIGCOMM*, 2003, pp. 99–100.
- [35] M. Conti and E. Gregori and G. Turi, “A Cross-Layer Optimization of Gnutella for Mobile Ad Hoc Networks,” in *Proc. of ACM Mobihoc*, May, 2005.
- [36] A. Klemm and C. Lindemann and O.P. Waldhorst, “A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks,” in *Proc. of IEEE VTC-Fall*, Oct., 2003.
- [37] H. Pucha and S.M. Das and Y.C. Hu, “Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks,” in *Proc. of WMCSA*, Jul., 2004.
- [38] D. Malkhi and M. Naor and D. Ratajczak, “Viceroy: A Scalable and Dynamic Emulation of the Butterfly,” in *Proc. of PODC*, Jul., 2002.
- [39] H. Siegel, “Interconnection Networks for SIMD Machines,” *IEEE Computer*, vol. 12, no. 6, pp. 57–65, Mar. 1979.
- [40] I. Stoica and R. Morris and D. Karger and M. Kaashoek and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” in *Proc. of ACM Sigcomm*, Aug., 2001.
- [41] M.F. Kaashoek and D.R. Krager, “Koorde: A Simple Degree-Optimal Distributed Hash Table,” in *Proc. of IPTPS*, Aug., 2003.
- [42] L. Galluccio, G. Morabito, S. Palazzo, M. Pellegrini, M. E. Renda, and P. Santi, “Georoy: a location-aware enhancement to viceroy peer-to-peer algorithm,” *Elsevier Computer Networks*, vol. 51, pp. 1998–2014, Mar. 2007.
- [43] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 88–97.
- [44] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, November 2006.
- [45] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2003, pp. 27–34.
- [46] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 145–158.
- [47] T. Small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. New York, NY, USA: ACM, 2005, pp. 260–267.

- [48] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. New York, NY, USA: ACM, 2005, pp. 244–251.
- [49] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of inter contact times between mobile devices," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2007, pp. 183–194.
- [50] V. Conan, J. Leguay, and T. Friedman, "Characterizing pairwise inter-contact patterns in delay tolerant networks," in *Autonomics '07: Proceedings of the 1st international conference on Autonomic computing and communication systems*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 1–9.
- [51] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2002, pp. 80–91.
- [52] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "CRAWDAD data set rice /ad hoc city (v. 2003-09-11)," uRL: [http://crawdad.cs.dartmouth.edu/rice/ad\\_hoc\\_city](http://crawdad.cs.dartmouth.edu/rice/ad_hoc_city).
- [53] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "Vanetmobisim: generating realistic mobility patterns for vanets," in *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. New York, NY, USA: ACM, 2006, pp. 96–97.
- [54] U. S. Census Bureau, "Tiger, tiger/line and tiger-related products," uRL: <http://www.census.gov/geo/www/tiger/>.
- [55] TSIS, "Corsim," uRL: <http://mctrans.ce.ufl.edu/featured/tsis>.
- [56] L.-J. Chen, Y.-C. Chen, T. Sun, P. Sreedevi, K.-T. Chen, C.-H. Yu, and H. hua Chu, "Finding self-similarities in opportunistic people networks," May 2007, pp. 2286–2290.
- [57] J. Jun and M. L. Sichtiu, "The nominal capacity of wireless mesh networks," *IEEE Wireless Communications*, vol. 10, no. 5, pp. 8–14, Oct. 2003.
- [58] B. Aoun and R. Boutaba, "Max-min fair capacity of wireless mesh networks," in *Proc. IEEE Int. Conf. On Mobile Ad-Hoc and Sensor Systems (MASS)*, 2006, pp. 21–30.
- [59] *802.16 IEEE Standard for Local and Metropolitan Area Networks*, Oct. 2004.
- [60] R. Krenz, "IEEE 802.16 wireless mesh networks capacity estimation using collision domains," in *Accepted for MESH'2009*, Athens, June 2009.
- [61] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *ACM MC2R*, vol. 10, no. 1, pp. 31–43, Jan. 2006.
- [62] C. Chereddi and P. Kyasanur and N. H. Vaidya, "Design and Implementation of a Multi-Channel Multi-Interface Network," in *Proc. of REALMAN*, 2006. [Online]. Available: {<http://http://www.crhc.illinois.edu/wireless/papers/chereddi2006Realman.pdf>}
- [63] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *Proc. of MobiCom'06*. New York, NY, USA: ACM, 2006, pp. 170–181.

- [64] L. Trung-Tuan, L. Bu-Sung, Y. C. Kiat, and A. Kessler, "Distributed Channel Scheduling Algorithms to Minimize Channel Interference in Multiple Channels Multiple Interfaces Environment," *submitted to: Transactions on Mobile Computing*, 2009.
- [65] A. Capone, S. Napoli, and A. Pollastro, "MobiMESH: An Experimental Platform for Wireless MESH Networks with Mobility Support," in *Proc. of ACM QShine, Workshop on Wireless mesh: moving towards applications (WiMESHNets)*, Aug. 2006.
- [66] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," in *IETF Experimental RFC 3561*, July 2003.
- [67] "IEEE P802.11/D2.02, Draft Standard, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 10: Mesh Networking," Sept. 2008.
- [68] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," in *IETF Experimental RFC 3626*, Oct. 2003.
- [69] B. N. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. of ACM/IEEE International Conference on Mobile computing and networking (MobiCom)*, Aug. 2000.
- [70] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better Approach To Mobile Ad-hoc Networking (BATMAN)," in *IETF Work In Progress Internet-Draft*, Apr. 2008.
- [71] J. Wang, Y. Yang, and R. Kravets, "Designing Routing Metrics for Mesh Networks," in *Proc. of IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.
- [72] L. Ferreira, A. Serrador, and L. Correia, "Concepts of simultaneous use in mobile and wireless communications," *Wireless Personal Communications*, vol. 37, no. 3, pp. 317–328, May 2006.
- [73] L. Hensgen, T. Kidd, D. S. John, M. Schnaidt, H. Siegel, T. Braun, M. Maheswaran, S. Ali, K. Jong-Kook, C. Irvine, T. Levin, R. Freund, M. Kussow, M. Godfrey, A. Duman, P. Carff, S. Kidd, V. Prasanna, and A. Alhusaini, "An overview of MHSN: the management system for heterogeneous networks," in *Proc. of 8th Heterogeneous Computing Workshop [(HCWt'99)*, San Juan, Puerto Rico, Apr. 1999.
- [74] J. Zander and S. Kim, *Radio Resource Management for Wireless Network*. Artech House, London, UK, May 2001.
- [75] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A resource management architecture for metacomputing systems," in *chapter in Job Scheduling Strategies for Parallel Processing*, Springer, Gernamy, May 1998.
- [76] F. Manola and E. Miller, "Rdf primer," in *W3C Recommendation*, Feb. 2004.
- [77] R. Koodli, "Fast handovers for mobile ipv6," in *Internet Engineering Task Force, Internet Draft*, Oct. 2003.
- [78] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in ipv6," in *Internet Engineering Task Force (RFC-3775)*, Jun. 2004.
- [79] WIP, *WIP, Solutions: mesh networking, multi-hop relaying, crosslayer design, communities, operator/cellular assistance*. IST-WIP project Deliverable 1.3, EC, Jul. 2007.
- [80] (2009) Community Wireless Solutions. [Online]. Available: <http://www.fon.com>

- [81] R. Verdone, “Environmental opportunistic networks (EONs) - a concept paper,” in *COST2100, 4th MCM, TD(08)450*, Wroclaw, Poland, Feb. 2008.
- [82] L. Rizzo, “Dummysnet: A simple approach to the evaluation of network protocols,” *ACM Computer Communication Review*, vol. 27, no. 1, pp. 31–41, January 1997.
- [83] S. Hemminger, “Network Emulation with NetEm,” in *Proc. of the Linux Australia Conference (linux.conf.au 2005)*, Canberra, Australia, April 2005.
- [84] M. Carson and D. Santay, “NIST Net: A Linux-based Network Emulation Tool,” *ACM Computer Communication Review*, vol. 33, no. 3, pp. 111–126, 2003.
- [85] J. Garcia, P. Hurtig, and A. Brunstrom, “The Effect of Packet Loss on the Response Times of Web Services,” in *Proc. of 3rd International Conference on Web Information Systems and Technologies (WebIST2007)*, Barcelona, Spain, March 2007.
- [86] E. Conchon, J. Garcia, T. Pérennou, and M. Diaz, “Improved IP-level Emulation for Mobile and Wireless Systems,” in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC07)*, Hong Kong, China, March 2007.