

A negotiation-based networking methodology to enable cooperation across heterogeneous co-located networks

Eli De Poorter, Pieter Becue, Milos Rovcanin, Ingrid Moerman and Piet
Demeester

Department of Information Technology (INTEC), Ghent University - IBBT
Gaston Crommenlaan 8 - bus 201, 9050 Gent, Belgium
Email: eli.depoorter@intec.ugent.be

Abstract

In a future internet of things, an increasing number of every-day objects becomes interconnected with each other. Current network solutions are not designed to connect a large number of co-located devices with different characteristics and network requirements. To cope with increasingly large and heterogeneous networks, this paper presents an ‘incentive driven’ networking approach that optimizes the network performance by taking into account the network goals (‘incentives’) of all individual devices. Incentive driven networking consists of the following steps. First, devices dynamically search for co-located devices with similar network preferences and hardware and/or software capabilities. Next, if such devices are found, communities consisting of interconnected objects with similar network expectations are formed on an ad-hoc basis. Due to the similarities between the involved devices, it is easier to optimize the network performance of each individual community. Finally, different communities can cooperate with each other by activating and sharing (software or hardware) network resources. The paper describes which (future) research is needed to realize this vision and illustrates the concepts with a number of simple algorithms. Through an experimental proof-of-concept implementation with two networks of resource-constrained embedded devices, it is shown that even these simple algorithms already result in improved network performance. Finally, the paper describes a large number of example use cases that can potentially benefit from our innovative networking methodology.

Keywords: Network discovery, network negotiation, network cooperation, network optimization, network coexistence, self-organization, self-configuration; experimental validation.

1. Introduction

In the future, an increasing number of objects will be (wirelessly) connected with each other [1, 2]. The rising popularity of wireless car ports, televisions, radios, rolling shutters and different types of environmental sensors demonstrates that even every-day household objects will come equipped with (wireless) communication possibilities. Supporting connectivity between these fixed and mobile objects enables an increased interactivity with our environment, which in turn enables wireless next-generation applications such as wireless building automation, automated e-health solutions, interactive museum exhibitions and personalized entertainment systems [3, 4].

Nowadays, supporting connectivity between these co-located devices is supported by manually grouping together the different devices in separate subnets based on their network technology. Regardless of the characteristics of the devices, the same network configuration and network policies are used for all the devices of a single subnet.

However, this manual approach is complex and inefficient [5].

- Due to the sheer amount of co-located devices, a manual configuration approach is very time-consuming and expensive, especially in large-scale networks.
- Manual or static configuration approaches do not take into account dynamically changing network requirements such as networks that change over time, networks that use mobile devices or networks that are formed spontaneously after impromptu encounters [6].
- Devices from different subnets typically ignore each other, resulting in harmful interference [7] and missed opportunities for cooperation.
- Finally, current networking approaches most often do not take into account the heterogeneity of next-generation networks. Different devices typically have (i) different network preferences and (ii) different hardware and software capabilities [8].

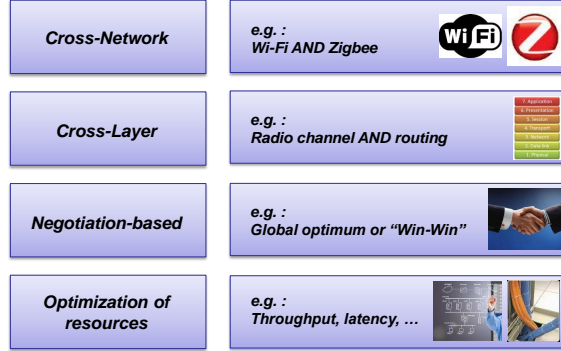


Figure 1: The characteristics of incentive driven networking.

As such, there is a need for network solutions that efficiently support at run-time cooperation between devices from different subnets while taking into account the diverging goals and capabilities of the networked objects. To fill this gap, this paper presents a networking approach called ‘incentive driven networking’ that aims to both (i) *increase the network performance* of co-located devices through cross-network cooperation and (ii) *simplify the configuration* and setup of networks for the end-users.

Incentive driven networking describes a *cross-layer, cross-network negotiation methodology* for optimizing network resources such as throughput or latency (Figure 1). Using our methodology, devices (even from different owners) can engage in efficient cooperation with co-located devices that have different network preferences and capabilities, ultimately resulting in an overall increase in the network performance.

The main goal of this paper is (i) to present and discuss an alternative networking approach that is better suited for heterogeneous networks and (ii) to list the research that is needed to realize this vision. The concepts are illustrated with simple example algorithms that can form the base of future, more advanced incentive driven solutions. To demonstrate the feasibility of the methodology several of these illustrative algorithms are implemented in a proof-of-concept implementation. Finally, the paper will demonstrate that the use of even these simple algorithms already results in an increased network performance.

The remainder of this paper is as follows. Section 1 argued that current networks are not designed to support interconnected objects that differ in terms of behavior and network requirements. As a possible solution, this

paper introduces incentive driven networking. Section 2 defines the terms and concepts that are used in incentive driven networking. Afterwards, Section 3 presents the incentive driven methodology that is used to realize the negotiation based network optimization. The feasibility of the proposed concepts is defended in Section 4, where the performance of an experimental implementation is evaluated. Next, Section 5 gives an overview of related network approaches. Afterwards, Section 6 gives an overview of potential marketable applications that can be realized using the proposed incentive driven networking concepts. Finally, Section 8 concludes the paper.

2. Terminology

Before exploring the methodology, the key components of the cooperation schemes are described in more detail.

2.1. Incentives

Each device involved in incentive driven networking has a number of well-defined incentives that describe the preferred high-level network behavior. An incentive can either (i) describe behavioral aspects of the network (i.e: ‘limit the battery consumption’); or (ii) express the need for additional functionality (i.e: ‘get internet access’); or (iii) give an indication of the expected performance network metrics (i.e: ‘support video streaming’). Example incentives are the following:

- HIGH_THROUGHPUT, HIGH_RELIABILITY or LOW_DELAY (to obtain better QoS guarantees)
- HIGH_NETWORK_LIFETIME (to prevent frequent battery replacement)
- HIGH_COVERAGE (to reach more clients)
- LOW_EXPOSURE (due to health regulations)
- GET_PUBLIC_ACCESS (to get internet connectivity)

Incentives describe the ‘reasons for cooperation’: devices will only engage in cooperation with other devices when this cooperation is beneficial for the incentives of the participating nodes. The incentives of a device are typically set by the application, or configured manually by a network administrator.

When devices have multiple incentives, some of these incentives can be contradictory. In this case, devices should indicate which incentives have the highest priority (see Section 3.3).

2.2. Communities

A community is defined as a set of nodes that have derived common incentives (‘network goals’). As such, a community describes a set of co-located nodes that have the same network behavior and the same network goals: they are similar in terms of capabilities (such as available services) and incentives. All devices of a single community should be able to communicate with each other (either directly, or through intermediate devices that are part of the same community). As an example, the devices of an office building can be divided into the following three separate communities: Wi-Fi enabled devices that are battery powered, Wi-Fi enabled devices that are plugged into a power line and UMTS capable devices.

Devices require a trust relation with all other community members before joining a community. There exist several approaches to create a trust relation between devices. Devices that belong to the same owner are implicitly assumed to trust each other. Otherwise, a trust relationship can be established through the use of a *Public Key Infrastructure (PKI)*, whereby a (remote) trusted certification authority (CA) issues digital certificates to reliable devices [9, 10]. Alternatively, a *decentralized trust model* can be used (the ‘web of trust’) whereby devices can vouch for the reliability of other devices [11]. An example of such a decentralized authentication approach is PGP [12], a security standard that provides decentralized cryptographic privacy and authentication for data communication.

2.3. Network services

An incentive can be realized using a large number of networking techniques. For example, the reliability incentive can be improved by using retransmission schemes, by increasing the transmission power or by using advanced error correction codes. These optimization techniques, that influence one or more of the incentives, are called network services. Thus, whereas incentives indicate network goals, network services are the means to realize these goals.

Table 1 lists several example network services and their influence on the incentives. A network service is not crucial for the correct working of the individual communities, but can be activated or deactivated in a community

Network service	Service description	Expected influence on the incentives			
		Throughput	Delay	Reliability	Network Lifetime
Shared routing	Allows cooperating communities to interpret and route packets from other communities	+	+	+	±
Interference avoidance	Communities cooperate by selecting the transmission frequencies which are least harmful for each other	+	±	+	±
Coordinated sleeping schemes	To conserve energy, the devices from the involved communities use matching sleep schemes	-	-	±	+
Packet aggregation	To reduce the number of transmissions, multiple information exchanges are aggregated into a single packet	+	-	-	+

Table 1: Example list of network services and their influence on community incentives (+: positive influence, -: negative influence, ±: variable or no influence).

depending on the required incentives of the communities. For example, activating retransmissions will positively influence the reliability, at the cost of a lower network lifetime.

2.4. Negotiation profiles

To enable negotiation, the characteristics of each community are described in negotiation profiles. A negotiation profile should contain at least the following information.

- A timestamp (time of last update).
- A certificate guaranteeing that the community can be trusted.

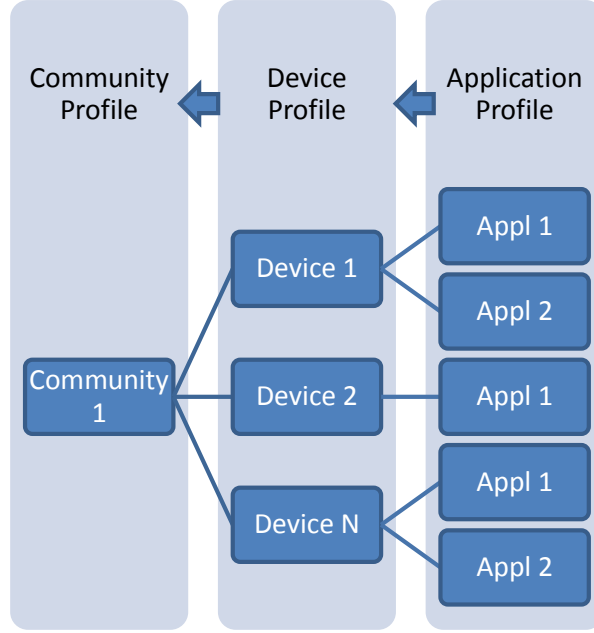


Figure 2: Profiles are constructed in a hierarchical manner. One or more application profiles are combined in a single device profile. Similarly, a community profile is generated based on the profiles of all participating devices.

- The community ID and priority.
- A list of incentives and their associated importance for the community.
- A list of available network services.
- A description of the configurable settings (transmission frequencies, available packet types, etc).

Profile information can be represented using new or existing standardized XML schemes [13]. Alternatively, resource-constrained networks can utilize more efficient binary formats to represent the profile information.

Negotiation profiles are constructed in a hierarchical manner (Figure 2). The incentives of the applications are described in an *application profile*. If only a single application is deployed on a device, a direct conversion from application incentives to device incentives is possible. However, if multiple applications are deployed on the same device, the application profiles are

merged into a *device profile*. In the case of conflicting incentives, different application priorities can be used to prioritize certain incentives. Similarly, the *community profile* represents a merged representation of the device profiles of all participating nodes (see Section 3.1).

2.5. Incentive driven networking

For optimal network performance, it is important to make intelligent decisions about which network services should be activated. Since incentives describe high-level network requirements, an incentive can be improved using different network services. For example, the incentive ‘reliability’ can be improved by utilizing better error correction codes, by activating packet acknowledgments or by using reliable routing protocols. To identify which set of network services should be activated, it is important to note that a single network service often influences multiple incentives. For example, the additional transmissions required for packet acknowledgments influence the network lifetime of battery-powered devices, and the use of reliable routing protocols might increase the end-to-end delays.

Incentive driven networking is defined as the selection and activation of the optimal set of network services in each community with the goal to optimize the incentives of each participating community. Based on this definition, the distinction between incentives and network services can be understood as follows: a network service can be activated or deactivated, whereas an incentive indicates a high-level application or management objective.

3. Incentive driven networking methodology

As stated before, our incentive driven networking approach aims to globally optimize network resources through negotiation based cross-layer and cross-network optimizations. Figure 3 gives a general overview of the discussed incentive driven network methodology. The methodology to support incentive driven networking consists of the following 5 phases, all of which will be discussed in more detail:

1. First, *communities* of similar devices *are created*.
2. The communities use varying communication technologies to *discover* each other.
3. After discovery, the communities *negotiate* about the optimal set of network services.

4. This is followed by the actual *activation* of the services.
5. Finally, the communities *monitor* if all services are actually deployed and if the communities behave correctly.



Figure 3: The 5 phases of the incentive driven network methodology

3.1. Phase 1 - Community Creation

Initially, devices are deployed with a simple (standardized) MAC and routing protocol. They can communicate with co-located devices, but no network services are yet activated. After deployment of the devices, the nodes first find out if they can form a community with similar co-located devices. Joining a community has both benefits and disadvantages for a device. When a device joins a community, the community incentives might differ from the incentives of the individual device. In this situation, the community will optimize towards incentives that are suboptimal for the joining device. On the other hand, by joining a community, the device enters a stronger negotiation position, since a community can negotiate on behalf of a large group of nodes.

The partitioning of devices into separate communities can occur either at run-time or at design-time. To cope with dynamically changing network

Algorithm 1 Outline of a simple algorithm for the partitioning the devices into communities

```

1: Variables:
2:    $P_a$  represents the profile of device  $a$ 
3:    $C_a$  represents the community  $a$  is part of.
4: Initial conditions:
5:    $C_a = \{\}$  (each devices is a separate community).
6:
7: whenever
8:   A profile  $P_i$  is received from another device  $i$ 
9:   Reply with your own profile  $P_a$ 
10: end
11:
12: repeat
13:   Broadcast  $P_a$  over all available communication interfaces.
14:   From the received replies  $P_i$ , select the one that best matches  $P_a$ 
15:   Propose to join community  $i$ 
16:   if community  $i$  accepts then
17:      $P_0 \leftarrow \text{merge}(P_0, P_i)$  (create common community profile)
18:   else
19:     Ignore  $P_i$  from now on
20:   end if
21: until No new devices can be added to the communities

```

conditions, as well as to avoid complex and time-consuming manual network configuration, a non-manual approach is preferred. The end result of this phase is a partitioning of the devices into communities of directly connected devices. As an example, the outline of a simple at run-time partitioning protocol is presented in Algorithm 1. The algorithm assumes that the incentives, services and settings of each community are described using a standardized community profile, which is transmitted over a predetermined radio frequency. The algorithm starts when each individual device creates a device profile based on the requirements of its high-level applications. During the algorithm, devices search for co-located devices which are similar enough to be part of the same community. When no more compatible co-located devices are found, the algorithm is finished. In environments that are strongly heterogeneous, a community might be as small as a single device.

Of course, alternative approaches are possible. Whichever method is used, the end-result of this step is that all co-located devices are divided over different communities consisting of devices with similar services and incentives. Since communities are independent entities, it is now possible to optimize the performance of each individual community. However, optimizing at this phase of the methodology might not always be beneficial due to the following reasons. (i) It is not yet known how the network services will influence the incentives of neighboring communities. To cooperate with other networks, the negotiation output might require that some of these network services are disabled again, thus resulting in unnecessary set-up and configuration overhead. (ii) Additionally, some optimization techniques complicate the discovery and negotiation process. For example, utilizing encryption schemes or channel hopping schemes might prevent different communities from detecting each other.

As such, depending on the complexity and length of the ‘community discovery’ and ‘community negotiation’ phases, it can be beneficial to delay some (or all) network optimizations until after the negotiation process.

3.2. Phase 2 - Community Discovery

In the previous phase, devices were partitioned into separate communities with similar incentives and services. Now, the different communities find out if they are co-located with other communities capable of incentive driven networking. Community discovery consists of the following steps:

1. *Assignment of discovery nodes.* Each community decides on the optimal number of devices that are needed to detect co-located communities. To bear minimal impact on the network performance, a subset of discovery nodes can suffice (see Figure 4).
2. *Community discovery.* Next, the discovery devices are used to detect the co-located communities. Community detection can be *passive* (i.e.: discovery devices passively scan for recognized packets on multiple frequencies in order to overhear existing communities [14]) or *active* (i.e.: by broadcasting community advertisement messages over multiple frequencies containing information about the network settings that should be used to contact the advertising community). The end result of this step is that single hop communication between the discovery nodes of different communities is possible.

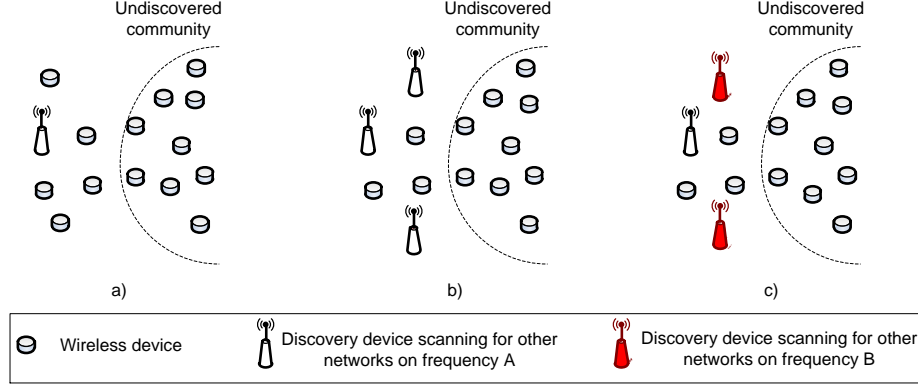


Figure 4: Distributed Community Discovery. a) A subset of the discovery devices is used for detecting other communities. b) Adding more discovery devices increases the probability of successful detection. c) Multiple discovery devices can transmit discovery beacons in parallel on different frequencies to ensure a timely detection of co-located communities.

3. *Profile exchange.* Afterwards, the discovered communication settings are used to exchange the community profiles between the discovery nodes.
4. *Forward the received profile.* Finally, received community profiles are forwarded to the 'negotiation' entity of each community (see next section).

3.3. Phase 3 - Community Negotiation

By now, the co-located communities have exchanged profiles which describe the incentives of each community. The next steps investigate if cooperation between different co-located communities (in the form of activating cross-network services such as interference avoidance) is beneficial. To be able to participate in this step of incentive driven networking, each participating community should have a negotiation entity. This negotiation entity is either a single, central manager that is trusted by both communities or an entity that is distributed over several nodes of each community. Negotiation consists of the following phases:

1. *Announcement of negotiation entity.* The negotiation entity of each community regularly announces its presence to all nodes of the community by broadcasting 'negotiation advertise' messages.

2. *Collection of community profiles.* All received community profiles are forwarded to the nearest negotiation entity where the negotiation process is initialized.
3. *Determine an influence rating for each service.* For each available network service, the negotiation manager determines how the activation of the available service will influence the incentives of each community. For example, enabling aggregation can increase the network lifetime incentive by 30% [15]. To agree on estimated influence of network services, results can be used from (i) existing literature, from (ii) network simulators or from (iii) network monitoring agents.
4. *Calculate optimal set of network services.* Based upon these influence ratios, the negotiation entity calculates the optimal selection of services that should be activated. In some cases, optimal incentive driven networking might require that the communities do not merge, but try to avoid each other (for example by using interference avoidance mechanisms).

To calculate the optimal set of network services, several negotiation approaches are possible based on methods such as game theory [16, 17], self-learning approaches [18] or mathematical formulas. To illustrate the negotiation concepts, this section derives an example negotiation algorithm that uses a heuristic ILP formulation that can be applied to any number N participating networks, using the notations from Table 2.

With:

$$\left\{ \begin{array}{ll} N = & \text{the total number of communities participating in the} \\ & \text{negotiation process.} \\ I = & \text{the total number of incentives.} \\ S = & \text{the total number of available services.} \end{array} \right.$$

It is assumed that:

$$\sum_{i=0}^I IW_{i,a} = 1, \quad \forall a = 0..N \quad (1)$$

Formula (1) enforces that the sum of the incentive weights of each community is normalized to one. For example, consider the situation where two applications are running on node a . One application requires a maximal

Symbol	Meaning
$profit_a$	The profit function of community a . This objective function should be maximized for each network to optimally profit from incentive driven networking.
CP_a	‘Community Priority’ of community a . Under normal operations, the priority of each community equals one. However, the performance of certified emergency networks can be improved by giving them a higher priority, at the cost of a lower network performance of the other participating communities.
$IW_{i,a}$	‘Incentive Weight.’ The weight factor that is given to incentive i in community a .
$SI_{i,a;s,b}$	‘Service Influence’. The percentage by which incentive i from community a is improved when service s is activated in community b . These values can be configured at design-time, or monitoring agents can use learning techniques to intelligently monitor and change these percentages at run-time.
$SA_{s,a}$	‘Service Activated’. A binary variable (0 or 1) that indicates if service s is activated in community a . These variables are determined as the end result of the linear program.

Table 2: List of variables used during the negotiation process.

throughput, whereas the other requires a long network lifetime. Assuming both applications are equally important, the incentive weights would be divided equally amongst both applications, resulting in $IW_{throughput,a} = IW_{lifetime,a} = 0.5$. This normalization requirement is introduced to prevent ‘greedy behavior’: devices are not allowed to set the weight factor of *all* incentives to very high values. Devices are forced to prioritize certain incentives at the cost of other (less important) incentives. This way, devices indicate that they accept a (potential) degrade of the network incentives that are less important to them, in exchange for an increase of the incentives that received the highest weight factors.

Then maximize:

$$\sum_{a=0}^N CP_a * profit_a \quad (2)$$

Subject to:

$$profit_a = \sum_{i=0}^I \left[IW_{i,a} * \left\{ 1 + \sum_{s=0}^S \sum_{b=0}^N (SA_{s,b} * SI_{i,a;s,b}) \right\} \right] \quad (3)$$

$$profit_a \geq 1, \quad \forall a = 0..N \quad (4)$$

$$SA_{s,b} = \begin{cases} 1 & \text{if service } s \text{ is activated in network } b; \\ 0 & \text{if service } s \text{ is deactivated in network } b. \end{cases} \quad (5)$$

The profit of each individual community a is calculated in Formula (3). The formula evaluates how each service influences the incentives of community a when activated. Since initially no services are used ($SA_{s,b} = 0, \forall s = 0..S, \forall b = 0..N$) the profit without incentive driven networking equals one. When new services are added, these services increase or decrease the value of the incentives (for example: using sleep schemes might improve the ‘network lifetime incentive’ by 60%). Depending on the weights of the incentives, the profit function of a community will favor different incentives.

The condition described in Formula (4) ensures that the performance of none of the participating communities is degraded after cooperation. If no solution is found that results in better performance for a community a , this community will not participate in the cooperation. Optionally, condition (4) can be omitted when a community agrees to accept a decreased performance (for example to support nearby emergency networks).

Finally, the last condition in Formula (5) indicates that $SA_{s,b}$ are binary variables. After solving the ILP formulation, the binary variables SA indicate which services should be activated to maximally increase the objective function of all involved communities.

Finally, based on the description of the network services, additional constraints can be added. For example: if a network service needs to be activated over both communities (such as when using frequency hopping), the condition $SA_{freqhop,a} = SA_{freqhop,b}$ is added. Similarly, if network service a requires the activation of network service b , the condition $SA_{service,a} \leq SA_{service,b}$ is added. In case no neighboring communities are found, or if the negotiation process with the neighboring communities fails, the linear program is used to optimize the performance of only a single community (by ignoring the incentive requirements of neighboring communities).

3.4. Phase 4 - Enabling of the incentive driven cooperation

After selecting the optimal set of services in each community, the next phase selects and activates the services in all involved communities. In addition, the settings of both networks are configured such that communication between the different networks is possible.

1. *Propagation of the proposition.* The negotiation entity forwards the decision about the service selection to the discovery nodes of the community. These devices know how to contact the other communities and relay the proposition to the co-located communities.
2. *Additional negotiation (optional).* Depending on the negotiation approach, the conclusions about the optimal set of services reached by both communities might differ. In this case, additional negotiation is required to find a set of services that both communities can agree on.
3. *Confirmation and distribution.* Once all involved communities agree on the service selection, the negotiation server distributes the chosen set of activated services and network settings in both communities.
4. *Service migration (optional).* It is possible that some devices are not up-to-date with the latest software components. If this is the case, these missing or outdated software components must first be installed. To update devices with new network services at run-time, modular network stacks can be used [19, 20, 21]. In some cases, it can be possible to directly exchange network services between different devices.
5. *Activate the settings and services.* Once both communities have received the optimal settings and services, the communities simultaneously switch to the selected configuration. It is possible that some services are activated only for specific packets. For example: a QoS service can be activated to process only packets that contain real-time information.

3.5. Phase 5 - Network monitoring and policy enforcement

Finally, communities will want to check if all other communities are 'playing by the rules', i.e.: are not cheating. For example, a monitoring agent can be used to (i) investigate if the selected services are actually activated and performing as expected and (ii) monitor the actual influence ratios of the activated services. If the set of services changes, or if the measured influence ratios differ greatly from the influence ratios used in the negotiation algorithm, a new negotiation process is started.

Since this phase can re-trigger negotiation, oscillated behavior can occur whereby negotiation is triggered too often (i.e.: for marginal profits). Triggering a new negotiation process only makes sense when the expected (long-term) benefits outweigh the short-term negotiation overhead. As such, the threshold values that trigger negotiation should be determined as a function of both (i) the expected network improvement after negotiation (see Section 3.3) and (ii) the overhead incurred by the negotiation process (see Section 4.3).

Finally, oscillated behavior can also occur when the estimation algorithm is too inaccurate (i.e.: the outcome of the profit estimation function *before negotiation* differs greatly from the measured network performance *after negotiation*). Therefore, if the used estimation algorithm is unreliable, the devices should either switch to a different profit estimation technique or indicate that they cannot participate in the negotiation process.

4. Proof-of-concept implementation

This section experimentally measures the benefits of incentive driven networking. To this end, a proof-of-concept implementation of the cooperation approach was implemented on resource-constrained TMoteSky sensor nodes [22].

4.1. Experimental setup

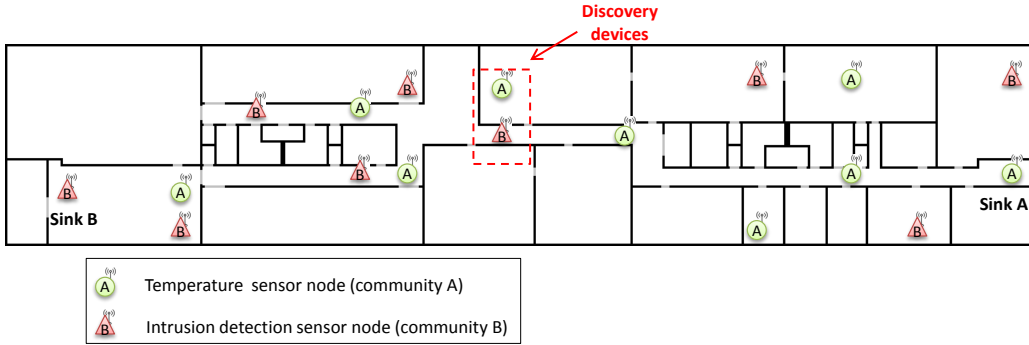


Figure 5: The network used in the proof-of-concept demonstrator. Two types of nodes are deployed: battery-powered temperature monitoring devices (A) and reliable intrusion detection security nodes (B).

For our experiments, the iLab.t wireless sensor testbed [23, 24] was used, which is located in the IBBT - Ghent University office building in Belgium. Figure 5 shows the location of the TMoteSky nodes.

For the proof-of-concept, temperature monitoring sensor nodes (A) were installed in multiple different rooms. These devices sent a temperature report every 10 seconds to the HVAC control unit ('sink device A'), so that the heating, ventilation and air conditioning system (HVAC) can be optimized to reduce its energy consumption. To prevent frequent battery replacements, the main incentive of these battery-powered nodes is the network lifetime ($IW_{lifetime;a} = 0.7$), with reliability as a secondary incentive ($IW_{reliability;a} = 0.3$).

Due to recent burglaries in the area, the proof-of-concept owner decides afterwards to also purchase a wireless anti-theft system, to be installed and operated by an external security firm. Intrusion detection sensor devices (B) are installed at key locations in the building. Since this network has a more critical function, these sensor devices are powered by high-capacity batteries. Every 10 seconds a security report is forwarded to a monitoring PC ('sink device B') which can be accessed remotely by the external security firm. These high-priority information exchanges represent information such as status updates, intrusion alerts or static images from a webcam. To ensure a timely reaction in emergency situations, the devices have stringent delay and reliability incentives ($IW_{delay;b} = 0.5$; $IW_{reliability;b} = 0.5$).

Multi-hop experiments are created by setting the transmission power of the sensor nodes to an output power of -15 dBm. Using these settings, packets require maximum 4 hops to be transmitted from one side of the building to the opposite side. The AODV protocol [25] is used to route all exchanged packets.

The sensor devices from the proof-of-concept are capable of activating two types of network services. (i) Activating '*packet sharing*' enables devices to interpret incoming packets from different communities. In addition, packets can be transmitted to any of the available sinks. (ii) The second network service is the *aggregation service* described in [15], which reduces the number of packet transmissions by aggregating information exchanges from multiple network layers into a single packet.

4.2. Overhead of the community discovery process

When the communities are deployed, they initially function independent from each other at different radio frequencies. On each floor, one device of

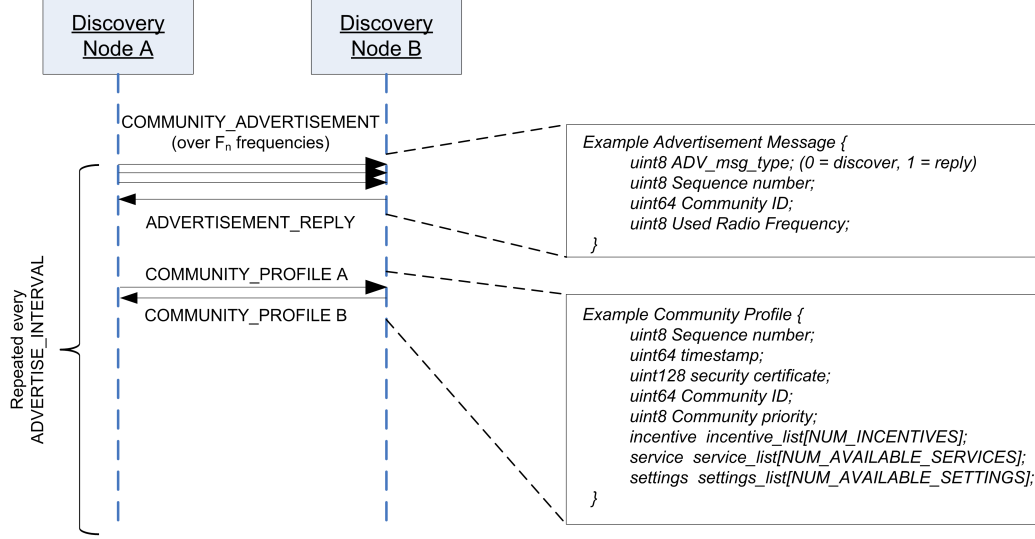


Figure 6: Sequence diagram of the community discovery process from the proof-of-concept demonstrator.

each community was manually assigned the role of discovery device (indicated on Figure 5). The discovery algorithm is illustrated in Figure 6. The discovery devices send out a COMMUNITY_ADVERTISEMENT message every ADVERTISE_INTERVAL time units, which is transmitted sequentially on all available radio frequencies F_n . Discovery nodes from neighboring communities use the information from the COMMUNITY_ADVERTISEMENT to send an ADVERTISEMENT_REPLY using the radio frequency used by the original discovery node¹. Once this connection is established, the COMMUNITY_PROFILE is exchanged between the discovery nodes.

The total number of packets per time unit (PPT) required for the discovery process in community C can be calculated as follows.

¹A short delay is introduced to give the original node time to finish transmitting its advertisements over all frequencies.

$$PPT_{(Discovery; Community C)} = (F_n * DN_C + \sum_{j=1}^{DN_C} \sum_{i=1; i \neq C}^{C_j} DN_{i,j}) * \frac{1}{ADV_C} \quad (6)$$

$$+ \sum_{j=1}^{DN_C} * \sum_{i=j; i \neq C}^{C_j} \frac{2 * DN_{i,j}}{ADV_i} \quad (7)$$

With:

$$\left\{ \begin{array}{ll} DN_C = & \text{The \# of discovery nodes in community C} \\ DN_{i,j} = & \text{The \# of discovery nodes in community i} \\ & \text{that are within reach of discovery node j.} \\ C_j = & \text{The \# of communities that are in reach of} \\ & \text{discovery node j.} \\ ADV_C = & \text{The advertise interval of community C.} \end{array} \right.$$

The first part of Formula 6 calculates the total number of advertisement messages that are transmitted by the discovery nodes of community C per time unit ($F_n * \frac{DN_C}{ADV_C}$). The second part of formula 6 calculates the number of community profiles that are sent in response to advertisement replies from neighboring discovery nodes. Part 7 of the formula expresses that an advertisement reply and a community profile is transmitted in response to each community advertisement that is received from a neighboring community.

In the proof-of-concept implementation, the ADVERTISE_INTERVAL is set to 5 minutes, and F_n equals 16 (all available IEEE 802.15.4 channels). Using these settings, the discovery overhead of our proof-of-concept network is limited to 3.8 packets per minute. For dynamic networks, the advertisement interval should be set to a low value, whereas energy constrained networks or networks that interact rarely would prefer a much higher value to reduce the energy consumption. In large networks, the overhead of the discovery process can further be reduced by intelligently choosing the location of the discovery devices; or by implementing more intelligent discovery algorithms².

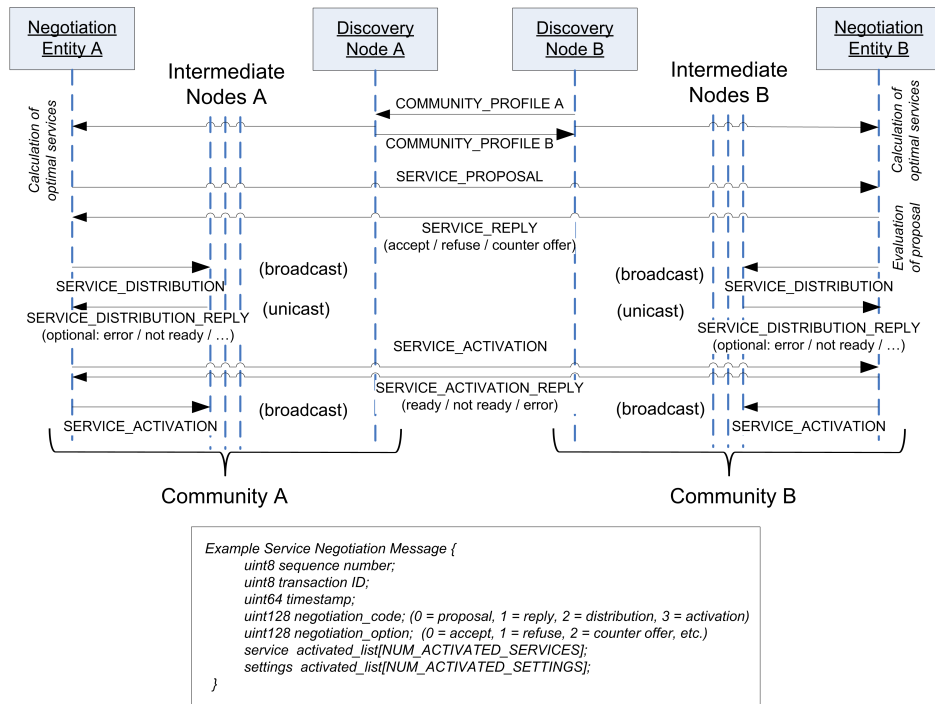


Figure 7: Sequence diagram of the negotiation process from the proof-of-concept demonstrator.

4.3. Overhead of the negotiation process

The negotiation process is illustrated in Figure 7. Whenever the discovery node receives a new or updated community profile, this profile is forwarded to the negotiation entity (the sink) of each community. The linear program from Section 3 is implemented to automatically calculate the optimal set of network services. The negotiation entity will only calculate new service proposals if a new profile is detected, or if the profile information of one of the communities has changed.

After the calculation of the optimal services, service negotiation messages (in the form of `SERVICE_PROPOSALS` and `SERVICE_REPLIES`) are exchanged between the negotiation entities to reach a common decision. Each service negotiation messages includes a transaction ID to keep track of the negotiation process. The type of the negotiation message (proposal, reply, etc.) is indicated by the negotiation code of the negotiation message (see Figure 7). Each message type can also include one or more options. For example, a `SERVICE_REPLY` message can have one of the following options: `PROPOSAL_ACCEPTED`, `PROPOSAL_REFUSED`, `PROPOSAL_COUNTEROFFER`. Finally, the list with services and settings may be omitted from the service negotiation message if they remain unchanged from the previous message with the same transaction ID.

Finally, once both communities reach an agreement on which network services should be activated, a `SERVICE_DISTRIBUTION` message is broadcast by the negotiation entities of each community (see Figure 7). These messages inform the individual devices of each community of the selected set of services and settings. If settings cannot be activated, a device may respond with a `SERVICE_UNAVAILABLE` message. If this occurs, the conflicts must be solved (by installing the missing service, by removing the device from the community, or by renegotiation) before the `SERVICE_ACTIVATION` message can be broadcast. Once every intermediate device has received the activation message the network settings will be changed.

The overhead for the negotiation process (in number of packets) can be calculated as follows.

²For example: a discovery node can choose to send only a single reply message if it is in reach of multiple discovery nodes from the same community.

$$\begin{aligned}
Packets_{(Negotiation; Community\ C)} = & \\
5 * \sum_{j=1}^{DN_C} \sum_{i=1; i \neq C}^{C_j} D_j & \quad (8) \\
+ 2 * (Nodes_C - 1) & \quad (9)
\end{aligned}$$

With:

$$\begin{cases} D_j = & \text{The distance (\# of hops) from} \\ & \text{discovery node } j \text{ to the negotiation entity.} \\ Nodes_C = & \text{The \# of nodes in community } C \end{cases}$$

It is assumed that each discovery node is capable of filtering duplicate profiles (for example, if the discovery device is in range of multiple discovery devices of a neighboring community). Formula 8 calculates the number of packet transmissions between the negotiation entity and the discovery nodes of the community. Part 9 of the formula adds the service distribution and service activation overhead.

In the proof-of-concept, $D_a = 2$ and $Nodes_a = 9$. Using the above formulas, the total overhead of a single negotiation round is 26 packets for each community. Re-negotiation occurs whenever (i) a new neighboring community is discovered; (ii) a neighboring community is no longer available for co-operation; or (iii) the profile of one of the participating communities changes. To account for failing nodes, the proof-of-concept implementation performs a new negotiation process once every hour. As such, the negotiation overhead corresponds to ± 2.9 packets per hour for each node. In most scenarios, this overhead is negligible when compared to the amount of traffic generated by the application(s).

4.4. Evaluation of the available network services

As mentioned in Section 4.1, two network services are available: a *packet sharing service* and an *aggregation service*. For the negotiation process, information about how these network services influence the incentives is required. Figure 8 describes the influence of activating *packet sharing* in both communities. As expected, packet sharing reduces the average number of packet transmissions because (i) nodes can select more optimal paths and (ii) two sinks are now available, thereby reducing the average distance to the sink.

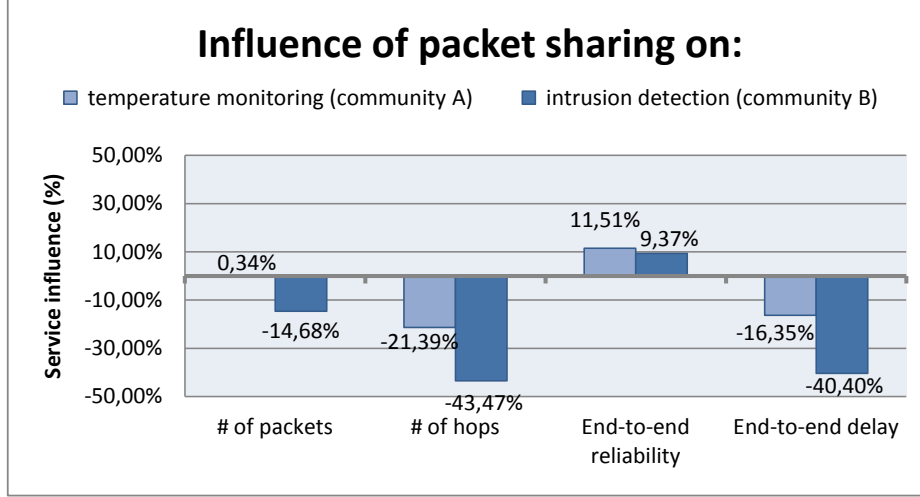


Figure 8: Influence of activating packet sharing on the network performance.

However, whereas community B indeed shows a reduction in the number of packet transmissions, community A instead shows a small increase in the number of packet transmissions, even though the two communities have a very similar topology. This can be explained by the better link quality of community A, which causes traffic from the intrusion detection community B to be off-loaded to the temperature monitoring community A. The average number of hops, the average end-to-end reliability and the average end-to-end delay are improved in both networks, as expected. The main conclusion from Figure 8 is that, while it is often easy to predict the general influence of a network service on the network performance (e.g: positive, negative, ...), it is sometimes difficult to calculate the exact influence rates for an actual network deployment.

Finally, the performance of the aggregation service corresponds closely to the results from [15], and will be discussed in more detail in the next section.

4.5. Performance of the proof-of-concept

To calculate the optimal set of services in the communities, the linear program from Section 3.3 was implemented. The temperature monitoring community A had the following incentives: $IW_{lifetime;a} = 0.7$ and $IW_{reliability;a} = 0.3$. The incentive weights of community B were $IW_{delay;b} = 0.5$ and $IW_{reliability;b} = 0.5$. Based on the influence rates from the previous section, the negotiation entity concluded that the best network performance

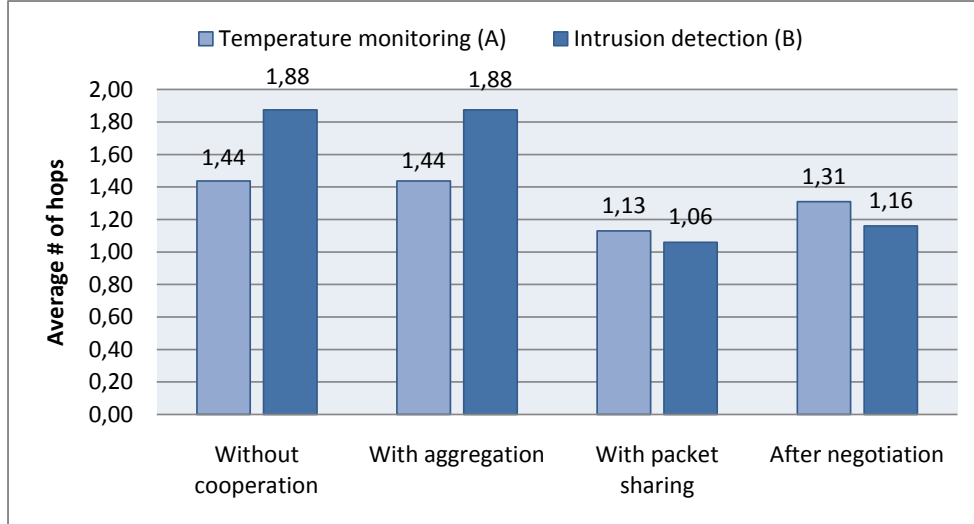


Figure 9: The average number of hops in communities A and B in the following situations: without cooperation between the communities, with packet sharing active in both communities, with aggregation active in both communities, and finally with optimal service selection after negotiation (packet sharing active in A and B, aggregation active in A)

is obtained by: (i) activating the packet sharing service and the aggregation service in the temperature monitoring community, (ii) activating packet sharing in the intrusion detection community, and (iii) using the same radio frequency for both communities.

The network performance for both communities was evaluated in the following situations: without cooperation between the communities, with packet sharing active in both communities, with aggregation active in both communities, and finally with optimal service selection after negotiation (packet sharing active in A and B, aggregation active in B).

Figure 9 shows the average hops that are required to reach the destination. Since aggregation does not result in different communication paths, the aggregation service has no influence on the average number of hops from the nodes to the sink. The number of hops is important however, as the number of hops directly influences the number of packet transmissions, the reliability and the delay of information exchanges, which in turn influence the incentives of the communities (see below).

The average number of packet transmissions is shown in Figure 10. Activating aggregation significantly reduces the number of packet transmissions

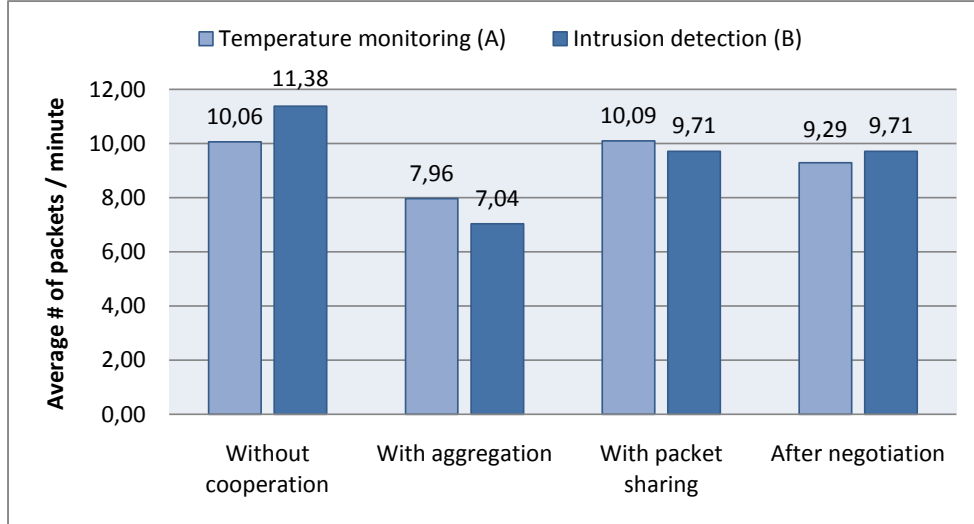


Figure 10: The average number of packet transmissions in communities A and B in the following situations: without cooperation between the communities, with packet sharing active in both communities, with aggregation active in both communities, and finally with optimal service selection after negotiation (packet sharing active in A and B, aggregation active in A)

(by 20-40%). To a lesser amount, activating packet sharing also results in a lower number of packet transmissions since less hops need to be traversed. However, it is worth noting the following. Even though both services lower the number of packet transmissions, the number of packet transmissions after negotiation (when both services are active at the same time in community A) is *higher* than when aggregation is the only active service. The reason is as follows: since less intermediate hops are used, the opportunities to aggregate information becomes more limited. As such, in situations where the only incentive is ‘obtaining a high network lifetime’, aggregation should not be activated together with packet sharing.

The average end-to-end delay is shown in Figure 11. The aggregation service temporarily stores information in buffers in order to aggregate multiple information exchanges. As a result, the delay increases significantly up to a (pre-configured) value of maximum 10 seconds. For this reason, aggregation is not activated in community B which requires low delay incentives. It is interesting that the delay caused by aggregation is larger when packet sharing is also activated, since less aggregation opportunities means that the

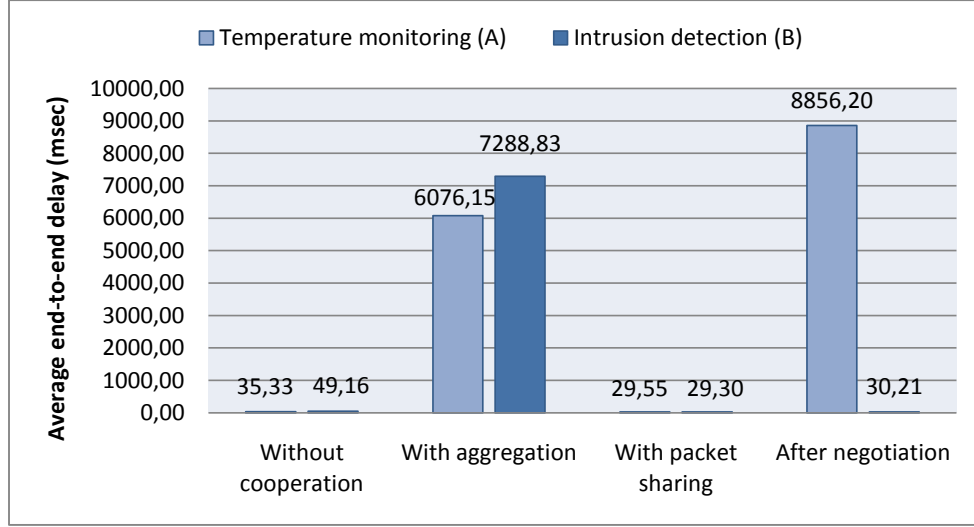


Figure 11: The average delay in communities A and B in the following situations: without cooperation between the communities, with packet sharing active in both communities, with aggregation active in both communities, and finally with optimal service selection after negotiation (packet sharing active in A and B, aggregation active in A)

information is stored longer before aggregated packets can be transmitted.

Finally, Figure 12 shows the average end-to-end reliability. Activating the aggregation service decreases the reliability by about 3%, since a single lost packet can now result in the loss of multiple information exchanges. In contrast, activating packet sharing increases the reliability by almost 10% since less intermediate packet transmissions are required. The drop in reliability that results from activating the aggregation service is offset in community A by activating the packet sharing service in both communities after negotiation (see Figure 12).

To summarize, the performance of both communities after negotiation is as follows. The packet transmissions and reliability of community A improve by respectively 7.7% and 2.4%. These improvements correspond closely to the requested distribution of the incentive weight factors from community A (that is, improving the network lifetime by twice as much as the reliability). The delay and reliability of community B improve by respectively 14.5% and 11.7%. This distribution matches closely the requested distribution of the incentive weight factors from community B (that is, equally improving the network lifetime and the reliability). As such, even when only

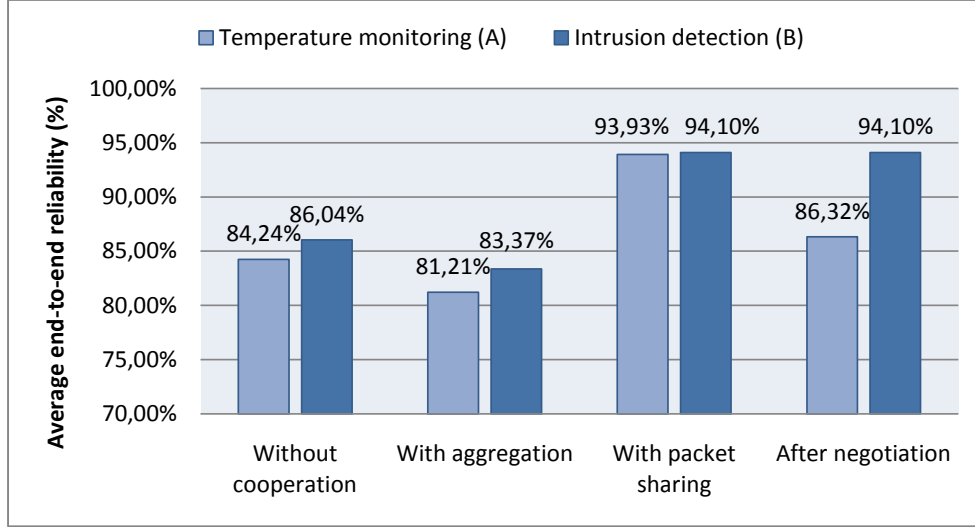


Figure 12: The average reliability in communities A and B in the following situations: without cooperation between the communities, with packet sharing active in both communities, with aggregation active in both communities, and finally with optimal service selection after negotiation (packet sharing active in A and B, aggregation active in A)

a limited amount of network services are available, communities can improve the performance of their incentives by negotiating and cooperating with each other.

4.6. Conclusion of the proof-of-concept

The lessons learned from this proof-of-concept are the following. The influence of network services can vary strongly, depending on the network topology and the used communication technologies [26]. In addition, the influence of network services also varies over time, especially in wireless environments [27]. Finally, networks services can behave differently when other network services are activated at the same time. These facts illustrate that accurate monitoring and policy enforcement solutions are needed to efficiently support dynamic network negotiation.

The implemented demonstrator serves to (i) demonstrate the feasibility of the discovery and negotiation strategies and to (ii) experimentally verify and measure the benefits of incentive driven cooperation. As shown in this section, incentive driven networking can result in better network performance for all participating devices. The gain in performance depends largely on the

network topology, the incentives of the devices and the available network services. As long as the influence rates give a correct indication of the influence of the network services, incentive driven networking will always result in a network performance that is at least as good as having different independent networks.

5. Related work

This section gives an overview of related network cooperation approaches that are designed for a closer collaboration between different wireless networks and discusses the differences with the presented approach. Figure 13 illustrates several of these related work approaches.

Network planning tools aim to optimize network criteria such as coverage or throughput by calculating the optimal placement and transmission power of devices. Network planning is very efficient in static and predictable network deployments [28, 29]. However, network planning solutions cannot be used in networks that (i) dynamically change network topology (such as ad-hoc networks), (ii) have network requirements that change over time, or (iii) in mobile environments, such as when portable devices such as PDAs, body-area-networks and laptops are frequently moved around. Even though network planning solutions are limited to static networks, planning tools can be used in combination with incentive driven network methodologies. For example, existing planning tools can be used to estimate the influence of network services on the incentives, which can be used as input for the negotiation phase.

The use of a *cognitive radio* [30, 31] enables devices to autonomously re-configure their transmission parameters based on the environment in which they operate. This allows the devices to reuse unused licensed spectrum without interfering with licensed users or to support an always best connected (ABC) paradigm [32]. When parameters of the higher network layers are optimized based on changes in the network environment, the term *cognitive networking* [33] is used. A cognitive network is capable of perceiving current network conditions and use this information to plan, learn, and act according to end-to-end goals [34]. Both cognitive approaches are focused on the optimization of a single protocol layer or a single device and do not usually involve negotiation or cooperation mechanics.

Whereas cognitive networking is designed for parameter optimization, in *cooperative networks* multiple devices work together towards reaching specific

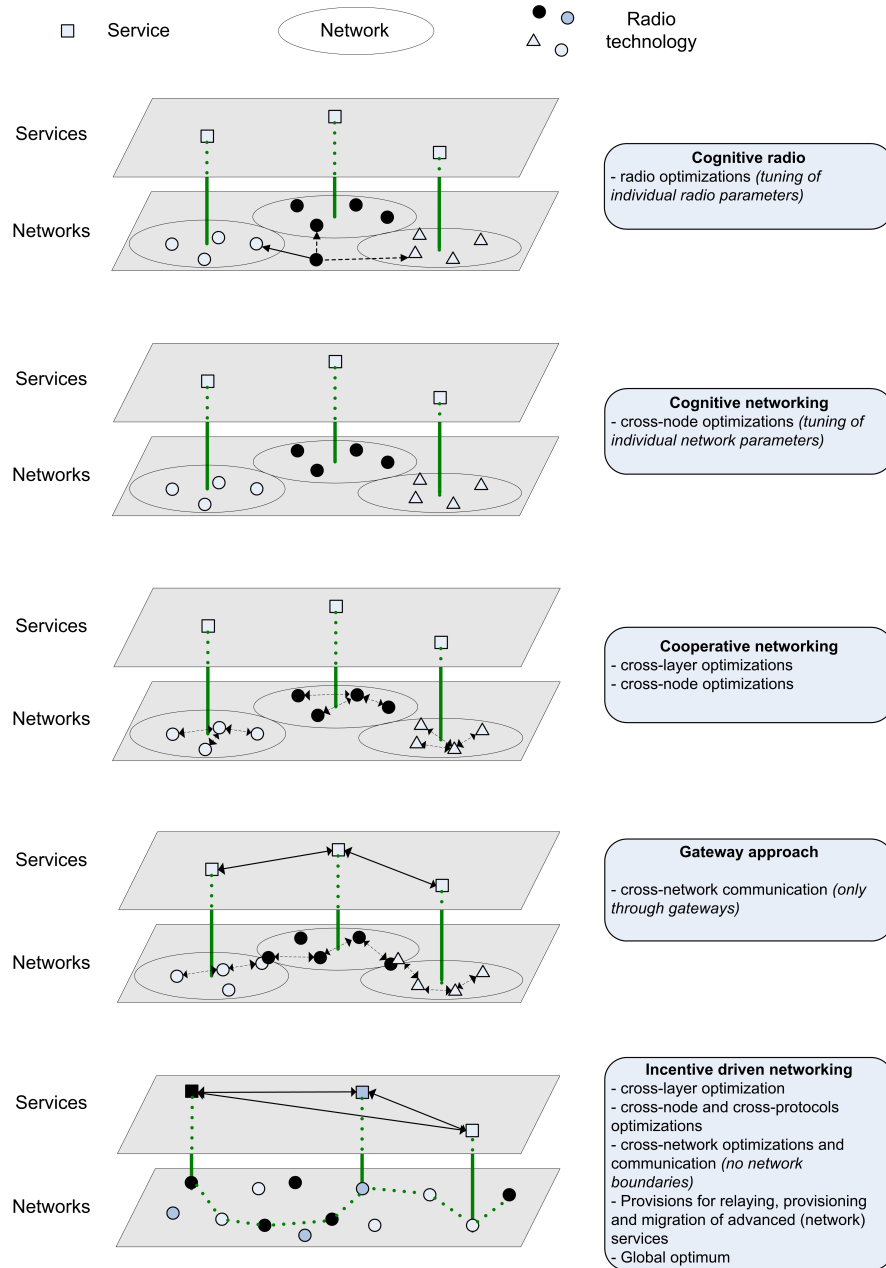


Figure 13: Comparison of incentive driven networking with related work approaches

goals. For example, in [35] two MAC protocols are presented that use a relay node to store the packets that failed transmissions in previous time slots and attempts to retransmit them in an empty time slot. Depending on the network layer that is optimized, different approaches are possible. However, barring some exceptions such as [36], most cooperation approaches have mainly considered one layer at a time.

Opportunistic or delay-tolerant networking [37] can occur when part of the infrastructure is not fixed but exists of mobile devices or exists in an environment in which devices often appear and disappear. Data exchanges can take place using the connection opportunities that arise due to impromptu encounters with other devices: nodes can forward data from the source to the destination by using connections with temporary neighbors. Opportunistic Networks enable users to communicate in disconnected environments, in which islands of connected devices appear, disappear, and reconfigure dynamically. Opportunistic networking can be regarded as a special form of cooperative networking.

Finally, *service oriented architectures* support automatic discovery of web services on newly discovered devices. Standards such as OWL-S [38] facilitate the automation of web service tasks including automated web service discovery, execution, interoperation, semantic descriptions, composition and execution monitoring. Unfortunately, these approaches are not designed to optimize networks at the lower network levels. However, incentive driven networking shares several concepts with service-oriented architectures (SOAs) [21].

- Service composability: network services can be combined to reach a specific goal (i.e: optimize the incentives of the communities).
- Standardized service contract: network services expose a standardized interface that is used to activate and configure them.
- Service loose coupling: network services can be activated independently and do not have any dependencies between each other.
- Service abstraction: network services hide their implementation logic, they only describe how they influence the incentives.
- Service reusability: network services can be reused for several topologies and communication technologies.

- Similar to SOAs the service broker from SOAs, a negotiation entity (see Section 3.3) decides which network services should be activated/deployed on which devices.

As such, the incentive driven approach could be extended to also include negotiation about high-level services.

In conclusion, even though many network optimization techniques exist, they typically have one of the following disadvantages: (i) optimizations must typically be included at the design phase and thus do not take into account dynamically changing environments, (ii) existing optimizations are often limited to optimizations in a single network layer and (iii) most solutions only allow interaction between different independent networks through translation gateways at fixed locations. In contrast, our approach (i) takes into account the incentives of each individual device, (ii) is not limited to a single network layer, (iii) is designed to cope with heterogeneous devices and (iv) can cross network boundaries that are traditionally fixed. For optimal network coexistence, our approach can be combined with some of the mentioned techniques that are complementary to ours.

6. Commercial opportunities

Finally, it is worth noting that incentive driven networking does not aim to replace all traditional networking solutions. Networks that have very stringent reliability and quality-of-service requirements should be deployed using accurate network planning tools in a static and reliable environment. However, our methodology offers a complimentary approach suited for application domains that cannot efficiently be solved using traditional network solutions. More specifically, applications that exhibit one or more of the following characteristics profit most from incentive driven networking.

1. *Multi-party applications* often benefit from negotiation. When devices or parties with different network requirements are co-located, the network performance of the individual parties can increase by making opportunistic use of each others resources.
2. *Dynamic applications*. That is, applications that deploy a variable number of devices, are set-up in an ad-hoc fashion without any planning, or that exhibit a variation in the number of users, traffic flows and traffic requirements.

3. *Heterogeneous networks.* Especially, multiple networks that are co-located, heterogeneous networks that consist of devices with different hardware characteristics and capabilities, and heterogeneous services with different incentives.
4. Applications that engage in, or benefit from, *multi-hop* behavior such as mesh networks, wireless sensor networks or ad-hoc networks.

Table 3 describes in more detail several example use cases that exhibit several of these characteristics.

Type of interaction	Example use cases	Characteristics
Ad-hoc relay between persons or objects	<i>Emergency services:</i> optimize the coverage of emergency networks (such as TETRA) by using collaboration (shared routing, power management, auto channel selection, etc.) between the ad-hoc networks from different emergency services present in a disaster area. (e.g. the fire brigade, the police, etc.), as well as by making opportunistic reuse of available deployed infrastructure networks such as corporate and consumer Wi-Fi access points.	<i>Multi-party:</i> yes (forced co-location) <i>Dynamic:</i> yes (# nodes, # users, ad-hoc) <i>Heterogeneous:</i> yes (multi-network, multi-technology, multi-service) <i>Multi-hop:</i> yes (ad-hoc)
	<i>Traffic jam:</i> improve network accessibility in densely populated situation such as traffic jam by having individuals connecting ad-hoc with each other in order to reach internet or mobile phone network.	<i>Multi-party:</i> yes (forced co-location) <i>Dynamic:</i> yes (# nodes, # users, ad-hoc) <i>Heterogeneous:</i> yes (multi-network) <i>Multi-hop:</i> yes (to circumvent crowded areas)

Coexistence of sensor networks	<i>Industrial environments:</i> allow easy configuration and growth of multiple sensor networks used to control machinery in a plant environment by dynamically optimizing radio and routing capabilities across these networks.	<i>Multi-party:</i> no <i>Dynamic:</i> yes (# nodes, # users) <i>Heterogeneous:</i> yes (multi-network, multi-service) <i>Multi-hop:</i> yes
Opportunistic offload of traffic	<i>Telecom offloading:</i> data and voice traffic from mobile network can be offloaded to the Wi-Fi Network when the user is in the range of a Wi-Fi access point. <i>BAN monitoring:</i> instead of mobile networks, available Wi-Fi networks can be used to backhaul medical information from body area networks (BAN) in order to reduce the transmission cost.	<i>Multi-party:</i> variable <i>Dynamic:</i> yes (# users, # traffic) <i>Heterogeneous:</i> yes (multi-network, multi-technology) <i>Multi-hop:</i> no <i>Multi-party:</i> yes (opportunistic) <i>Dynamic:</i> yes (# traffic) <i>Heterogeneous:</i> yes (multi-network, multi-technology) <i>Multi-hop:</i> no
Sharing of information and capabilities between vehicles	<i>Vehicle to vehicle:</i> allow the sharing of sensed information (e.g. ice detection, break detection, warning of accident) between vehicles in the near environment. Guarantee the reliability of transmissions between road equipment and vehicles by dynamically selecting the best radio configuration between the different vehicles and infrastructures.	<i>Multi-party:</i> yes (opportunistic, forced co-location) <i>Dynamic:</i> yes (# users, # traffic) <i>Heterogeneous:</i> yes (multi-service) <i>Multi-hop:</i> yes

Smart dynamic radio planning across services and buildings	<i>Home environment:</i> improve co-existence of the different wireless networks present in a home environment (alarm system, Wi-Fi, DECT, home automation, ...) by dynamically optimizing the radio settings. Allow collaboration and improve coexistence of the different wireless networks across flats or offices in the same building.	<i>Multi-party:</i> yes (forced co-location) <i>Dynamic:</i> yes (# users, # traffic) <i>Heterogeneous:</i> yes (multi-network, multi-technology, multi-service) <i>Multi-hop:</i> no
Temporary installation of multi-service networks	<i>Fairs and festivals:</i> allow easy setup of ad-hoc festival telecom infrastructure by minimizing interference and optimizing usage of resources and coverage. Dynamically optimization of Wi-Fi parameters (channel, power). <i>Construction site:</i> allow connection sharing across construction site infrastructure used for security (typically meshed wireless network with sensors and cameras) and communication devices	<i>Multi-party:</i> yes (forced co-location) <i>Dynamic:</i> yes (# users, # traffic) <i>Heterogeneous:</i> yes (multi-network, multi-technology, multi-service) <i>Multi-hop:</i> yes <i>Multi-party:</i> yes (forced co-location) <i>Dynamic:</i> yes (# nodes, # users) <i>Heterogeneous:</i> yes (multi-network, multi-technology, multi-service) <i>Multi-hop:</i> yes

Table 3: Example use cases that can benefit from incentive driven networking.

Based on this (non-exhaustive) overview, it is clear that incentive driven networking can be used in a wide range of applications.

7. Research opportunities

Finally, it is clear that each step of the proposed methodology can be custom-tailored towards a specific application domain. These custom-tailored implementations can lead to interesting research opportunities or patentable network algorithms:

- Efficient network discovery algorithms are needed that are capable of detecting multiple communication technologies.
- Network monitoring algorithms can be developed that are capable of estimating the (real-time) influence of cross-network services on the network performance.
- New negotiation approaches can be developed based on game-theory or machine learning for networks that have more complex network requirements.
- Heterogeneous network discovery can include methods for deducing and translating the network settings of neighboring communities, such as the type of MAC protocol, the structure of the supported packet types, the used routing protocols and the used communication settings.
- New dynamic addressing schemes can be designed that cope with dynamically created communities.

As such, our future work will focus on (i) optimizing the current algorithms to cope with more complex and dynamic networks, (ii) designing new negotiation methods that are suitable for devices that require several incentives to be satisfied at the same time and (iii) comparing the efficiency and suitability of different types of algorithms and solutions for each step of the methodology.

8. Conclusion

This paper introduced incentive driven networking: a cross-layer, cross-network networking approach that supports cooperation between heterogeneous networked devices. Incentive driven networking aims to (i) simplify the configuration and setup of networks for the end-users and (ii) increase the network performance of co-located devices. Rather than using manually

configured (and time-consuming) fixed network boundaries, network creation and negotiation is based on the concept of ‘network incentives’ or ‘device goals’. The methodology comprises the following steps:

- Devices cluster together with other devices that have similar incentives, thus forming communities of like-minded, interconnected objects.
- Different communities broadcast their existence to each other.
- The communities exchange profiles which describe their available network services, their incentives and their network settings.
- A negotiation entity determines the optimal set of network services so that each participating community benefits from cooperation.
- Finally, the selected services are activated, so that the incentives of each participating community are improved.

Depending on the device incentives, incentive driven networking results in better use of the scarce spectrum, better scalability, more efficient energy consumption, lower radio emissions, sharing of service capabilities (such as GPS, processing power or internet connectivity) and/or better QoS guarantees.

The proposed methodology was validated in the form of a proof-of-concept implementation. Experimental results showed that the main requirement for successful negotiation is the accurate estimation of the influence of network services on the incentives of co-located devices. However, as long as the influence rates give a correct (broad) indication of the influence of the network services, incentive driven networking will always result in a network performance that is at least as good as having different independent networks. Indeed, after cooperation, both networks in the proof-of-concept benefited from cooperation in the form of having better reliability, delay and/or longer network lifetime.

It is clear that the methodology is not limited to the example proof-of-concept scenario. In general, networks that exhibit one or more of the following characteristics are likely to profit from incentive driven networking: (i) multi-user networks, (ii) dynamic networks, (iii) heterogeneous networks, and/or (iv) multi-hop networks. To prove this fact, the paper gave an overview of a large number of potential marketable applications that can be implemented using our incentive driven cooperation methodology.

To conclude, the incentive driven networking paradigm is applicable to a wide range of applications domains and can ultimately lead to an improved coexistence of co-located networked devices.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreement n 257542 (CONSERN project), from the IWT SBO SymbioNets project and the FWO G.0243.10 and FWO G.0291.09 grants.

References

- [1] The internet of things, ITU Internet Reports (2005).
- [2] S. Buljore, H. Harada, S. Filin, P. Houze, K. Tsagkaris, O. Holland, K. Nolte, T. Farnham, V. Ivanov, Architecture and enablers for optimized radio resource usage in heterogeneous wireless access networks: The ieee 1900.4 working group, IEEE Communications Magazine.
- [3] A. Wheeler, Commercial applications of wireless sensor networks using zigbee, Communications Magazine, IEEE 45 (4) (2007) 70 –77. doi:10.1109/MCOM.2007.343615.
- [4] C. F. Garca-Hernandez, P. H. Ibargengoytia-Gonzalez, J. Garca-Hernandez, J. A. Prez-Daz., Wireless sensor networks and applications: a survey, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.3,.
- [5] N. Wakamiya, S. Arakawa, M. Murata, Self-organization based network architecture for new generation networks, First International Conference on Emerging Network Intelligence (Oct. 2009) pp.61–68.
- [6] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Communications and Mobile Computing, Special Issue: Mobile Ad Hoc Networking Research, Trends and Applications Vol. 2, issue 5 (August 2002) pp 483–502.
- [7] R. Musaloiu-E., A. Terzis, Minimising the effect of wifi interference in 802.15.4 wireless sensor networks, International Journal on Sensor Networks Vol. 3 (December 2008) pp. 43–54.

- [8] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, S. Singh, Exploiting heterogeneity in sensor networks, In Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2005).
- [9] A. Nash, W. Duane, C. Joseph, PKI: Implementing and Managing E-Security, McGraw-Hill, Inc., New York, NY, USA, 2001.
- [10] M. Toorani, A. Shirazi, Lpki - a lightweight public key infrastructure for the mobile environments, in: Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on, 2008, pp. 162–166. doi:10.1109/ICCS.2008.4737164.
- [11] M. B. J. Feigenbaum, J. Lacy, Decentralized trust management, Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos, available as a DIMACS Technical Report from <ftp://dimacs.rutgers.edu/pub/dimacs/TechnicalReports/TechReports/-1996/96-17.ps.gz> (1996) Pages 164–173.
- [12] OpenPGP Message Format (IETF standard), <http://tools.ietf.org/html/rfc4880>.
- [13] Device description structures, <http://www.w3.org/TR/dd-structures/>.
- [14] T. Yucek, H. Arslan, A survey of spectrum sensing algorithms for cognitive radio applications, IEEE Communications Surveys & Tutorials Vol. 11 (First Quarter 2009) pp.116–130.
- [15] E. De Poorter, S. Bouckaert, I. Moerman, P. Demeester, Non-intrusive aggregation in wireless sensor networks, Ad Hoc Networks Vol 9. Issue 3 (May 2011) pp. 324–340. doi:DOI: 10.1016/j.adhoc.2010.07.017.
- [16] L. Dasilva, H. Bogucka, A. Mackenzie, Game theory in wireless networks, Communications Magazine, IEEE 49 (8) (2011) 110–111. doi:10.1109/MCOM.2011.5978423.
- [17] A. MacKenzie, S. Wicker, Game theory and the design of self-configuring, adaptive wireless networks, Communications Magazine, IEEE 39 (11) (2001) 126–131. doi:10.1109/35.965370.

- [18] M. Lee, D. Marconett, X. Ye, S. J. B. Yoo, Cognitive network management with reinforcement learning for wireless mesh networks, in: Proceedings of the 7th IEEE international conference on IP operations and management, IPOM'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 168–179.
URL <http://dl.acm.org/citation.cfm?id=1775321.1775341>
- [19] E. De Poorter, E. Troubleyn, I. Moerman, P. Demeester, IDRA: a Flexible System Architecture for Next-Generation Wireless Sensor Networks, *Wireless Networks* (Springer) Vol. 17 (6), 24 May 2011.
- [20] D. Hughes, K. Thoelen, W. Horr, N. M. et al., Looci: a loosely-coupled component infrastructure for networked embedded systems, Proceedings of the 7th International Conference on Advances in Mobile Computing & Multimedia, Kuala Lumpur, Malaysia (14-16 December 2009) pages 195–203.
- [21] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.
- [22] Tmotesky datasheet, <http://www.snm.ethz.ch/projects/tmotesky>.
- [23] The IBBT W-iLab.t wireless sensor testbed. <http://ilabt.ibbt.be/>.
- [24] L. Tytgat, B. Jooris, P. De Mil, B. Latr, I. Moerman, P. Demeester, Demo abstract: Wilab, a real-life wireless sensor testbed with environment emulation, published in European conference on Wireless Sensor Networks, EWSN adjunct poster proceedings (EWSN), Cork, Ireland.
- [25] Ad hoc on-demand distance vector (aodv) routing. networking group request for comments (rfc): 3561, <http://tools.ietf.org/html/rfc3561> (July 2003).
- [26] M. Wirz, Btnode application for automated link measurements, term thesis, Master's thesis, ETH Zurich (2007).
- [27] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, S. Wicker, Complex behavior at scale: An experimental study of low-power wireless sensor networks, *Wireless Sensor Networks*. Technical Report CSD-TR 02-0013, UCLA, February 2002.

- [28] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, S.-Y. Kung, Network planning in wireless ad hoc networks: a cross-layer approach, *Selected Areas in Communications, IEEE Journal on*, 2005 23 (1) 136 – 150. doi:10.1109/JSAC.2004.837362(410) 23.
- [29] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, S.-Y. Kung, Network planning in wireless ad hoc networks: a cross-layer approach, *IEEE Journal on Selected Areas in Communications*, 2005 Vol. 23 pp. 136 150.
- [30] S. Haykin, Cognitive radio: brain-empowered wireless communications, *Selected Areas in Communications, IEEE Journal on* 23 (2) (2005) 201 – 220. doi:10.1109/JSAC.2004.839380.
- [31] F. K. Jondral, Software-defined radio - basics and evolution to cognitive radio, *EURASIP Journal on Wireless Communications and Networking* vol. 2005, no. 3 (2005) pp. 275283.
- [32] I. Akyildiz, W. Lee, M. Vuran, S. Mohanty, Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey, *Elsevier Computer Networks* Vol 50 (2006) pp. 2127–2159.
- [33] R. W. Thomas, D. H. Friend, L. A. Dasilva, A. B. Mackenzie, Cognitive networks: adaptation and learning to achieve end-to-end performance objectives, *Communications Magazine, IEEE* 44 (12) (2006) 51 –57. doi:10.1109/MCOM.2006.273099.
- [34] C. Fortuna, M. Mohorcic, Trends in the development of communication networks: Cognitive networks, *Computer Networks*, 2009.
- [35] A. K. Sadek, K. Liu, A. Ephremides, Collaborative multiple-access protocols for wireless networks, *ICC* 2006.
- [36] J. Garca-Vidal, M. Guerrero-Zapata, J. Morillo, D. Fust, A protocol stack for cooperative wireless networks, *Wireless Systems and Mobility in Next Generation Internet*, In *Lecture Notes in Computer Science (LNCS)* Volume 4396 (2007) pp 62–73.
- [37] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, *Communications Magazine, IEEE* vol.44, no.11 (November 2006) pp.134–141.

- [38] Owl-s: Semantic markup standard for web services,
<http://www.w3.org/submission/owl-s/>.