Open access • Journal Article • DOI:10.1007/S11235-011-9470-Z

Network virtualization as an integrated solution for emergency communication

Peter Dedecker, Jeroen Hoebeke, Ingrid Moerman, Joris Moreau ...+1 more authors

Institutions: Hogeschool Gent, Ghent University

Published on: 01 Apr 2013 - Telecommunication Systems (Springer US)

Topics: Network virtualization, Dynamic circuit network, Networking hardware, Active networking and Delay-tolerant networking

Telecommunication Systems Journal manuscript No.
(will be inserted by the editor)

Network virtualization as an integrated solution for emergency communication

Peter Dedecker[⋆] [⋆⋆] · **Jeroen Hoebeke**[⋆] · **Ingrid Moerman**[⋆] · **Joris Moreau**[⋆] [⋆⋆] · **Piet Demeester**[⋆]

Received: 10-3-2010 / Accepted: 10-3-2010

Abstract In this paper the Virtual Private Ad Hoc Networking (VPAN) platform is introduced as an integrated networking solution for many applications that require secure transparent continuous connectivity using heterogeneous devices and network technologies. This is done by creating a virtual logical self-organizing network on top of existing network technologies reducing complexity and maintaining session continuity right from the start. One of the most interesting applications relies in the field of emergency communication with its specific needs which will be discussed in this paper and matched in detail against the architecture and features of the VPAN platform. The concept and dynamics are demonstrated and evaluated with measurements done on real hardware.

Keywords virtualization · emergency communication · ad-hoc · mobile · networking · wireless · experimentation

CR Subject Classification C.2.1
Distributed networks, Network topology, Wireless communication

1 Introduction

For National Security and Public Safety (NSPS) workers, a special purpose communication network is necessary with its special requirements. For emergency workers, a permanent connection enabling voice and data communication, is a prerequisite for a successful mission. The Internet, which is nowadays becoming a large "network of networks" with its broad spectrum of different wired, wireless and mobile communications and technologies, offers ubiquitous connectivity. Nevertheless, it may be clear that this large-scale communication network is not directly suitable for emergency applications as the secure interconnection of a selected set of distributed and often mobile devices over this network requires technological skills and very precious configuration and management time. Reducing this configuration and management complexity must be a key feature of our communication platform.

In this paper we will first highlight in section 2 some key features that NSPS-communication systems and emergency applications require and illustrate this with a simple scenario. In 3, an overview of the most important related work in emergency communication as well as overlay networking and other enabling protocols is given revealing there are no integrated solutions that tackle all requirements at the same time. Our platform, based on network virtualization, provides this integrated solution and is briefly presented in section 4. To validate this architecture, a proof-of-concept implementation has been built of which a summary is given in section 5. This implementation is evaluated with some performance measurements on real hardware in section 6. A critical evaluation of these results and an evaluation against the requirements is given in section 7, next to some primary points of interest for future work.

2 Emergency communication requirements

In emergency situations, time is crucial. Especially during the golden hour, every gained minute can save many lives.

[⋆]Ghent University - IBBT - IBCN
Department of Information Technology (INTEC)
Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium
Tel.: +32-9-3314900
Fax: +32-9-3314899
E-mail: Name.Surname@intec.UGent.be

[⋆⋆] University College Ghent
Department of Applied Engineering Sciences
Schoonmeersstraat 52, 9000 Gent, Belgium
Tel.: +32-9-3314977
Fax: +32-9-2649969
E-mail: name.surname@hogent.be

So efficient communication and data exchange is of great importance. A continuous communication platform providing voice and data allows emergency workers to agree on intervention strategies while on the road, thereby viewing intervention plans and updated stock lists of eventual dangerous goods on a mobile device. Permanent updates of a dynamic geographical information system (GIS) and crisis management systems as developed in the GeoBIPS [3] and ADAMO [2] projects enrich this data exchange and provide the authorities with concise and exact information in real-time. Support for non-interactive data communications is a basic requirement also requested by the SAFECOM program [11] of the US Dept. of Homeland Security.

Upon arrival on site, this communication platform must allow very fast deployment to all individual emergency workers, providing instant ad-hoc usage. No time may be lost setting up complex infrastructures. We cannot expect emergency workers to have any technical knowledge, nor do they have time to handle network configuration on their devices. Their only focus is the intervention itself. Therefore, ease-of-use is of prime importance.

A first group is sent out to explore the site. Permanent reliable communication with this group, as well as other team members on-site, stays crucial. Our network, growing in space as team members get scattered on location, must be able to handle mobility without manual device configurations. Self-organization, self-maintenance, self-optimization and self-healing capabilities are necessary. When coverage gets lost, additional communication equipment must be setup, again without technical knowledge or user administration. Also here, communication and data exchange is important to provide other team members and especially the commanding officer with a good view of the situation. Streaming video capabilities can be of great help. The fast setup of this additional deployed infrastructure is even more important when, like with the hurricane Katrina, lots of wireless base stations, cables and central offices are destroyed and one can only rely on the ad hoc network.

Over time, the needs of the emergency network evolve. In the very first phase, rapid deployment, using highly portable, small, lightweight equipment by non-technical users is a key requirement. In this phase most communication is voice-based between the end-user and the headquarters (HQ), while also initial access to the emergency management application is necessary. However, in a second phase, cooperation gains importance as other teams get involved. Commanding officers (CO) need permanent communication lines with their colleagues as well as their team members, without both groups getting mixed or overwhelmed with information that was not destined for them. Different fenced but inter-operable communication groups are necessary. [36] recalls the importance of these interoperability requirements as illustrated in previous events like the 9/11 attacks where radio communications were a lifeline for the hundreds of police officers who received the word to evacuate the building and all but 60 of them escaped. Tragically, hundreds of New York firefighters did not receive that warning because they were using a different radio communications system, with the well known consequences.

Increased activities demand for increased access to the emergency management application and thus a higher bandwidth [11]. Todays private mobile radio (PMR) networks like TETRA [42] can provide adequate service when there are a few users, but when there are many, these applications can block all other communication. Therefore we must be able to use and combine different available network technologies, like WiFi, GPRS, UMTS, HSPA, (mobile) WiMAX, satellite, LTE [19] or even an Ethernet or DSL connection from a nearby building. Compatibility with upcoming technologies like LTE Advanced must be kept in mind.

Of course, the emergency communication platform must be flexible and future proof in order to support different kinds of applications, like voice, video, document sharing, etc. No restrictions on the application level are allowed. This also counts for the used devices where a great heterogeneity must be taken into account. Regular computers must be supported as well as laptops, tablet PCs, smartphones, servers, etc.

Last but not least: security is a key requirement. Emergency communication should be shielded from other applications using the same shared medium. Not only to reduce configuration complexity and information overwhelming to other present teams, but in the first place to protect the intervention and all involved parties from intrusion or information leakage. Eavesdropping paparazzi or sabotage through the used communication platform must be made impossible.

From the above description, we can summarize the following basic requirements:

- Membership configuration and management: only certain people (further called members) may have access to the provided emergency services
- Ease of use and fast ad hoc deployment: reduce technical complexity to avoid losing time. Assume users may not have technical knowledge and have other priorities.
- Local and distributed operation: devices in each other's neighborhood should be able to communicate securely without infrastructure. Using the Internet and heterogeneous access technologies, distributed members should be automatically and securely interconnected.
- Security: access to the network based on membership, application and service access rights and secure data transport. As such, security mechanisms for trust, encryption, authentication and authorization are indispensable.
- Self-organization and mobility management: connectivity and session continuity should be guaranteed regard-

less of member mobility, topology changes or access network changes.
- Application support: users (or network managers) should be able to specify to which applications, services, data... the users in the NSPS network have access. Additionally, support for all kinds of applications is required.
- Scalability: depending on the scenario, the number of members can become quite large, making scalability a potential issue that has to be taken care of.
- Heterogeneity: a wide spectrum of devices should be supported, as well as different access networks and technologies

3 Related work

A lot of research has already been done in the area of NSPS networking and will be matched here against our requirements.

3.1 TETRA, TEDS and MVNO

Emergency servants in lots of countries use their own TETRA-based network. This TErrestrial Trunked RAdio [42] system (standard proposed by ETSI) is a digital solution built upon a private mobile radio (PMR) system. TETRA provides specialized safety services, direct mode operation, voice encryption, all informed net group calls and a wide range of data services over a wide area. However, even TETRA 2 release with its TETRA Enhanced Data Services (TEDS) has a very limited (downlink) bandwidth theoretically ranging from 15.6 kbps (DQPSK modulation over a 25 kHz channel) up to 538 kbps using 64-QAM over a 150 kHz channel. This prevents enriched information streams such as live video from inside a crisis situation or even frequent database consulting. Also the private aspect of the system prevents coupling with public networks.

Creating a secure mobile virtual network operator (MVNO) using todays 3GPP-based mobile broadband networks is an alternative to TETRA promoted by operators and technology companies like Ericsson in white papers like [23], primary focused on the higher bandwidth and general (IP-) compatibility. However, other requirements as listed in section 2, like local ad hoc optimizations, are not met. Also this infrastructure can be demolished in certain situations, so we really need fast deployment of own local infrastructures. However, when available, it must be possible to use this network as one possible access network to support emergency communication.

3.2 (Proxy) Mobile IPv6 - Host Identity Protocol - NEMO

In [29], Iapichino et al. propose a combination of Proxy Mobile IPv6 (PMIPv6) 3.2 and Host Identity Protocol (HIP) in order to combine mobility and heterogeneous networking for emergency management. This is based on their previously proposed unmodified IPv6 approach [30] with locally deployed mobile ad hoc mesh networks interconnected by satellite using vehicle communication gateways. This very interesting architecture allows public safety users to use the different technologies of their multi-homed devices while supporting session continuity. Intra and inter-technology handovers are calculated to be very efficient occurring low latency.

However, other crucial requirements are not met. No trusted secure environment is created. There is no possibility of grouping a selected set of devices in a shielded environment but still allowing other NSPS organizations to use the communication infrastructure as base layer without getting mixed up.

In that respect it is also worth to mention other mobility management related work such as Mobile IP [39,31] and NEMO. NEMO [4] is a solution concerned with managing the mobility of an entire network that changes its point of attachment to the Internet in a way that is completely transparent to the nodes inside the mobile network. Issues such as route optimizations and security are also considered. These solutions provide interesting mechanisms to manage mobility of nodes and networks with respect to all other nodes in the Internet and concepts could be applied to the management of the mobility of distributed members in an integrated emergency communication framework. Of course, it is clear that the scope of these solutions is also far too limiting with regard to aspects such as membership management, ad hoc routing, infrastructure-less communication, providing a shielded environment for network members.

3.3 WiMAX backbone and grid computing

In [15], Chiti et al. present a wireless infrastructure based on a WiMAX interconnecting backbone and local wireless ad-hoc (IEEE 802.11x) and sensor networks for coverage extension and monitoring purposes. This architecture is especially interesting for connectivity roll-out in and monitoring of bigger areas without (or with destroyed) infrastructure avoiding satellite-like round trip times. Some issues (like the need for distributed authentication) are identified. Fantacci et al. go a step further in [24] integrating the (same) communication infrastructure and the processing layer as a grid, proposing the ASSISTANT programming model.

While [15] focuses mostly on the establishment of a local communication infrastructure, we think the roll-out of a

secure network for emergency workers offering connectivity to all involved actors must be the main issue.

3.4 Overlay-like networking techniques

Realization of secure communication between all involved actors in an emergency setting has similarities with the creation of logical groupings of devices. Therefore we discuss here some networking techniques that allow the creation of such logical groups and match them against the requirements.

- VLANs [25]: a virtual LAN is the logical grouping of a subset of devices belonging to an Ethernet system which appear to be on a separate LAN.
- VPNs [10,7,32]: a VPN is a private data network that makes use of an underlying communication network (eg. the Internet) to securely connect two sites or add a (mobile) client to a site. An interesting solution is offered by Hamachi [1]: a centrally managed solution for direct peer-to-peer VPNs with an open security architecture and NAT-to-NAT traversal capabilities.
- P2P application overlays [22,34]: in P2P application level overlays, applications running on distributed systems create logical links between each other using native Internet routing and standard IP addresses. The result is a self-organizing semantic layer above the basic transport protocol level.

These solutions only tackle one specific aspect, but fail to tackle other requirements. VLANs are limited to one Ethernet environment, while VPNs are too static for dynamic environments which emergency scenarios definitely are. VPNs may be set up from every location, but there is no mobility management, no session continuity and some solutions even put limitations on the application level. There is also no local ad hoc organization possible. P2P-overlays may be more self-organizing but these only run on top of an operational network and are limited to the application and its services: the overlay cannot be used by other applications. It may be clear that the above techniques all have their limitations and do not meet the requirements as stated in 2.

3.5 Conclusion

This state-of-the-art overview reveals that, although existing solutions partially can meet the requirements, none of them is capable to tackle all requirements. Simply combining a number of these solutions is also too limiting in order to meet all requirements. This motivates the design of an integrated solution that tackles all these requirements and will be subject of the remainder of this paper.

4 VPAN architecture as an integrated solution

As stated in the introduction, the Internet is not directly suitable for emergency applications due to its complexity to securely interconnect distributed groups of devices. However, emergency scenarios do not need this complex Internet as such. As discussed in the requirements section, they just need a permanent secure connection between a dynamic subset of distributed and mobile devices with a limited number of crucial applications, using the available (heterogeneous) infrastructure, appended with their own infrastructure.

Therefore, we see interesting opportunities in the expected evolution towards network virtualization, where a logical structure is built on top of these base networks [14, 13]. Such small and secure logical overlay networks, grouping the previously mentioned subset of devices and making the underlying base network invisible, create a shielded and trusted environment for their participants. Enhancing them with ad hoc protocols and techniques can result in self-creating, self-organizing and self-administering communities on top of existing network infrastructures, drastically reducing complexity so their users can focus on their main task: the emergency intervention.

The Virtual Private Ad Hoc Networking (VPAN) platform as described in [27] and [26] is developed with these key insights as main drive. It is based on the creation of virtual overlay networks consisting of selected subsets of permanently connected trusted devices. Nodes in an overlay network or VPAN can be thought of as being connected by virtual or logical links. These virtual links correspond to a path in the underlying network, perhaps through multiple physical links. Such an overlay network drastically reduces size (in terms of connected devices) and complexity. Participating devices can join multiple overlay networks. In each VPAN, a node can share selected services or resources. A graphical illustration of the VPAN concept is given in Fig. 1 and is further explained in the following paragraphs.

All devices participating in an overlay network must share a common cryptographic trust relationship which must be installed in advance or can be added on the fly.

Based on this trust relationship, neighboring devices (i.e. devices having link layer connectivity) can discover and authenticate each other and establish subsequent short-term link-level security associations. This process is called neighbor detection and is illustrated in Fig. 2.a. All members that only rely on trusted nodes for their communication form a cluster. Internally, this cluster uses private VPAN IP addresses and ad hoc routing over the secured links, resulting in a fully secured and self-organizing network, potentially using different heterogeneous networking technologies (WiFi, Bluetooth, Ethernet...).

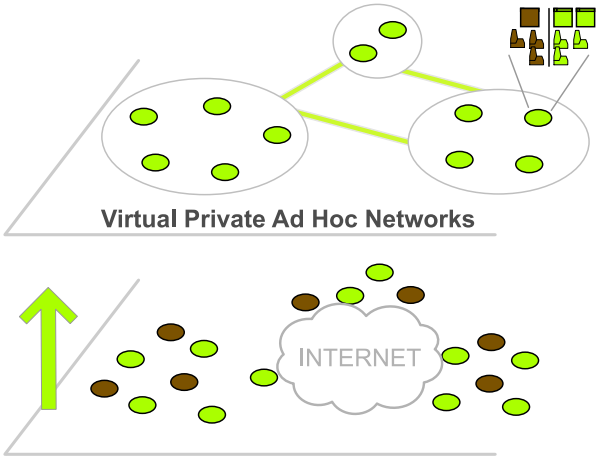

Fig. 1 The VPAN concept.

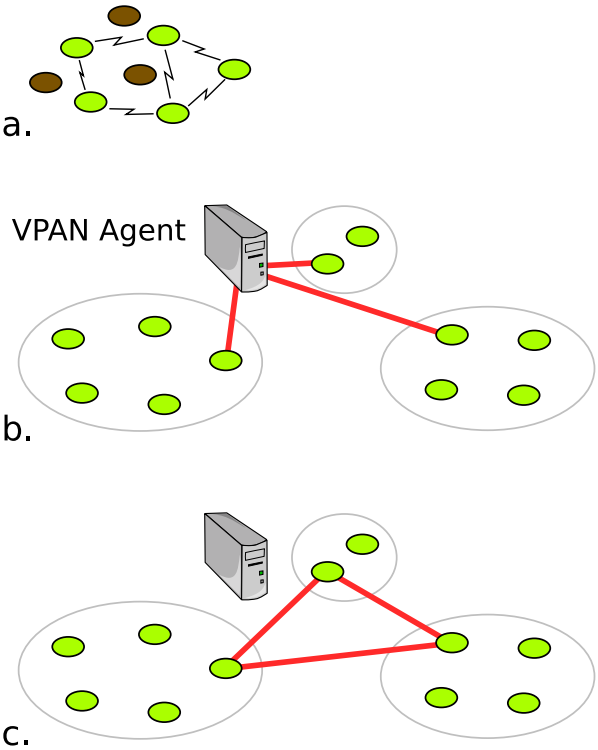

Fig. 2 VPAN creation: a. Neighbor discovery; b. Cluster registration; c. Tunnel negotiation

Next, clusters on different geographical locations discover each other's point(s) of attachment to the Internet with the help of a VPAN Agent (see Fig. 2.b). This information is then used to dynamically establish secure tunnels between the clusters, again including authentication of the gateway nodes between which the tunnels are established (see Fig. 2.c). Again the underlying technology does not matter, as long as it is IP capable (WiFi, WiMax, TETRA, UMTS, GPRS, HSXPA, LTE,...).

At this point, any member in the VPAN network can securely communicate with any other member using the internally fixed private IP addresses. Mechanisms to handle member mobility, changes in network interfaces, link breaks, changes in the point of attachment to the Internet... ensure that connectivity is maintained between all VPAN members in a way that is transparent to the applications running in the VPAN network. Also cluster-wide and VPAN-wide broadcasting is supported. A service announcement and discovery protocol enables service sharing, discovery and usage. Finally, API's are provided to monitor the network and retrieve status information.

To summarize the aspect of virtualization, each physical node that is member of a VPAN becomes available in the resulting virtual network as one virtual node. Physical (L2) links between trusted nodes become virtual direct (secured) links. Virtual tunnels at their turn can be mapped on a L3 path consisting out of multiple physical links and physical (non-trusted) nodes, between two physical IP endpoints. On each physical node which is a VPAN member, a virtual IP adapter is created. All communication through this adapter is encrypted, translated and sent into the virtual network as with traditional VPN solutions.

This integrated approach makes the platform generally usable in a wide range of applications like healthcare [28], machine-to-machine communication [9], personal networking grouping all your devices, education, public transport company networking, enterprise networking grouping collaborating people etc. This paper however focuses in detail on its implementation and its application in emergency networking and its performance in these highly dynamic environments.

5 VPAN implementation summary

In this section the building blocks are explained while focusing on their functionality and their cooperation in the framework. The used protocols are not treated in detail but further references are given.

5.1 Establishment of a Trust Relationship

A Virtual Private Ad Hoc Network is established between VPAN members that have a common trust relationship. This common trust relationship is established through the use of certificates. Every VPAN member has a private key, a member certificate (a public key signed with a private key common to the VPAN) and a common VPAN certificate. Also, every VPAN member has a unique VPAN ID. Finally, all VPAN members have a signed common VPAN profile that contains general information about the VPAN (such as the VPAN IP addressing space, VPAN Agent IP...) and a signed member profile that contains member specific settings such

as the device name, user name.... At this moment, the generation and installation of the certificates and profiles can be done through a web portal, but of course this can also be done remotely by a network administrator.

5.2 Neighbor discovery

VPAN members sharing such a common trust relationship that are able to connect to each other without using not-trusted nodes, organize themselves in clusters during a cluster formation process. Therefore, beacons are periodically broadcasted over all Personal Area Network (PAN) interfaces, interfaces that are used for establishing direct connections with neighboring VPAN members. Upon reception of a beacon of another VPAN member, a three-way challenge-response session takes place, resulting in mutual authentication using their member certificates and private keys, storage of all link information and the exchange of short-term unicast and broadcast session keys. These short-term keys are then used to encrypt all further communication using symmetric encryption, resulting in a secure communication link. The link information makes the use of ARP messages redundant. For the creation of the certificates and the implementation of the security protocols, the OpenSSL library [44, 17] has been used. In order to enable network connectivity between the IP-capable VPAN members within the same cluster, address assignment and routing capabilities are needed that can deal with these ad hoc characteristics.

5.3 Addressing

We have adopted a flat addressing scheme, in which each VPAN member receives one unique VPAN IP address, independent of the number of underlying network interfaces. This address consists of a VPAN prefix, which is part of the common VPAN profile, followed by a Node Identifier that uniquely identifies each VPAN member within the VPAN. Address assignment is realized through the use of an automatic address assignment protocol. Initially, the address is generated automatically by the VPAN member. A duplicate address detection mechanism ensures that the generated address is unique within an isolated cluster. When the cluster is able to communicate with the VPAN Agent (see further), the generated addresses are verified by the VPAN Agent for uniqueness within the entire VPAN. Once the addresses have been verified centrally, in the VPAN Agent for their uniqueness, the same (or another in case of a duplicate address) address will be reused for the remainder of the lifetime of the VPAN member in the VPAN, avoiding additional addressing overhead. In addition, when clusters merge or split, no address reassignments or complex duplicate address detection procedures are needed. As a consequence, this addressing scheme has the advantage that it incurs only a one-time overhead until verification has taken place.

5.4 Intra-cluster Routing

As a flat addressing scheme does not reveal any information on the location of the node within the VPAN, this problem has to be solved by the proposed VPAN-wide routing protocol. Since a VPAN is inherently a hierarchical network, i.e. groups of nodes forming clusters that are interconnected over the Internet, we exploited this by developing a hierarchical VPAN routing protocol, separating intra-cluster and inter-cluster routing. As clusters are self-organizing and self-maintaining, properties that can also be found in ad hoc routing protocols, it is a natural choice to integrate an ad hoc routing protocol into the architecture to fulfill the intra-cluster routing functionality. We have chosen to adopt a flexible multi-mode approach for the intra-cluster routing component in our solution, offering both proactive and reactive intra-cluster routing. The desired type of routing can be defined in the common VPAN profile.

5.5 Dynamic Tunneling

After cluster formation, the clusters can have access to the infrastructure through their VPAN gateway nodes, locate each other and establish secure tunnels between each other. The functionality for finding the location of the other remote clusters is offered by a VPAN Agent. Currently, we assume that the VPAN Agent is a centralized server that is always online and reachable through its public IP address or DNS name. VPAN gateway nodes securely register with their VPAN Agent to which they will establish a tunnel, revealing their physical location. At the same time, the presence and type of Network Address (Port) Translation (NA(P)T) boxes is discovered. This information is communicated to all other VPAN gateway nodes that have registered, providing them with an up-to-date view of all other remote clusters, and is then used to proactively, i.e. immediately after registration of a new VPAN gateway node, or reactively, i.e. when communication with a VPAN member in a remote cluster is needed, establish tunnels. As such, the establishment of these tunnels can take place proactively or on demand. Again, this is defined in the common VPAN profile. This results in a virtual overlay network that encompasses all VPAN members. By default, IPSec tunneling, applying both encryption and encapsulation, is used. However, in case one or multiple VPAN gateway nodes are behind NA(P)T, IPSec over UDP is used. In case both gateway nodes are behind NA(P)T, hole punching is used to go through the NA(P)T boxes. If no direct tunnel can be established, traffic is tunneled via the VPAN Agent.

5.6 Inter-cluster Routing

Similar to the multi-mode solution for intra-cluster routing, we have adopted a multi-mode approach for the inter-cluster routing component of the VPAN routing protocol, allowing both proactive and reactive inter-cluster routing. The desired routing and tunneling types are defined in the common VPAN profile. Further, since inter-cluster forwarding takes place over tunnels, the inter-cluster routing protocol does not make use of next hop information as traditional routing protocols do, but uses tunnel identifiers. In addition, a gateway discovery and selection mechanism (see further) is implemented. Finally, next to unicast functionality, 1-hop, cluster-wide and VPAN-wide broadcasting functionality is offered.

5.7 Universal Convergence Layer and Interface Management

We already stated that, at the network level, a VPAN is an overlay network in which every VPAN member has a unique VPAN IP address independent of the number of interfaces and independent of any other IP addresses that are already assigned to these interfaces. Apart from the fact that a device can have multiple interfaces, we also need to discriminate between two types of interfaces, namely Personal Area Network (PAN) interfaces and Wide Area Network (WAN) interfaces. PAN interfaces are interfaces that are used for direct communication with neighboring VPAN members (e.g. WLAN, Bluetooth...). WAN interfaces are interfaces that are used to connect to a public network like the Internet (e.g. Ethernet, WLAN, UMTS...). Of course, some types of interfaces (e.g. WLAN) can be used at the same time as PAN and WAN interface. Also, interfaces can be added or removed dynamically for use by a VPAN member. To this end, a universal convergence layer (UCL) [41] is needed that manages the complexity, characteristics and heterogeneity of the underlying interfaces and the dynamic addition and removal of them, thereby providing a uniform view and interface to the higher layers.

Further, the status of an interface, such as (wired or wireless) connectivity to an access point or the (re)assignment of a (public) IP address, can change during the use of the interface and has an impact on other VPAN components (for instance the management of mobility, optimization of link break detection...). Therefore, an event-based interface manager listens to NETLINK events sent out by the kernel, indicating changes in the interfaces or routing tables, and handles them to maintain the VPAN overlay. As such, any dynamics taking place within the VPAN can be taken care of, thereby guaranteeing the VPAN connectivity. This also speeds up some processes, like the neighbor detection where an additional beacon is sent after a new interface comes up, or removal of a link when a cable is unplugged. Finally, interfaces have different properties in terms of bandwidth, cost, security... which are taken into account in order to optimize the VPAN communication according to user preferences.

5.8 Gateway discovery

As a cluster can consist of multiple gateways, an intelligent choice should be made selecting the right gateway. Even more: gateways can have multiple WAN-interfaces with all different characteristics like bandwidth, delay, cost, QoS, energy usages,... Therefore a gateway discovery mechanism has been implemented. When a node joins the cluster and the proactive routing protocol is used, all gateways announce their availability and the characteristics of their available WAN-interfaces in the cluster. When hopcount is the used metric, packets for nodes in a remote cluster are sent to the next hop on the path towards the nearest gateway. When other metrics, like cost or energy, are used, the packet is encapsulated in another IP-packet with the selected gateway as destination. Upon arrival at the gateway, the packet is decapsulated and sent to the remote gateway. As gateway selection is done by the individual nodes, encapsulation is necessary to prevent bouncing when two nodes select another gateway.

5.9 Integrated security mechanisms: summary

As stated in the definition of a VPAN in section 4, security is a key element of the platform, integrated in all components and mechanisms. Using a public key infrastructure (PKI) every node is equipped with an X.509 certificate and a private key and the public key of the certification authority (CA) for this particular VPAN. All certificates are signed using the private key of the CA. In the current implementation, a key length of 1024 bits is used. This allows all nodes to authenticate each other and establish temporary secure communication channels over which symmetric pairwise keys are exchanged. This approach is applied to the neighbor detection process and the tunnel establishment process as described in 5.2 and 9. Next, all data traffic being sent in the overlay is encrypted using the much faster symmetric cryptography. In our implementation, a 160-bit SHA-1 digest and replay counter are added, after which the whole packet is encrypted using a 128-bit AES-key. This is all done using the well-known OpenSSL library.

5.10 Implementation

Currently, a proof-of-concept version of the VPAN-software has been built for different operating systems including Linux, Mac OS X and Windows as well as Maemo, which is a

Linux variant for the Nokia Internet tablets. Also a version for the Neo Freerunner smartphone is available.

The implementation of the above described building blocks is done in the Click Modular Router framework [33]. Click is a software architecture for building flexible and configurable routers, but can be used for implementing any network level packet processing functionality. Next to this, a rudimentary GUI and a complete local and remote API are developed allowing status and statistics retrieval as well as full configurability of the running overlays.

The software has been deployed and tested on a testbed consisting of multiple clusters, multiple overlay networks, access points and different access networks for a wide range of setups and parameter values. Details can be found in [26]. The used routing protocols are tested on a simulator as well. Currently, the software is being tested on a real large scale using the Virtual Wall [6] and Wireless lab of the i-Lab.t Technology Center [5] at IBBT. Results and details of performance measurements of the current implementation are given in section 6.

6 Performance of the proof of concept implementation

In this section most important aspects of the current proof of concept implementation will be evaluated for a number of aspects. First of all we will measure the additional delay introduced by the VPAN middleware: how long does it take to send a packet from one node to another node due to the additional packet processing of the VPAN platform? The used metric therefore is the RTT averaged over 60 ping requests during one minute with packets of 1400 bytes. This is done for intra-cluster as well as inter-cluster traffic in two scenarios. Also the maximum throughput (for intra-cluster traffic as well as inter-cluster traffic) is measured for both TCP and UDP traffic. Regarding UDP-traffic, we measured the maximum bandwidths without observed packet loss, unless mentioned otherwise. For these throughput tests, we will talk about TUN-traffic, which is the VPAN traffic sent by the application to the private VPAN IP and thus using the TUN virtual network interface. The impact of the platform on the CPU (measured relative to the maximum possible CPU load) for a set of data bandwidths will also be evaluated in these scenarios. For inter-cluster traffic, local Ethernet connections as well as commercial available Internet connections are used. Next, the overhead of additional packets for cluster formation, addressing and gateway selection will be discussed. Of course, as self-organization and mobility-management is a key feature, the ability of the platform to cope with changing topologies will be thoroughly evaluated. For this aspect, we will measure how long it takes to detect these changes, how fast packets are re-routed to their destination and how many packets are lost (in queues) due to these changes. Different kinds of topology changes will be investigated. Therefore, each used scenario will be introduced further in this chapter.

6.1 VPAN software overhead

As already stated, the networking components of the VPAN software have been developed in Click Router, running in user level. All VPAN related traffic is processed by these Click Router components thereby introducing additional overhead (e.g. beaconing, routing, encryption,...) compared to traditional processing by the operating system networking stack. To this end, we have measured the highest achievable TCP and UDP throughput and round-trip-time (RTT) for standard Linux forwarding, VPAN forwarding using encryption and VPAN without using encryption. Encryption uses 128-bit AES (Advanced Encryption Standard) [18], complemented with a 160-bit SHA1 digest and replay counter using the OpenSSL [44, 17] library.

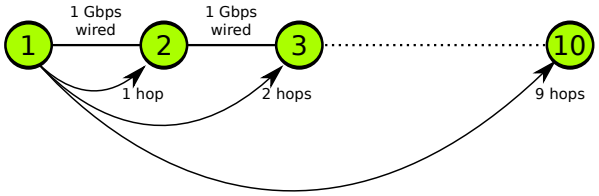

Fig. 3 Intra-cluster forwarding performance test setup.

In Fig. 4 the RTT of plain Linux forwarding for a setup consisting out of 10 interconnected Virtual Wall [6] nodes (as shown in Fig. 3, using 2 Dual-Core AMD Opteron™ 2212 processors[1] running at 2010 MHz with 1 MB cache) is compared to the RTT of VPAN-forwarding with encryption turned off and on. We observe that the additional processing of the userlevel Click and flow through the VPAN routing elements adds nearly 30 % to the normal RTT when not using encryption. When using hop-by-hop AES-encryption, another 30 % is added. This is caused by the time needed for encrypting and decrypting the traffic.

When using automatic TCP window scaling, the throughput, averaged over 10 runs, is limited to 183 Mbps (not shown) for one hop forwarding and a more or less constant 178 Mbps for 2 till 9 hops without using encryption. Encryption limits this further to 105 Mbps and 78 Mbps for 1 and 2 till 9 hops respectively. These values show the limits of the current VPAN implementation, but it may be clear that this does not hinder most applications on normal devices with normal network bandwidths of, e.g. 54 Mbps for IEEE802.11 wireless networks which will be the real bottlenecks. When emulating this behavior by applying a fixed TCP-window of

[1] Click is single threaded so only one core is used for Click/VPAN

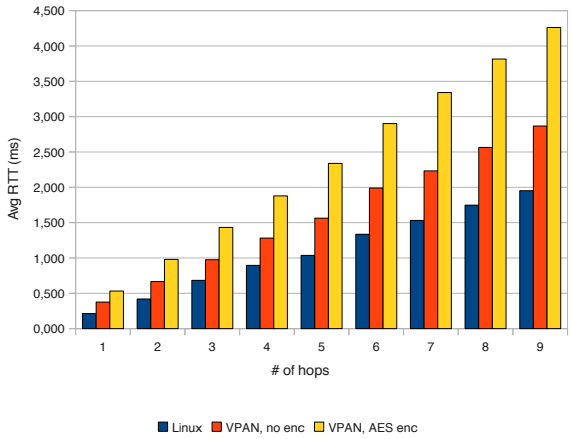

Fig. 4 Average round-trip-time versus number of hops.

16 kB we get results as shown in Fig. 5 where the achieved decrease in throughput is due to the increased RTT.

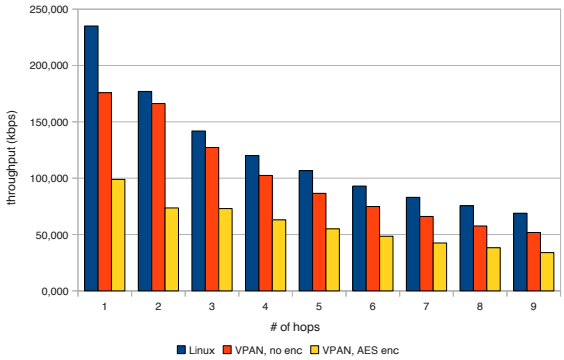

Fig. 5 Throughput versus number of hops with 16k TCP window, averaged over 10 runs.

The above discussion reveals that the use of hop-by-hop encryption is probably the most contributing factor to the increased RTT and thus decreased TCP throughput. To verify this, measurements have been done on the same test setup, but now both with and without encryption and for unidirectional UDP traffic. As the unidirectional UDP throughput is now insensitive to the increasing round-trip-time, there is no decrease in throughput (not shown) when using 100 Mbps links without encryption: plain Linux forwarding and VPAN forwarding both deliver a throughput of 97 Mbps. When using encryption however, this is limited to 91 Mbps. Again, in normal situations, this will not become a bottleneck.

It may be noted that when using Click Router as a kernel module with network cards that support polling, forwarding rates may be achieved that completely outperform moderate Cisco hardware routers on regular hardware [33] of not even a tenth of the cost of the mentioned Cisco router. So, when a higher throughput is required for e.g. certain central application servers, further research can be done on running VPAN in kernel level and using hardware encryption. Our research however focuses on supporting VPAN on a broad spectrum of devices, so no assumptions on the used kernel can be made for the end user devices.

Of course, running VPAN in user level comes with a certain cost of additional CPU usage due to context swaps between the application and the kernel and a more expensive kernel path when using the universal TUN/TAP-devices. CPU usage versus requested throughput, on the same hardware as previously in this section, is shown in Fig. 6 for a node playing the role of forwarding (F) node: accepting packets from other nodes (S) and sending them out on another interface to the destination (D) or next node. Different roles also give different CPU usage, as shown in Fig. 7 for a throughput of 10 Mbps.

In practice, the role of most nodes will be mixed as they will have incoming as well as outgoing traffic and they may have a forwarding function as in a wireless mesh on site. The results of measurements with this kind of mixed traffic are listed in table 1. Symmetric loads as well as a typical load scenario with only a small part of traffic originating from or destined for the considered node are given. The CPU load remains acceptable but some slight packet loss can be observed with increasing bandwidth. As no packet loss can be observed for traffic generated by the node, this might be an implementation issue for which future work is required. However, we can conclude that additionally placed intermediate nodes, expanding the coverage area of the wireless mesh network, will be able to support decent bandwidths. We also made certain that CPU usage remains almost constant during the whole time a constant throughput test is run.

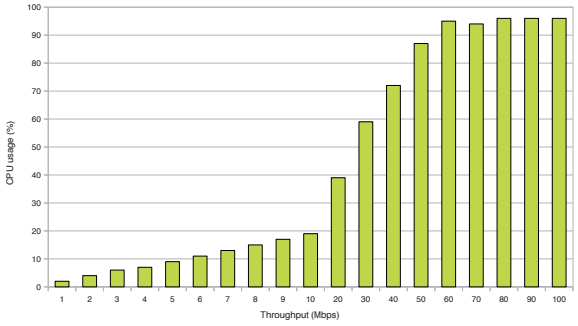

Fig. 6 CPU usage versus throughput in a forwarding node.

6.2 Cluster Formation

6.2.1 Impact of Beacons

The detection of neighboring VPAN members and the detection of link breaks is done by sending beacons on a pe-

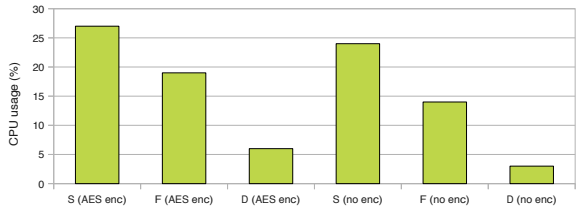

Fig. 7 CPU usage for sending, forwarding and destination nodes, all with and without encryption, for a stream of 10 Mbps.

Table 1 CPU load of the middle node and packet loss in a scenario with 3 nodes on one line and 3 UDP data streams. Flow A streams from Node 1 to Node 2, B from 2 → 3 and C from 1 → 3.

BW A (Mpbs)	BW B (Mbps)	BW C (Mpbs)	CPU load (%)	Loss A (%)	Loss B (%)	Loss C (%)
10	10	10	60	0	0	0
20	20	20	77	1.3	0	1.4
30	30	30	91	0.22	0	0.1
40	40	40	98	5.6	0	5.4
10	10	30	57	5.6	0	4.5

riodically base. These beacons are small packets being sent for every VPAN the node participates in, introducing additional overhead, especially when wireless links are used. Decreasing the beacon interval will deliver faster neighbor and link break detection, but involves more overhead. Also an increasing number of VPANs running on (partly) the same devices, or on devices using the same shared medium, will result in more beacons being sent. [26] shows that the achievable throughput stays acceptable for beacon intervals above 5 ms for 1 VPAN (where however some false link breaks will be detected) and 200 ms for up to 10 VPANs. In practice, the choice of the beacon interval depends on the requested responsiveness of the system. For mobile systems, an interval between 200 ms and 1000 ms seems reasonable as link breaks are detected at least after 3.5 times this interval. Increasing performance can be achieved when this beacon interval can be dynamically adapted to the device mobility and its moving speed.

6.2.2 Link setup time

Once a VPAN member receives a beacon of a neighboring VPAN member, a secure link can be established through a three-way handshake process. This link setup time is approximately equal to $2 * RTT + T_{proc}$ for the node sending the beacon, where RTT is the round-trip-time of the link and T_{proc} the time needed for packet processing and cryptography. The node receiving the beacon only has to send the challenge and awaiting the first response so it takes only $RTT + T_{proc}$ for this node to consider the link as established. The share of each cryptographic step in this time period is given by Fig. 8. The average time it takes for both neighbors to detect each other is thus equal to $2 * RTT + T_{proc} + \frac{BEACON\_INTERVAL}{4}$ since both nodes are sending beacons. For wireless links, the link delay is the largest contributor to the link setup time.

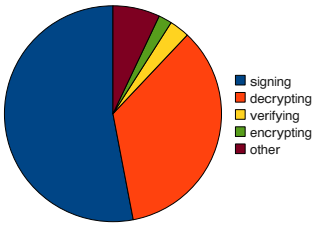

Fig. 8 Share of each cryptographic step using 1024-bit RSA.

When triggered by a NETLINK event generated by the kernel (after e.g. plugging in a cable), measurements show that the challenge-response session is started within 46 ms. On a low-powered embedded device (an ALIX system board [8] with an AMD Geode LX800 CPU running at 500 MHz), the handshake is completed and the link established after another 160 ms.

Priority scheduling with priority for control packets prevents these packets to be dropped and thus prevents false link breaks due to high background traffic. The resulting link setup time is also much better, but far not as good as the link setup times that have been achieved for lower bit rate background streams for the following three reasons.

First of all, this is due to the fact that the network cards have their own queue. Consequently, the control packets, although prioritized in VPAN software, will arrive in the queue on the network card. There, they have to wait until all other data packets already in this queue have been sent. Next, it is also possible that there is also non-VPAN traffic. This traffic, together with the traffic sent by the VPAN software, will, before it arrives at the interface, be aggregated into a queue in the operating system. Finally, when trying to send the control packet over the wireless medium, the node still has to contend with other nodes in order to access the wireless channel, which can again lead to larger link setup times when the channel is saturated. In order to solve these problems, additional priority scheduling mechanisms at these three levels are needed in order to really prioritize our VPAN control traffic. With all these mechanisms, link setup times comparable to those obtained in the absence of background traffic could become realistic in all scenarios.

6.3 Addressing Overhead

In a cluster that is not registered, some nodes can have an unverified address. Such an address is generated by the nodes

themselves and is not guaranteed to be unique in the cluster. Therefore, nodes that have an unverified address periodically announce this address to all other nodes, using blind flooding, in order to detect duplicate addresses. Upon detection of an address collision, one of the nodes generates a new address. This introduces an overhead of $\frac{N_{UNVERIFIED}*N}{T_{ADDR_INTERVAL}}$ packets per second with $N_{UNVERIFIED}$ the number of nodes with an unverified address, N the total number of nodes in the cluster (to take into account the blind flooding process) and $T_{ADDR_INTERVAL}$ the interval (in seconds) with which the addressing updates are sent. Once the cluster has registered, all addresses are verified by the VPAN Agent. From that point on, the nodes continue to use their verified address as long as they remain a member of the VPAN and consequently, $N_{UNVERIFIED}$, and thus the addressing overhead, becomes equal to 0.

6.4 Tunneling

For the tunneling of the inter-cluster traffic a number of possibilities exist. By default, IPSec tunneling is used to encrypt and encapsulate the packets. Since encryption is a time consuming process, we also offer the possibility to turn off encryption of specific inter-cluster traffic. In that case, IP in IP encapsulation is used. This can be interesting when already using a secured WAN-technology like for example an MVNO as suggested in [23]. Finally, when one or both endpoints are behind a NA(P)T box, normal IPSec or IPinIP will not work anymore and IPSec over UDP or IPinIP over UDP must be used. We evaluated the impact of these different tunneling types on the maximum achievable TCP throughput and average round-trip-time of ping packets of 1400 bytes between two VPAN gateway nodes and compared these results with the values obtained when having direct communication between the gateway Nodes. This for a lab setup with 100 Mbps links as well as using the commercial Internet service providers.

The results are shown in Fig. 9. The results show that using VPAN with its AES IPSec encryption and decryption leads to a reduction in the maximum achievable throughput (≈ -9 % for all scenarios without NAT and -10 % for the NAT-situation where additional UDP-encapsulation is necessary) and an increase of the average round-trip-time with $\approx$ 7 % when using the bandwidth-constrained commercial providers and 80 % and 70 % when using a direct connection respectively a NAT-box (regular desktop PC with iptables). Compared to IPSec, IPSec over UDP does not significantly decrease the throughput, but has mainly an impact on the round-trip-time. Using IPSec over UDP, a slightly more time is needed to encapsulate and decapsulate packets, but no impact on the rate at which packets can be encrypted and decrypted is observed. It should be noted that the achieved throughput is still sufficiently high, making the impact of encryption unnoticeable when using lower bandwidth WAN interfaces (e.g. WLAN, UMTS, broadband access...), which will often be the case. Also, with hardware encryption mechanisms better results could be achieved. Finally, it is clear that when a node is part of multiple VPANs at the same time, the maximum available throughput per VPAN becomes lower since the available bandwidth is shared by the different VPANs. QoS scheduling between VPANs could be applied by the node to offer bandwidth guarantees to specific VPANs.

In our implementation, when 2 VPAN gateway nodes are behind NA(P)T and no direct tunnel (possibly using hole punching) can be established, both nodes will establish a tunnel to the VPAN Agent. Communication between both nodes will then take place over two concatenated tunnels. In this case (not shown), throughput significantly drops with almost 40% mainly due to the time needed for the additional cryptographic steps and the increased round-trip-time.

6.5 Routing and Tunneling Strategies

The VPAN routing and tunneling framework allows different types of intra-cluster and inter-cluster routing and overlay establishment strategies. The intra-cluster routing strategies are based on WRP [35] and AODV [38] and exhibit similar performance that will not be repeated here. More details can be found in [26].

6.6 Gateway selection

In 5.8, the concept of gateway selection is explained. Of course, the announcement of all gateways in a cluster can cause overhead as this is done by sending cluster-wide broadcast packets announcing the gateway and its properties. The time between two announcements can be quite long when the gateway information stays rather static. When a new node joins the cluster, all gateways need to broadcast their information again to inform the new node as well as possibly other nodes in case of a cluster merge enclosing more nodes.

It can however be interesting to decrease the time between two packets for a certain gateway, e.g. when using a load balancing algorithm where nodes spread their traffic across different gateways and access networks. In that case, the remaining available bandwidth at the different gateways must be broadcasted more often into the cluster.

6.7 VPAN dynamics

As VPAN is designed to ease communication in dynamic environments, by creating a secure overlay where session

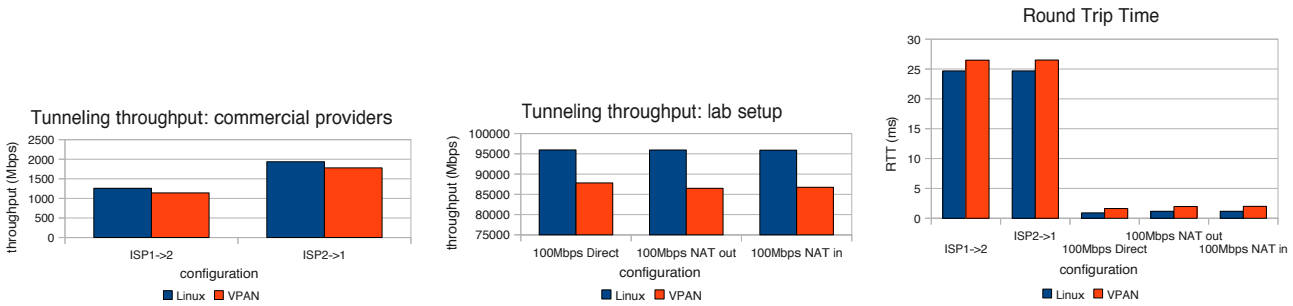

Fig. 9 Throughput and round trip time for different tunneling scenario's

continuity is guaranteed, it is interesting to estimate how the overlay reacts on certain topology changes. The basic principles are already explained in the previous sections.

6.7.1 New links and link break detection

In 6.2.1, the concept of neighbor detection using beacons is explained and an indication of the speed of establishing new links is given in 6.2.2. It is also mentioned that these beacons are also used to detect link breaks: when a node did not receive a beacon from a neighboring node during a certain time (currently: 3.5 times the beacon interval) it considers the link as broken. This implies that using the default beacon interval of 2 s, it can take up to 7 s for a link break to be detected[2]. When this link is used for forwarding packets, this means we can have 100% packet loss during 7 s before the route is deleted from the routing table and an alternative route is selected and used.

Additional triggers can speed up the process: when using the kernel's NETLINK events, it takes only 3 ms to delete a route from the routing table after a cable has been unplugged. For wireless disconnects, this can not be used. Additional research is necessary in order to decrease these timeouts. One can think for example of monitoring the wireless retransmissions counter for a sudden increase.

6.7.2 Cluster join, merge and leave

An interesting scenario illustrating the local optimalization and session continuity in dynamic environments is given in Fig. 10. Node 3 (generic laptop) has an ongoing session with Node 2 (generic laptop) while a link between Node 3 and Node 1 (ALIX-box) is established or removed. Before the link establishment, Node 3 uses its WAN-interface to reach Node 2. Upon joining the cluster and link establishment between Node 3 and Node 1, Node 3 reroutes all traffic over its PAN-interface and starts using the (probably shorter) path over Node 1 towards Node 2. Sending a 1 Mbps UDP stream from Node 3 to Node 2, plugging the cable in, out and back in, results in traffic flows measured and shown in Fig. 11 and 12.

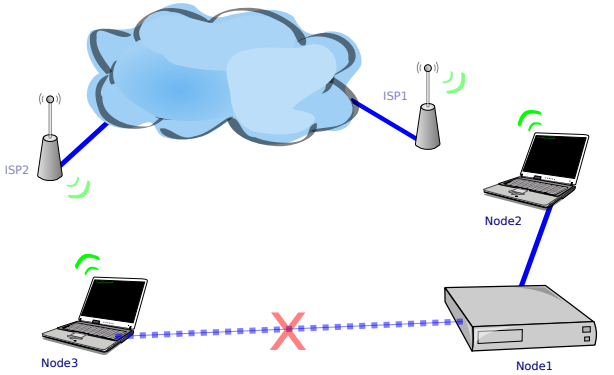

Fig. 10 Cluster join and leave scenario: Node 3 has an ongoing session with Node 2 while a link between Node 3 and Node 1 is established or removed.

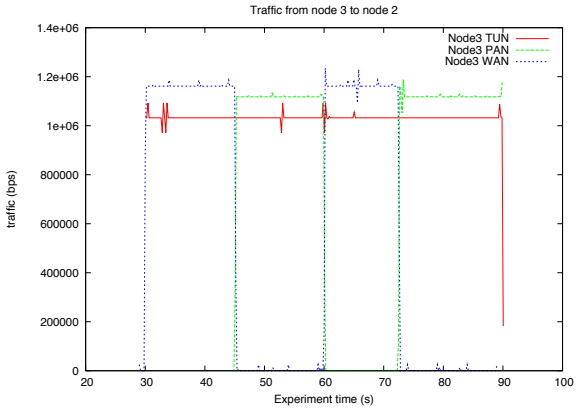

Fig. 11 Measured traffic on the interfaces of Node 3 for the scenario in Fig. 10.

Thanks to the use of NETLINK events, the plugged cable is immediately detected and a new link is established in 513 ms and all traffic is rerouted inside the cluster, as seen on Fig. 11: WAN-traffic drops to zero while at the same

[2] Keep in mind that all parameters, for example the link break timeout interval, can be adapted in advance resulting in more overhead but also in a more stable connection

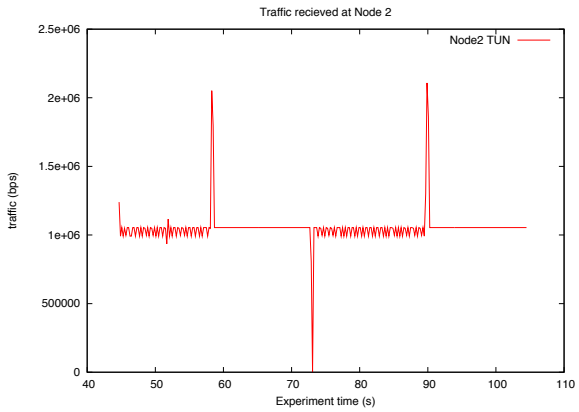

Fig. 12 Measured incoming VPAN traffic on Node 2 for the scenario in Fig. 10.

time PAN-traffic increases till a little bit more than 1 Mbps. For 1 Mbps of VPAN traffic, WAN-traffic level is higher than PAN traffic levels as WAN traffic is L3 encapsulated while PAN-traffic is L2 encapsulated. A peek can be seen on Fig. 12: there are still some packets traveling over the Internet (or just released from send buffers on interfaces) while the first intra-cluster packets are already arriving due to the lower delay.

When we pull out the cable again, this is detected by the kernel and the VPAN framework is notified through a NETLINK event resulting in deletion of the route within 13 ms. PAN and WAN lines switch again in Fig. 11 as the other interface is used now, while traffic drops to zero for a little while in Fig. 11 as the new route causes extra delay and some packets may get lost while pulling the cable. After putting the cable back again and streaming for a few seconds, the experiment is finished resulting in a packet loss of 0.34 % for the whole time. 61 packets out of 5104 arrived out of order. All these values are averaged over five runs.

Of course, such fast handovers are only possible thanks to the kernel detection of the cable status and its announcement using NETLINK events. When we connect Node 1 and Node 3 using a switch, detection only happens at one side of the switch and thus only at one of both nodes. When putting in the cable, this is not a problem as at least one node detects this and sends a beacon immediately. While pulling the cable out however, the other node does not notice the link break until a timeout occurs while not receiving any beacons from the other node anymore. This results in a time frame of ≈ 5 *s* where traffic is being sent but not received at the remote side: 10 s of streaming results in ≈ 60 % packet loss. Decreasing the beacon interval from 2 s to 200 ms results in a gap of ≈ 1 *s* and 5.94 % packet loss over a total time of 10 s at 1 Mbps.

6.7.3 Gateway dynamics

In section 5.8, the concept and features of gateway selection is explained. In this constellation, three main types of events can occur that have impact on routing choices: interface changes at the currently used gateway (a better interface comes available or the currently used interfaces becomes unavailable), a better gateway becomes available or joins the cluster, and finally the currently used gateway can become unavailable (e.g. loosing its registration) of leaves the cluster. We've put all these events together in one big scenario as shown in Fig. 13. ISP 1 is the Flemish cable operator Telenet while ISP 2 is the Belgian ADSL provider Belgacom and ISP 3 is the Belgian research network provider BELNET, where also the VPAN Agent is hosted.

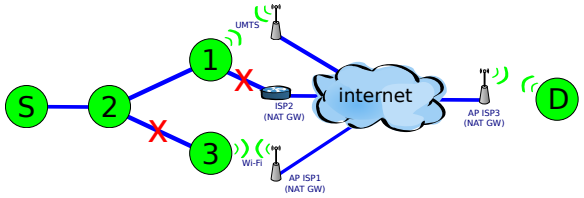

Fig. 13 One scenario to illustrate all kinds of gateway changes.

Interface changes In this case, we neglect Node 3 and focus on Node 1. Currently, the UMTS-interface is used, as the cable to the Internet is unplugged. When plugged in, an ADSL connection becomes available and is favourized against the UMTS connection. All traffic gets rerouted over this interface. Note that this only affects Node 1: all other nodes might receive a gateway discovery update packet, but they keep sending to the same node and do not change anything in their internal routing.

In our scenario, the plugged in cable is detected immediately and the old tunnel is deleted after 6 ms. However, it takes 2734 ms to establish a new tunnel over the newly installed interface. This is due to the NA(P)T detection process. Depending on the used type of NAT, some packets get not through the NA(P)T box and do not arrive at the gateway node so we have to wait for a timeout. This timeout is configurable in the gateway and has been set to 2500 ms for this experiment. When streaming at 100 Kbps (because of the UMTS interface) during 10 s, we loose 24/87 packets or 28 %, averaged over 5 runs. Future work is necessary in order to immediately start using the available tunnels (the VPAN Agent tunnel in most cases) and only switch to another tunnel when this tunnel is established. This should result in 0 lost packets.

The other way around (unplugging the cable) results in the same values for this situation, with a faster tunnel setup (2134 ms) but slower route propagation due to the long delay over UMTS.

Gateway joins cluster The impact on an active session, when a gateway, that is an endpoint of that session, joins the cluster, is demonstrated in section 6.7.2. When the new gateway however is not an endpoint, it can have an impact on other sessions as it can be interesting to reroute traffic over this new gateway if its properties (e.g. monetary cost) are better. This is demonstrated in Fig. 13 where Node 2 sends traffic to Node D using the UMTS connection of Node 1 as gateway. Both cables (Node 1 - ISP 2 and Node 2 - Node 3) are unplugged. When we make Node 3 joining Node 2's cluster by plugging in the cable, Node 3 is selected as new best gateway because of the low cost of its WiFi connection to ISP 1. For a UDP stream of 200 kbps from Node 2 to Node S, plugging in the cable results in loosing 1 of 172 packets and 1 packet received out of order. Rerouting the stream takes as long as the link establishment process and receiving the (immediately triggered) gateway announcement packet of Node 3 by Node 2.

We repeat this process also with a stream of 1 Mbps during 20 s marking Node 3's WiFi-connection as expensive and Node 1's fixed connection as cheap. UMTS is not used in this experiment. Results show a loss of 4 packets of the 1702 sent out ones and 8 packets received out of order. Detailed throughput measurements can be seen in Fig. 14. There is no notable throughput drop, but even a throughput peek right after rerouting traffic towards Node 1. This is due to the fact that packets are still on there way through the network of ISP 1 and some may still be buffered in the network interface(s) while the first packets through the new route already arrive.

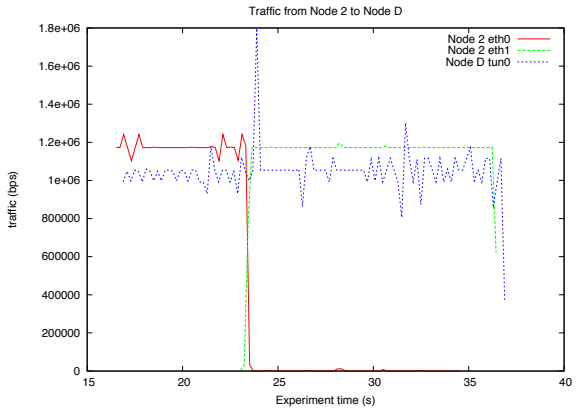

Fig. 14 Measured traffic from Node 2 to Node D for the scenario in Fig. 10.

Gateway leaves cluster The inverse situation is also investigated: when Node 3 (which is used as best or cheapest gateway) leaves the cluster, all traffic from Node S towards Node D will be rerouted over Node 1 again, using the UMTS connection. For a 200 kbps stream of 10 s, this results in 9 lost packets or a loss rate of 5 %.

When we repeat the same experiment with Node 3 as expensive gateway and the wired interface of Node 1 as best interface, we get a loss rate of 1 % or 17 packets out of 1702 from a 1 Mbps UDP stream of 20 s when we disconnect the cable between Node 1 and Node 2, which is the sender now. The used ISPs are the same as in the previous paragraph. However, when we use TCP with the maximum available bandwidth, we get a throughput evolution as shown in Fig. 15. There we disconnected and connected the cable between nodes 2 and 1 a few times. As ISP 2 has a higher up-link bandwidth than ISP 1, we get a better throughput when the cable is connected than when it is disconnected. Notice the fact that no packets get lost and the TCP stream continues. The peaks are again due to packets that are sent by the old gateway and were still traveling over the Internet when the stream was rerouted.

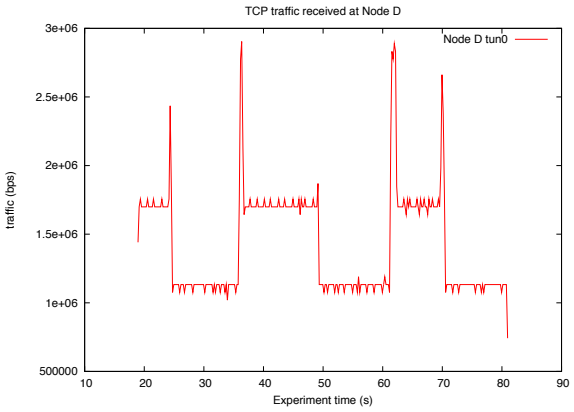

Fig. 15 Measured traffic from Node 2 to Node D for TCP throughput maximization of the scenario in Fig. 10.

7 Evaluation and future work

7.1 Mapping the architecture on emergency scenarios

As also proposed by different authors [40,29,30,15,24,37] and projects [3,2], a wireless mesh network deployed at crisis sites can be a great tool to allow a fast setup of basic communication infrastructure providing higher bandwidths, interconnected by other publicly available or dedicated networks like TETRA or satellite or 3G. As described in [21], VPAN allows usage and fast deployment of this mesh network as it takes care of security, mobility and interconnecting remote clusters seamlessly. VPAN can even create this mesh network itself.

All emergency workers, organized in teams, use their own VPAN and share the ad hoc deployed and public infras-

tructure in a secure and shielded way. Certain services (eg. location-service providing the current GPS-position) are shared in the VPAN, next to Voice-over-IP (VoIP) services as well as an emergency management service and GIS-services installed in the back-end. Some nodes participate in multiple VPANs. For example team leaders participate in an emergency coordination VPAN as well as their own team VPAN. Even integration between a VoIP application in the VPAN and the TETRA voice services is possible as demonstrated by [2].

Thanks to its session mobility and hierarchical approach, VPAN allows emergency workers seamless data consultation while on the way, upon arrival on site, and when switching underlying communication technologies or choosing other gateways [20]. Cluster-wide and VPAN-wide broadcast allows bandwidth optimalization when all cluster members (eg people inside a truck) need to receive the same information using a shared or bandwidth-limited up-link like TETRA or 3G: information is only sent once through the tunnel, while distributed locally in the cluster.

7.2 Evaluation of the requirements

In this paragraph, our architecture is matched against the requirement summary of section 2:

- Membership configuration and management: this is done through use of VPAN profiles and certificates that link members to VPANs. Only member devices with the right certificate installed, can connect to other members and the VPAN Agent.
- Ease of use and fast ad hoc deployment: once installed (which is also easy), the platform acts as middleware and can operate autonomously without requiring user interaction. Once turned on, the software immediately starts the neighbor detection and registration process allowing fast deployment.
- Local and distributed operation: thanks to the neighbor discovery and cluster formation, the VPAN platform can be used locally, without Internet connection. When an interconnecting infrastructure is (or comes) available, distributed operation is possible as registration and tunnel setup is done automatically.
- Security: access to the network is based on membership, controlled by the installed certificate. As such, these applications can only be accessed through the VPAN and so only by members of that VPAN. Next, all traffic, intra-cluster as well as inter-cluster, is encrypted.
- Self-organization and mobility management: session continuity is guaranteed thanks to the use of a private addressing space with fixed IP-addresses. Neighbor discovery, ad hoc routing and dynamic tunneling allow dealing with topology and access network changes.
- Application support: access to services and applications can be controlled by configuring them to bind to the correct IP-addresses in the right VPANs or limiting access by a firewall. Therefore a service sharing and discovery mechanism is developed allowing to easily give applications access rights to the selected VPANs in a user friendly manner. This also works the other way around: by specifying to which VPAN some applications have access to, we also control which users have access to which applications and services running on the remote devices, based on the VPAN they participate in. Support for all kinds of applications is guaranteed as the only requirement at the application level is plain IPv4 compatibility. Transition to IPv6 is possible.
- Scalability: cluster scalability is impacted by the number of interfering beacons (discussed in 6.2.1 and [26]) and the scalability of the intra-cluster ad hoc routing protocol (extensively evaluated in literature). For the proactive routing solution, the hierarchical routing approach ensures that nodes in a cluster only need a route to all other nodes inside the cluster and that gateway nodes only need to distribute changes in the composition of the cluster between each other, reducing the routing overhead and improving scalability (see [26]). However, when the number of clusters or gateways increases, the number of tunnels will also increase, since a full mesh is being established in the proactive approach. An approach introducing super-members in order to avoid a full mesh could then improve this scalability. Alternatively, the reactive solution could be used to only establish tunnels and routes when needed (see [26]). The VPAN Agent can be a bottleneck when all member nodes are gateways behind NAT, so further research is necessary to provide a distributed Agent functionality. Also the number of nodes in a VPAN is limited to the number of IP-addresses in the private addressing subnet used by that particular VPAN. The number of VPANs that can exist next to each other is also limited by the number of IP subnets.
- Heterogeneity:currently, a proof-of-concept implementation has been built on various operating systems and hardware platforms. In fact, support for every device or communication technology with standard L2 and L3 networking capabilities should be possible.

7.3 Future work

It is clear that our platform provides an integrated approach capable of dealing already very well with the requirements imposed by emergency communications. Of course, the performance results also revealed some possibilities for enhancements and optimizations.

As already mentioned in section 7.2, a distributed VPAN Agent must be developed in order to improve scalability. Also other scalability issues like the limitations of the used IPv4 subnets must be tackled. Next to this, we already pointed to a dynamic beacon interval in 6.2.1, which may be implemented using GPS integration on some devices or other methods.

Next to cluster-wide and VPAN-wide broadcasting, multicast functionalities in a VPAN are a point of ongoing research. Also better intra-cluster ad hoc routing protocols must be investigated to take eg. link quality into account and focus on (TCP) throughput, reliability, load balancing and quality of service. A lot of literature in this field is already available [12, 16, 43] and new protocols are being developed by lots of research groups around the world.

Intelligent (type of service based) gateway selection is now only available for outgoing traffic. An extension should be developed in order to make a better choice as remote gateway when there are more gateways in the remote cluster. Quality of Service in general is a topic where lots of opportunities for further improvements reside.

Of course, VPAN development should keep phase with new technologies and platforms. For example a port to the iPhone and Android platform can bring VPAN to even more mobile devices as these platforms are growing in popularity.

And last but not least: sensor networking is a hot issue. Integration of sensor networking, by using a Zigbee coordinator as VPAN-gateway to the sensor network like in the ITEA2 USENET project [9], or by porting the minimal VPAN functionalities (like neighbor detection and intra cluster routing) to sensor platforms, are investigated.

8 Conclusion

In this paper we have presented the concept of VPANs, a next-generation communication paradigm that tackles an important communication need absent in current or future networks, especially in emergency and NSPS networks: i.e. the need for secure communication between a limited number of local and distributed devices and for a variety of services and applications. Through its generality, the concept is capable to support a wide range of applications. We have shown that existing technologies are not capable to meet all challenges involved with this concept and all emergency and NSPS communication requirements, while the VPAN platform tackles them all, or at least partly regarding scalability.

The VPAN concept has been translated into an architecture and a fully working prototype has been implemented and extensively evaluated. The functionalities of this prototype are described and performance results are presented. Results are analyzed offering valuable insight into the performance of the implemented system and providing guidelines for future extensions, of which a few are listed above.

We hope that more research can make this powerful concept succeed in the real world, eases communication and add real value to save lives.

Acknowledgements This research is partly funded through the ITEA2 UseNet (Ubiquitous M2M Service Networks) project and the Interdisciplinary Institute for Broadband Technology (IBBT) projects GeoBIPS and ADAMO. Peter Dedecker is research assistant at University College Ghent and affiliated researcher at Ghent University.

References

1. Hamachi: Instant, zero-configuration vpn. URL https://secure.logmein.com/products/hamachi/vpn.asp
2. IBBT ADAMO Advanced Disaster Architecture with Mobility Optimizations. URL http://www.ibbt.be/en/project/adamo
3. IBBT GeoBIPS Geographical Broadband Integration for Public Services. URL http://www.ibbt.be/en/project/geobips
4. Ietf network mobility working group. URL http://www.ietf.org/html.charters/nemo-charter.html
5. iLab.t Technology Center @ IBBT. URL http://ilabt.ibbt.be
6. The iLab.t virtual wall. URL http://ilabt.ibbt.be/?p=page5
7. Leetnet: The true dynamic vpn. URL http://www.leetnet.org
8. PC Engines^{TM}ALIX system boards. URL http://www.pcengines.ch/alix.htm
9. Ubiquitous M2M Service Networks (UseNet). URL http://usenet.erve.vtt.fi/
10. Vpn technologies: Definitions and requirements, white paper (2004). URL http://www.vpnc.org/vpn-technologies.html
11. Statement of Requirements for Public Safety Wireless Communication and Interoperability (2006). URL http://www.safecomprogram.gov/SAFECOM/library/technology/1258_statementof.htm
12. Abdulrahman, Altalhi H., u., Golden, Richard G., u.: Load-balanced routing through virtual paths: Highly adaptive and efficient routing scheme for ad hoc wireless networks. In: H. Hassanein, R. Oliver, G. Richard (eds.) The 2004 IEEE International Performance, Computing, And Communications Conference (IPCCC), pp. 407–413. IEEE (2004)
13. Anderson, T., Peterson, L., Shenker, S., Turner, J.: Overcoming the internet impasse through virtualization. Computer **38**(4), 34–41 (2005)
14. Birman, K.P.: The next-generation internet: Unsafe at any speed? Computer **33**(8), 54–60 (2000)
15. Chiti, F., Fantacci, R., Maccari, L., Marabissi, D., Tarchi, D.: A broadband wireless communications system for emergency management. IEEE Wireless Communications **15**(3), 8–14 (2008)
16. Couto, D.S.J.D., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. Wireless Networks **11**(4) (2005)
17. Cox, M., Engelschall, R., Henson, S., Laurie, B., et al.: The OpenSSL project (2002)
18. Daemen, J., Rijmen, V.: The design of Rijndael: AES–the Advanced Encryption Standard. Springer Verlag (2002)
19. Dahlman, E., Parkvall, S., Skold, J., Beming, P.: 3G Evolution, Second Edition: HSPA and LTE for Mobile Broadband. Academic Press (2008)
20. Dedecker, P., Hoebeke, J., Moerman, I., Moreau, J., Demeester, P.: Multipath routing issues in virtual private ad hoc networks. Personalized Networks, 2009. IEEE CCNC '09. Workshop on (2009)
21. Dedecker, P., Hoebeke, J., Naudts, D., Moerman, I., Moreau, J., Demeester, P.: Fast and safe emergency communication through network virtualization. In: International Conference On Communications And Mobile Computing, p. 4 (2009)

22. Doval, D., O'Mahony, D.: Overlay networks: A scalable alternative for p2p. Internet Computing, IEEE **7**(4), 79 – 82 (2003)
23. Ericsson: Keeping lifelines open: national security & public safety communication over mobile broadband (white paper) (2010). URL http://www.ericsson.com/article/nsps_20100216130134
24. Fantacci, R., Vanneschi, M., Bertolli, C., Mencagli, G., Tarchi, D.: Next generation grids and wireless communication networks: towards a novel integrated approach. Wireless Communications and Mobile Computing **9**(4), 445–467 (2009)
25. Freeman, J., Passmore, D.: The virtual lan technology report (1996)
26. Hoebeke, J.: Adaptive ad hoc routing and its application to virtual private ad hoc networks. Ph.D. thesis, Ghent University (2007)
27. Hoebeke, J., Holderbeke, G., Moerman, I., Dhoedt, B., Demeester, P.: Virtual private ad hoc networking. Wireless Personal Communications **38**, 125–141 (2006)
28. Hoebeke, J., Steenhuyse, M., Ackaert, A., Moerman, I., Demeester, P., Jacobs, A., Veys, A., Verhoeve, P., Piepers, B.: A new concept towards secure personal healthcare platforms. In: Proceedings of TTEC2008, the Tromso Telemedicine and eHealth Conference, pp. 25–26 (2008)
29. Iapichino, G., Bonnet, C., del Rio Herrero, O., Baudoin, C., Buret, I.: Combining mobility and heterogeneous networking for emergency management. ACM Press, New York, New York, USA (2009)
30. Iapichino, G., Bonnet, C., Del Rio Herrero, O., Baudoin, C., Buret, I.: A mobile ad-hoc satellite and wireless mesh networking approach for public safety communications. In: SPSC 2008, 10th IEEE International Workshop on Signal Processing for Space Communications, October, 6-8, 2008, Rhodes Island, Greece (2008)
31. Johnson, D., Perkins, C.: J. Arkko," Mobility Support in IPv6. Tech. rep., RFC 3775, June 2004
32. Khanvilkar, S., Khokhar, A.: Virtual private networks: an overview with performance evaluation. Communications Magazine, IEEE **42**(10), 146 – 154 (2004)
33. Kohler, E.: The Click Modular Router Project. http://www.read.cs.ucla.edu/click/
34. Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. Communications Surveys Tutorials, IEEE **7**(2), 72 – 93 (2005)
35. Murthy, S., Garcia-Luna-Aceves, J.J.: An efficient routing protocol for wireless networks. Mob. Netw. Appl. **1**(2), 183–197 (1996)
36. National Task Force On Interoperability (NTFI) And United States: Why Can't We Talk? Working Together to Bridge the Communications Gap to Save Lives: A Guide for Public Officials-Supplemental Resources (2005)
37. Naudts, D., Bouckaert, S., Bergs, J., Schoutteet, A., Blondia, C., Moerman, I., Demeester, P.: A wireless mesh monitoring and planning tool for emergency services. End-to-End Monitoring Techniques and Services, 2007. E2EMON '07. Workshop on pp. 1–6 (2007)
38. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental) (2003)
39. Perkins, C., et al.: IP mobility support for IPv4 (2002)
40. Portmann, M., Pirzada, A.: Wireless mesh networks for public safety and crisis management applications. IEEE Internet Computing **12**(1), 1825 (2008)
41. Sánchez, L., Lanza, J., Munoz, L.: Experimental Assessment of a Cross-Layer Solution for TCP/IP Traffic Optimization on Heterogeneous Personal Networking Environments. Lecture Notes In Computer Science **4217**, 284 (2006)
42. Stavroulakis, P.: TErrestrial Trunked RAdio - TETRA: A Global Security Tool. Springer Publishing Company, Incorporated (2007)
43. Triantafyllidou, D., Agha, K.A.: The impact of path-delay routing on TCP in ad hoc networks. International Conference On Communications And Mobile Computing (2009)
44. Viega, J., Messier, M., Chandra, P.: Network security with OpenSSL. O'Reilly Media, Inc. (2002)

Peter Dedecker received the Masters degree in engineering (Computer Science) from Ghent University in 2006 and joined the Intec Broadband Communications Networks (IBCN) group (Ghent University) as research engineer. In October 2007, he moved to the department of Computer Science of Ghent University College where he is currently working as a research assistant in cooperation with IBCN. His PhD research includes optimizations of the virtual private ad hoc networking technology developed at Ghent University while his main research interests are mobile broadband communication networks, and more specially ad hoc wireless communications and overlay networks.

Jeroen Hoebeke received the Masters degree in Computer Science engineering and the Ph.D degree from the Ghent University, Gent, Belgium in 2002 and 2007, respectively. He is currently a postdoctoral researcher at the Broadband Communication Networks group of Ghent University where he is conducting research on mobile and wireless networks, personal networks and network overlays.

Ingrid Moerman received her degree in Electrical Engineering (1987) and the Ph.D degree (1992) from the Ghent University, where she became a part-time professor in 2000. She is a staff member of the research group on broadband communication networks and distributed software, IBCN (www.ibcn.intec.ugent.be), where she is leading the research on mobile and wireless communication networks. Since 2006 she joined the Interdisciplinary institute for BroadBand Technology coordinating several interdisciplinary research projects. Her main research interests include: wireless broadband networks for fast moving users, mobile ad hoc networks, personal networks, virtual private ad hoc networks, wireless body area networks, wireless sensor and actuator networks, wireless mesh networks, fixed mobile convergence, protocol boosting on wireless links, QoS support in mobile & wireless networks, intelligent transport systems, self-optimization in next-generation mobile networks, network architectures and protocols for heterogeneous mobile and wireless networks. She is author or co-author of more than 400 publications in international journals or conference proceedings.

Joris Moreau graduated in Theoretical Nuclear Physics from Ghent University and has worked as a systems engineer in the banking industry for about 15 years. He currently teaches in computer networking, operating systems, relational database systems and computer graphics at University College Ghent.

Piet Demeester received the Masters degree in Electro-technical engineering and the Ph.D degree from the Ghent University, Gent, Belgium in 1984 and 1988, respectively. He is a full-time professor at Ghent University where he teaches courses in communication networks. He is the head of the Broadband Communication Networks group (www.ibcn.intec.ugent.be). His research interests include: multi-layer IP-optical networks, mobile networks, end-to-end quality of service, grid computing, network and service management, distributed software and multimedia applications. He has published over 500 papers in these areas in international journals and conference proceedings. In this research domain he was and is a member of several program committees of international conferences, such as: OFC, ECOC, ICC, Globecom, Infocom and DRCN. He is a fellow of the IEEE.