Department of Applied Mathematics
Faculty of EEMCS

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893400
Fax: +31-53-4893114
Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1802

Architectural aspects of QoS-aware personal networks

T.J.M. Coenen, P.T.H. Goering, A. Jehangir,
J.L. van den Berg, R.J. Boucherie,
S.M. Heemstra de Groot, G.J. Heijenk,
S.S. Dhillon,[1] W. Lu[1], A. Lo,[1]
P.F.A. van Mieghem[1] and I.G.M.M. Niemegeers[1]

June, 2006

ISSN 0169-2690

[1]Delft University of Technology, Delft, The Netherlands

Architectural Aspects of QoS-aware Personal Networks

T.J.M. Coenen, P.T.H. Goering, A. Jehangir,
J.L. van den Berg, R.J. Boucherie,
S.M. Heemstra de Groot, G.J. Heijenk
University of Twente
Enschede, The Netherlands
{t.j.m.coenen, a.jehangir, p.t.h.goering}@utwente.nl

S.S. Dhillon, W. Lu,
A. Lo, P.F.A. van Mieghem, I.G.M.M. Niemegeers
Delft University of Technology
Delft, The Netherlands
{w.lu, s.dhillon}@ewi.tudelft.nl

Abstract

Personal Networks (PN) are future communication systems that combine wireless and infrastructure based networks to provide users a variety of services anywhere and anytime. PNs introduce new design challenges due to the heterogeneity of the involved technologies, the need for self-organization, the dynamics of the system composition, the application-driven nature, the co-operation with infrastructure-based networks, and the security hazards. This paper discusses the challenges of security and QoS provisioning in designing self-organized personal networks and combines them all into an integrated architectural framework.

Keywords-Personal Network Architecture, Security, Quality of Service, Service Discovery
AMS Subject Classification: 68M10, 90B18

I. INTRODUCTION

The future mobile communication system is envisaged to be the convergence of wireless ad hoc networks and infrastructure based networks to provide the user a variety of services anywhere and anytime. Personal networks (PN) [24] as user-centric enablers for future wireless communications, start from the user and extend the user's personal area network (PAN) to a global coverage of his personal devices and services in his home, car, office etc. as well as other foreign networks and services regardless of their geographical locations. This extension will physically be made available via infrastructure-based networks e.g., the Internet, UMTS networks etc., together with mobile ad hoc networks. Some examples of personal devices that may be involved in a personal network are mobile phones, PDAs, laptops and digital cameras. Each device may have its dedicated functionalities and may be equipped with one or more wireless access technologies such as UMTS/GPRS networks, IEEE 802.11 WLAN technology, and IEEE 802.15 short range wireless technologies. The Internet Protocol (IP) is used as a common network protocol for all these heterogeneous underlying technologies. The use of IP as a unified network layer provides a generic solution to organize all the devices of a person into a self-organized network on top of existing and emerging networking technologies.

Furthermore, personal networks are dedicated for personalized usage, which poses additional emphasis on security and privacy issues. However, most of the networking technologies utilized in personal networks are vulnerable to security attacks such as eavesdropping and spoofing. Security mechanisms need to be taken into serious consideration throughout the design. Taking into account that a large amount of personal devices are power constraint mobile devices, any proposed security solution must be simple and lightweight such that it does not create a performance bottleneck.

In order to realize a self-organized personal network as characterized above, the following topics require specific attention. First of all, a secure PN architecture at the network layer that is independent of underlying network technologies needs to be defined. On top of that, PN communication, service discovery and provisioning mechanisms could be implemented. Finally, QoS needs to be provided to live up to customer expectations and to support current and future multimedia applications. This paper discusses these challenges in designing self-organized personal networks.

The organization of this paper is as follows. Section II presents a detailed description of a personal network and its components. A secure PN architecture is proposed, based on these descriptions. Sections III and IV discuss the self organization and security issues of the PN. Section V deals with the QoS aspects of PNs at the MAC layer and routing protocols and algorithms. Section VI concludes the paper and suggests topics for further research.

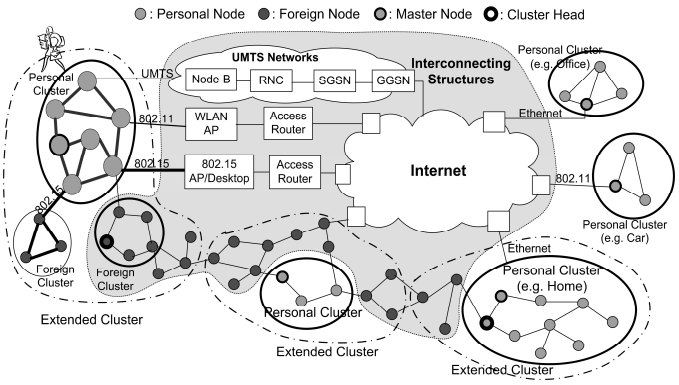

Fig. 1. A Personal Network.

II. SECURE PERSONAL NETWORK ARCHITECTURE

An instance of a personal network is illustrated in Figure 1. We start to introduce the personal network from its basic element, a personal node. Notice that, in the figure, the nodes in blue (or gray in grayscale print) are personal nodes. They are distributed over different locations, for example, staying with the person, at home, in a car or at the office. These personal nodes are equipped with at least one networking technology, which makes it possible for them to interconnect with each other or with the Internet. You might notice that there are not only personal nodes, but also foreign nodes denoted in black in the figure. Foreign nodes, as the name says, are nodes not belonging to the owner of the personal network and appear to the personal network as foreigners. Foreign nodes may belong to other personal networks, owned by other persons. Being in such an environment full of nodes belonging to a multitude of PNs, there has to be some mechanism for nodes to distinguish others belonging to their own PN from amongst all the rest. This is necessary because we do not want personal nodes from one user to be continuously trying to connect to others of a different PN and failing, yet wasting precious energy in the process. To solve this problem, we assume the existence of a PN identifier, calculated randomly over a sufficiently large space as to reduce the possibility of collisions. Therefore, wireless nodes sharing a common PN identifier will recognize each other as personal nodes and know they belong to the same PN.

Furthermore, personal nodes of the same owner that are in close vicinity may form *Personal Clusters* by interconnecting with each other in an ad hoc fashion without intervention of any foreign nodes. Personal clusters are denoted with dark circles in Figure 1. Similarly, in contrast to personal clusters, there are foreign clusters. Foreign clusters are a set of foreign nodes that may belong to another person sharing the same PN identifier. Personal clusters are basically a highly cooperative and self-organized multi-hop ad hoc network of personal nodes. Personal nodes within a personal cluster may have multiple interfaces such as UMTS/GPRS, IEEE 802.11 and IEEE 802.15 technologies that can be utilized simultaneously cooperating with each other to achieve bandwidth aggregation and load balancing and to minimize the handoff latency.

In addition, each personal cluster will elect a *Master Node*, which is responsible for the management of that cluster. The roles of the master node are multifold:

- First of all, it acts as a security agent to authenticate new nodes that join the cluster, initiate periodic cluster key updates and generate cluster advertisements. The Master node is also responsible for trust relationship establishment between different personal clusters and between personal clusters and foreign clusters if they need to communicate with each other. Additionally, the master node is able to evict members on demand and is also responsible for setting the cluster policy which lets devices joining the cluster know about various cluster parameters like the frequency of cluster advertisements and key updates, as well as the duration of timers, etc.
- Secondly, the master node is responsible for route management within the personal cluster and exchanging route information with master nodes of other clusters.
- Thirdly, the master node is responsible to collect the services provided within the personal cluster and present them to the outside world such as other personal clusters or even foreign clusters.

Furthermore, personal clusters are not in isolation, they need to extend their communication to the outside world via gateways. There could be multiple gateways that connect the personal cluster with its outside world using various technologies as depicted in Figure 1. Personal clusters can be connected to the outside world in two ways. One is via infrastructure based networks such as the Internet and the other is via ad hoc networks involving foreign nodes. Normally, infrastructure access is preferable if it is available. However, there may be some situations where infrastructure access is not available or not convenient. In this case, personal clusters can also be extended in an ad hoc fashion. The Internet and ad hoc networks of foreign nodes are therefore called *Interconnecting structures.* And in this way, personal clusters dispersed at different locations are connected and a PN is established.

As illustrated in Figure 1, connections of the personal cluster to the infrastructure based networks are enabled by connecting to an access point or a base station connected to the Internet. Due to the dynamics of personal networks, personal clusters may move from place to place. This requires nodes in a personal network to employ mobility management mechanisms so that they are reachable while moving around. A number of mobility types are identified for personal networks including personal mobility, host mobility, network mobility and session mobility. As mobile nodes are becoming more heterogeneous for specific functionalities and are equipped with multiple interfaces, mobility management at a granularity of per-session, namely session mobility, is highly desirable [21]. According to this, Mobile IP is selected as the fundamental mobility management solution for personal networks due to its global reachability and relatively low handover latency [33]. An additional functionality is added to Mobile IP to enhance the session mobility support for personal networks [21].

The other extension of personal clusters to the outside world is made available via ad hoc networking. Consider a situation when several people, with their personal clusters at hand, come together and want to exchange information with each other. Their personal clusters that are in close vicinity could form an extended cluster in an ad hoc fashion. Figure 2 illustrates an example of several personal clusters forming extended clusters based on their geographical locations.

A *hierarchical structure* is adopted for PN ad hoc communication in order to improve the scalability and reduce the control packet overhead in ad hoc routing. The personal clusters belonging to different personal networks automatically form the first level clusters and they are managed by their own master nodes. A reactive (e.g. AODV [27]), proactive (e.g. OLSR [9,12,]) or hybrid routing protocol (e.g. ZRP [14]) may

be employed for routing within a cluster and communicating with the master node. On the first level, a master node is elected to keep track of the routing information within its personal cluster and is responsible for exchanging route information with master nodes of other clusters. In this manner, routing update is only restricted within a local range; and a proactive routing protocol, such as OLSR [12], can be adopted for routing inside the personal cluster. At the second level, personal clusters belonging to different personal networks, which are in close vicinity of each other, form an extended cluster. A cluster head can further be elected to perform route maintenance within the extended cluster. At the third level, extended cluster heads could further exchange their routing information with each other. A distance vector routing scheme can be adopted to perform routing updates between master nodes and between cluster heads such as specified in [16]. Or, a hierarchical OLSR [12] scheme could be applied similarly. In this way, ad hoc routing for personal network communications can be well established even though the ad hoc network might be large in size.

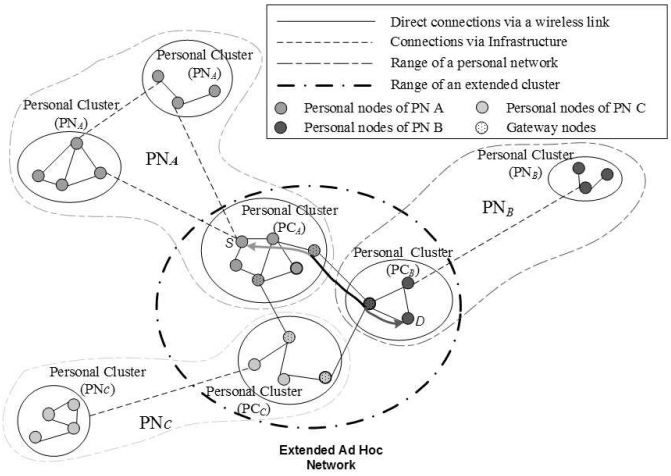

Figure 2. Extended Ad Hoc Networks

III. SERVICE DISCOVERY FRAMEWORK

A personal network consists of several nodes that self organize into clusters. For this self organization we need a service discovery mechanism to find what is available in the neighborhood, so we can form clusters as needed. These clusters can be seen as services to be discovered. Here, we define a client as the node that wants to make use of a specific service and the server as the node that offers this service. We consider the case where one or both of them are mobile wireless devices, connected in an ad-hoc fashion to other nodes. Note that they are not necessarily part of the same PN. The service discovery architecture for PNs consists of multiple distinct parts.

First, we distinguish the case where the client and server are located in each others vicinity, either in the same PN, in separate PNs or not in any PN at all. Here, the client wants to locate the nearest server with the best matching service. We can think of services like printers, displays, speakers, email or file servers, but also nodes belonging to a specific PN. We can use this information to form a PN cluster.

Second, we consider the situation where the client and the server are part of the same PN. A user of the PN wants to find one of his/her own services that could be located nearby but also in some other place anywhere in the world.

Third, we need a global service discovery, where any client in anybody's PN can find any service. This service could be located in somebody else's PN, located anywhere in the world. The client and server can be connected through some infrastructure, like the Internet. This can be seen as an extension of the second case.

We will briefly discuss local service discovery approaches in Section III.A. Section III.B will explain how we apply this local service discovery in a PN context and how this can be used for global service discovery.

A. *Local service discovery*

For service discovery in computer networks several protocols have been developed, each with their own strengths and weaknesses in different areas. We can distinguish between *centralized* and *distributed* solutions. Centralized protocols use a central node, a directory that stores all services available, while in a distributed system all nodes keep information about a part of the available services.

An example of a centralized protocol is SLP [32], where a connection to some infrastructure is assumed to be present most of the time. All devices communicate with the central directory server when they need or announce a service. The communication path and thus the nodes in the communication path towards the directory server will likely be loaded more than the rest of the network. This is undesirable for mobile devices in ad-hoc networks that have limited network capacity and power. Moving ad hoc network nodes make it more difficult to keep a stable communication path, as these protocols were not designed for these kinds of networks. Furthermore the directory server itself might disappear or get out of range for some nodes.

A distributed system has some advantages in a mobile ad-hoc network [15] and can be proactive or reactive. In a proactive system services are announced through broadcasts, while in a reactive system queries for services are propagated through the network.

Zeroconf [8], e.g. implemented as Apple Bonjour, is an IETF protocol that enables the discovery of services on a local area network. A usable IP network is automatically created without the need for configuration or special servers, but it is limited to a single subnet.

A peer to peer (P2P) based solution has the advantage of being distributed over a larger number of nodes in the network. For example, Chord [31] can be used to distribute objects evenly over a large number of nodes, but the location of a service description can be placed anywhere in the network. It is usually more efficient and robust to have services and descriptions at least close to each other. When a

personal network cluster gets disconnected from the infrastructure, all local services should still be available to the nodes in that cluster. Further a group or cluster of nodes has to be established before the system can be used. Also when all nodes have knowledge about all available services there will be problems with scalability. The Intentional Naming System (INS) [1,3] solves the scalability problem by separating sets of nodes in virtual spaces. Nodes are only aware of all services in their virtual space and have to rely on a directory server entity to find services in other virtual spaces.

For service discovery in personal networks, where we want to discover services located nearby, we need a fully distributed system, suitable for multi-hop wireless networks. Furthermore the system should work as soon as a new node joins, without the need to pre-establish a personal cluster. Not all nodes in the neighborhood are expected to be members of the same PN, they may belong to a different personal network or to some organization's network.

1) Bloom filters

Bloom filters were introduced in [5] as a hash coding technique that provides a trade-off between the space usage or hash size and the time needed to test the membership of a text string in a given set of strings. Several strings are represented in one set of bits. A small chance of false positives is allowed, that is a string is not a member of the given set while the system claims it probably is. A Bloom filter consists of an array of *w* bits, initially all set to 0. A number of *b* independent hash functions is used to map a text string to the Bloom filter. For every string represented by the Bloom filter *b* bits are set as specified by the hash functions. A false positive appears when a string is represented by bits already set by one or more of the other strings represented in the Bloom filter.

In the Bloom filter example given in Table 1 several services are represented. When a user wants to test whether a color printer is one of these services, a hash function will be used on the string "Color Printer". Suppose this hash function returns (0,3,6). This means the color printer is probably represented in this filter as the bits 0 , 3 and 6 are all enabled. When the hash function on the string "Camera" returns (1,4,5) this signifies that the camera service definitely is not a member of the set of service in the Bloom filter, as bit number 4 is false. As strings are added to the Bloom filter, more bits in the filter are set. Also the possibility of overlap in the bits that are set for specific services will grow with the number of strings the filter represents.

Table 1: Bloom filter example

0	1	2	3	4	5	6	7
1	1	0	1	0	1	1	0

2) A protocol using attenuated Bloom filters

For local service discovery in ad-hoc networks we propose to use attenuated Bloom filters [29]. They were introduced as a method to optimize the performance of location mechanisms especially when objects to be found are located nearby. An attenuated Bloom filter is an array of standard Bloom filters of depth *d*. Every row in the filter represents objects at a different distance, indicated by the number of hops. Every outgoing link will have a separate attenuated Bloom filter. This enables to select a link where an object most likely can be found, a matching link. Periodically broadcast packets are sent to all direct neighbors. The packets contain Bloom filters that represent the services reachable through the sending node. Figure 3 shows the actions taken when packets arrive. When a client wants to find a specific service it will check whether the service is available locally. If this is not the case the client will send a query packet to any link with a matching Bloom filter. Unless there is a false positive, the query packet will be forwarded to a matching service. The destination will send a response packet back along the path the queries followed in reverse order. When the client receives the response packet it can try to use the service.

Advantages of using Bloom filters are the simple computations and efficiency with space and bandwidth. It is a distributed system, suited for locating services in the vicinity that can be used to find and set up clusters. For details on the service discovery protocol, refer to [13]

```
switch (received packet){
  case broadcast:
    if (new information in packet){
      store received attenuated Bloom filter;
      for each (layer){
       combine attenuated Bloom filters from links;
      }
      send (broadcast packet);
    }
  case query:
    if (service locally available)
      send (response packet);
    else {
      for each (link L) {
        if (available through L)
          send (query packet to L);
          store link Q query was received from;
      }
    }
  case response:
    send (response packet to link Q);
}
```

Figure 3. Algorithm (Run by Each Node Independently)

B. Service discovery in PNs

The service discovery system described in the previous section will be used in PNs as follows: nodes located in the same local area will distribute the services they know of between each other. Some nodes with limited resources may only advertise services they offer to nodes in the vicinity. Services available in the network are not stored in these nodes, so they do not need to listen to advertisements of other nodes.

For this local service discovery mechanism to work, we only need to know our direct surrounding neighbors; there is no need yet for setting up routes or forming clusters. The system can be used to get information about the nodes in the

neighborhood to form a personal cluster. Another use is to find other clusters, belonging to other persons or organizations so we can form an extended cluster, see Section II. Nodes will advertise services they consider to be public to all neighbors. As soon as a PN cluster is formed, nodes in this cluster can communicate securely, see section IV, and all services are advertised within the cluster. Two persons can exchange information or services when they are in each others vicinity. For non public services a mechanism is needed to allow discovery and use of those services. There is not always a need to go through an infrastructure network that might not be available at all times. A personal cluster can also advertise services available in a user's personal network, but not necessarily locally, when the user needs it. E.g. a personal cluster can temporarily announce the user's calendar service that is located at home locally to make a new appointment with a person in the vicinity.

For locating other clusters belonging to the same personal network or to any other personal network, but not in the vicinity we will need another mechanism, possibly also based on Bloom filters. After a personal cluster has been formed it will notify a directory server located at home or at an Internet service provider of all services available in this cluster. Anybody trying to find a service in the personal network will contact the directory server that will give the location the personal cluster containing the requested service can be found. When the query arrives in the personal cluster, it will be handled as if it was a query for a local service.

IV. PN Security Architecture

Providing security for PN devices is a challenge. The resource constrained nature of many PN devices makes it impractical to use the majority of the current security algorithms which were designed for more powerful workstations. As a result, often for the sake of feasibility and efficiency, security is sacrificed. We believe that as technology advances and devices become more ubiquitous, strong security is necessary for a trustworthy system. The limited computational and energy resources of many PN devices indicates that any proposed solution must be simple and lightweight so that it does not create a performance bottleneck of its own. Therefore, symmetric cryptography is deemed to be the only feasible option [28, 18].

As energy is the scarcest resource in our system, security mechanisms must be selected based on their power consumption. Additionally, as communication uses the lion's share of available power, any overhead arising from the transmission of extra bits comes at a significant cost. Consequently our approach restricts key management activities in order to conserve energy. We believe that it is sufficient for PN devices to demonstrate group membership of a cluster rather than their individual identity. This improves efficiency by using one group key to verify cluster membership, instead of as many keys as members in the cluster which is required by individual authentication. This reduces key management overhead, such as the amount of processing required, the number of messages exchanged, authentication delay, storage space and search time.

A. *Cluster key*

The symmetric group key mentioned above, henceforth called the cluster key, is used to guard against unauthorized access that can degrade the QoS for PN users. All intra-cluster traffic is protected using a message authentication code derived from the cluster key shared by all cluster members. Once an un-clustered device is recognized as part of the PN, it receives the cluster key, enabling it to take part in secure communication. This cluster key is then periodically refreshed at an interval that depends on the security level required and the need to forcibly evict cluster members.

The aim of our security architecture is to provide integrity, authentication and availability for all cluster traffic. Devices append a message authentication code[1] calculated using the cluster key, to all traffic they generate. Consequently any other device receiving this traffic can verify that it was generated by another cluster member and not modified in transit by any un-trusted device. Unauthenticated traffic is not forwarded into the cluster.

Such a mechanism, while efficient, makes it difficult to identify any malicious behavior from inside the cluster. Although our design makes it the responsibility of the user to identify such malicious devices, such devices can be blacklisted once they are identified

Lastly, as the aim of our security architecture is the dependability of the communication infrastructure and not confidentiality, messages are not automatically encrypted. Even though our design does not exclude the option of confidentiality, it is difficult to justify confidentiality as a requirement for all traffic because it uses already scarce resources. Most applications that require confidentiality already encrypt their traffic end-to-end, so duplicating the same functionality at the lower layers is not efficient.

B. *The System*

1) *Security Agents*

In the context of our security architecture, we define a new role for a device, that of a security agent. Each cluster must have one device functioning as a security agent; however there is no restriction to any other role that such a device may serve. The security agent authenticates new devices joining the cluster, initiates periodic cluster key updates and generates cluster advertisements. Additionally, it is able to evict short-term members on demand and sets the cluster policy which lets other members know about cluster parameters like the frequency of cluster advertisements and key updates, etc.

In terms of security agent functionality, we define two new types of devices. Some devices have capabilities to function as security agents and others do not. When not connected to other devices, a Security Agent Capable (SAC)

[1] As the message authentication code increases the original packet size and thus the energy required for transmission, it should not be too large.

device will function as a security agent and is thus considered a special form of a cluster that only contains itself. Security Agent Incapable (SAI) devices, unable to act as security agents, do not constitute a cluster when alone and need to join existing clusters. SAI devices are less sophisticated (e.g. sensors) and typically not useful by themselves. They are designed to be used in conjunction with other smarter devices when networked in a cluster.

2) Cluster Discovery

Clustering, the process by which all PN devices within each others transmission range connect to extend a cluster is an integral part of our vision for a PN. Cluster discovery is done by listening for cluster advertisements, which are generated by the security agents. These advertisements are forwarded by other cluster members and thus propagate to the edges of the cluster, quite like a ripple on a pond. The periodicity of the cluster advertisements and the decision on which members take part in propagating them depends on the cluster policy.

Un-clustered devices periodically wake up to listen for such advertisements. When an un-clustered SAI device receives a cluster advertisement from a cluster belonging to its own PN, it will attempt to authenticate itself and to join that cluster. Some devices may be left powered up for extended periods either purposely or mistakenly by the owner of the PN. We do not want such (otherwise idle) devices to spend precious energy continuously advertising their existence to possibly non-existing neighbors. Clusters on the other hand, as shown by their interconnected state, are more active in nature. Therefore the responsibility of advertising is placed on them.

C. Cluster Dynamics

Apart from advertising the existence of a cluster to non-members, cluster advertisement also lets cluster members know that their cluster is alive. A cluster that is alive has a functioning security agent and can therefore grow by adding new members. Conversely, a zombie cluster is one that has lost its security agent but has a valid cluster key. Devices belonging to a zombie cluster can still communicate securely with each other, but the cluster cannot grow since there is no security agent to authenticate new members. Zombie clusters can only be resuscitated by the return and the resulting cluster advertisement generated by the original security agent.

In the absence of a security agent, as there is no one to update the cluster key, it will eventually expire. If devices have not joined new clusters in the meantime they can no longer communicate amongst each other. SAI devices enter the orphan state where they wait to join other clusters while SAC devices form their own clusters.

D. Device Authentication

Earlier we stated that devices wishing to join a cluster need to authenticate with the security agent of that cluster. We also said that members of a cluster only forward authenticated cluster traffic. This implies that for supplicant devices to authenticate with a cluster, they need to be within the transmission range of the security agent (belonging to the cluster they wish to join). Similarly, for two clusters to merge together, the two security agents also need to be within each others transmission range. Such a restriction on the extensibility of the cluster is not practical. We would like clusters to extend with devices that are within the range of even peripheral cluster members. Similarly, two clusters should be able to merge when their periphery overlaps and not only when the transmission range of the two security agents overlaps. To that end, cluster members enable IEEE 802.1X based port authentication.

Therefore besides authenticated cluster traffic (i.e. traffic protected with the cluster key) cluster members also accept unauthenticated EAP [33] requests which are forwarded to the security agent for authentication. EAP is an extensible protocol that can carry a variety of authentication mechanisms like shared keys, digital certificates etc. Predictably, devices that are not part of a cluster do not accept EAP requests.

However, the mechanism we propose for use has some important differences with IEEE 802.1X. For instance, after a successful authentication the supplicant no longer maintains any relationship with the authenticator. One consequence of this is that authorized EAP traffic does not go through the authenticator because devices can send further EAP messages (e.g. EAP logoff) to the security agent themselves. We also propose an extension which will allow complete clusters to merge instead of just permitting individual devices to join a cluster. For more details on the security architecture refer to [17].

V. QOS ASPECTS OF PNS

The ad hoc nature of PN brings many issues and difficulties for provisioning QoS, needed for real time and broadband applications. The main issues complicating QoS provisioning in PNs are:

- *Unpredictable link properties***:** Interference and signal fading make the media unpredictable.
- *Limited battery life:* Mobile devices have limited resources, so QoS must be power aware and power efficient.
- *Hidden and exposed terminal problem:* Nodes may cause collisions because they do not sense each other, or may unnecessarily block each other.
- *Node mobility:* The network topology can be dynamic, changing the links between nodes as they move in and out of each others transmission range.
- *Route maintenance:* Maintenance of state information is very difficult. Routes may break during data transfer, which calls for route recovery.

QoS measures such as available capacity and response times are therefore hard to guarantee. In particular, the last two issues show that robust routing protocols are needed to ensure certain levels of QoS. In general, there are three levels of QoS as depicted in Figure 4.

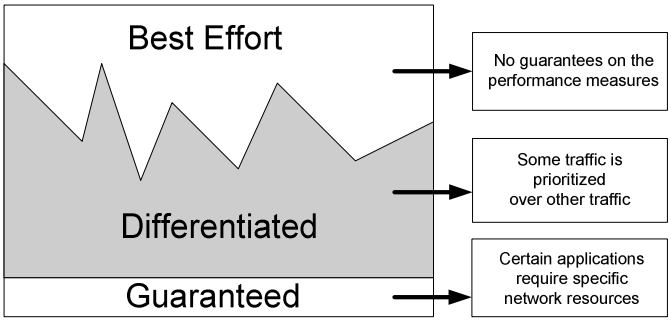

Figure 4: Levels of QoS

Best effort does not give guarantees on the performance measures. *Differentiated QoS* means that some traffic is prioritized over other traffic, giving it a statistic preference, but no hard guarantees. *Guaranteed QoS* means that resources are reserved so that certain performance measures are definitely met. In mobile ad hoc networks, due to the issues stated above, guaranteed QoS can not really be achieved; it can, at most, be 'approximated' by applying appropriate packet handling and resource management at the MAC layer in conjunction with sophisticated routing at the network layer.

A. QoS at the MAC layer

The MAC layer plays an important role in QoS provisioning. For achieving differentiated QoS, priority levels are assigned to packets from different applications. This differentiation is on a hop-by-hop basis, not end-to-end. More stringent QoS requirements can be met when, in addition, all nodes between sender and destination reserve resources for a (real-time) traffic flow. Obviously, this is more complex and requires appropriate cooperation with routing at the network layer, see Subsection V.B.

The basic IEEE 802.11 MAC protocol [37] for ad hoc networking is the Distributed Coordination Function (DCF). The DCF uses Carrier Sense Multiple Access with Collision Avoidance where all nodes sense if the channel is idle. Each node holds a contention window (CW), from which a random backoff time is taken. After the channel has been idle for a distributed interframe space (DIFS), the backoff timer is decremented and when it expires, transmission is initiated. First, a short RTS packet is sent after which a CTS packet is returned by the receiver to reduce the effect of the well known hidden terminal problem. When this handshake is successful, transmission of the data packet starts. In case of a collision the CW is doubled and the process repeats itself. The IEEE 802.11B standard [37] only provides best effort service. Some proposals have been made to extend the protocol with service differentiation.

The IEEE 802.11E MAC protocol [36] is the standardized packet scheduling approach to QoS provisioning in ad-hoc networks. To make the protocol QoS aware, IEEE 802.11E stations have different queues (access categories, ACs) for packets originating from applications with different service requirements. For all ACs different DCF parameter settings can be used, for instance a smaller contention window, DIFS size and allowing multiple packets to be sent after winning the contention.

Other approaches use a more explicit differentiation between the traffic classes in different nodes by exchanging information about the rank of their highest priority packet to synchronize their scheduling parameters e.g. [19,20]. An out of band approach is also possible to make fast reservations for the high priority traffic, see e.g. [34]. A centralized approach is possible as well where some nodes are chosen to coordinate access to the channel of the other nodes in their neighbourhood. This could for instance be done by the cluster heads in a PN. A polling scheme like the Point Coordination Function (PCF) from IEEE 802.11B can then be used [2].

B. QoS Routing in PNs

Delivering end-to-end QoS in mobile ad hoc networks is intrinsically linked to the underlying routing protocol and algorithm. Routing protocols capture the network state information and disseminate it throughout the network, while routing algorithms use this information to compute appropriate paths. Thus, the goal of QoS routing is to identify and utilize paths required to manage and support PN features such as distributed multimedia services, mobile users and networks, heterogeneous inter-networking, service guarantees, point-to-multipoint communications, real time applications, etc.

Much work has been done in the areas of QoS routing for static networks i.e., the networks with non-varying topology [4,23,25] and ad hoc routing [9,14,27]. But QoS routing in ad hoc networks is a challenging issue due to the changing topology and non-uniform propagation characteristics of wireless transmissions.

The solution of the static QoS routing problem can be considered as sufficiently solved to be useful in practice. In [25] SAMCRA is proposed, an exact QoS routing algorithm for static networks. But the solutions proposed for QoS routing in static networks are not straightforwardly extended to ad hoc networks. Most of the QoS algorithms for static networks assume the availability of precise state information (e.g., the probability distribution for link delay) besides the topology of the network [25]. In ad hoc networks, the topology and the link parameters e.g., available bandwidth, packet loss etc. are changing, although, the topology is changing on a slower time scale. Moreover, if the topology of an ad hoc network changes too fast, QoS routing may become impossible. Due to the inherent characteristics of the wireless medium in ad hoc networks, the available bandwidth is shared between the neighboring nodes. Thus, QoS routing in ad hoc networks is heavily dependent on how well the resources are managed at the medium access control (MAC) layer. Different MAC layers have different requirements for successful transmissions, and a QoS routing protocol developed for one type of MAC layer does not generalize to others easily.

Most of the proposed ad hoc routing protocols can be classified into reactive or on-demand (e.g. AODV [27]), proactive or table-driven (e.g. OLSR [9]), and hybrid (e.g. ZRP [14]) based on the information stored at the nodes and the route discovery mechanism. But all the current routing

protocols such as AODV [27], OLSR [9] and ZRP [14] are best-effort. They are targeted at finding a feasible route from the source to the destination without considering current network traffic or application requirements.

There has been some work to develop QoS routing algorithms and protocols for ad hoc networks [7,11,26,35]. QoS extensions have been proposed to both on-demand and table driven routing protocols [26,11]. Sivakumar *et al.* [30] have proposed a core-extraction distributed routing algorithm (CEDAR) for QoS routing in ad hoc networks. Zhu and Corson [35] have proposed an on-demand QoS routing protocol based on AODV for TDMA-based ad hoc networks. Since hard QoS, i.e. guaranteed constant bit rate and delay, is difficult to achieve for ad hoc networks, the aim of many QoS protocols such as the ticket-based algorithm proposed by Chen and Nahrstedt [7], and QoS OLSR [11] has been to develop soft QoS or better than best-effort services.

An alternative solution to the problem of QoS routing is the AntNet algorithm [6]. In AntNet, the network topology and the end-to-end delays for different paths are represented by probabilistic routing tables. The probabilistic routing tables are updated by the mobile agents (control packets) depending on the end-to-end delay. The data packets travel using probabilistic routing tables leading to load-balancing or multi-path routing. Due to the inherent characteristics of AntNet, no additional routing protocols and algorithms are required. AntNet has been shown to provide load balancing and it performs well under heavy traffic conditions [6]. Further investigation of AntNet [10] shows that AntNet performs well for small static networks with sparse topologies. But the performance of the AntNet algorithm for ad hoc networks is an open issue and needs further investigation.

There are multiple challenges in developing QoS protocols and algorithms for PNs. The study of areas such as connectivity of the ad hoc networks, existence of multiple paths and the stability of paths is critical for developing QoS routing algorithms. The development of QoS routing protocols for PNs also presents multiple challenges. Algorithms such as random walks rather than constrained flooding should be used for discovering data and optimal paths in the wired infrastructure-based networks [22]. Both table-driven and on demand routing protocols have their advantages and disadvantages and cannot be universally applied to all networks. Hence, a flexible routing protocol may be needed. On-demand routing protocols such as AODV, DSR use flooding for information dissemination leading to a large overhead. Efficient schemes such as flooding with self pruning or dominant pruning [33] may need to be employed for reducing the overhead in information spreading. The issue of scalability for current ad hoc routing protocols such as AODV, OLSR, and DSR needs to be addressed. Finally, due to the interdependence between MAC layer and routing protocols, a cross layer design may provide an effective solution to QoS in ad hoc networks.

VI. CONCLUSION

This paper proposed a self-organized personal network architecture taking into consideration both QoS and security aspects. Solutions for a secure architecture, service discovery, security mechanisms and QoS support are presented. In achieving self-organized PN architecture, a master node is introduced for local management of each personal cluster and a hierarchical structure is considered appropriate for ad hoc communication of personal networks. For the security architecture we use secure but lightweight mechanisms suitable for resource constrained devices and wireless communication. Our proposal uses pair-wise keys for secure cluster formation and group keys for securing intra-cluster communication. The service discovery in personal networks is done by making efficient use of attenuated Bloom filters. Nodes can find services and other nodes or clusters belonging to a personal network in the vicinity, in order to interact with them and form clusters. Finally, by distinguishing between different access categories, differentiated QoS can be reached for higher priority traffic. For QoS routing in Personal Networks, a flexible protocol will be needed that can handle the dynamic nature of ad hoc networks.

Multiple issues still exist for the different aspects involved. As PNs introduce new design challenges due to the heterogeneity of the involved technologies, the need for self-organization, the dynamics of the system composition, the application-driven nature, the co-operation with infrastructure-based networks, and the security hazards, much work needs to be done before we can claim that the problem of designing QoS-aware Personal Networks is sufficiently solved.

ACKNOWLEDGEMENTS

This research is partly supported by the Dutch Ministry of Economic Affairs under the Innovation Oriented Research Program (IOP GenCom, QoS for Personal Networks at Home).

REFERENCES:

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, “The design and implementation of an intentional naming system”, *Proc. 17th ACM SOSP Conf*, 1999.

[2] J.N. Al-Karaki and J.M. Chang, “Quality of service support in IEEE802.11 wireless ad hoc networks”, *Ad Hoc Networks 2*, pp. 265-281, 2004

[3] M. Balazinska, H. Balakrishnan, and D. Karger, “Ins/twine: A scalable peer-to- peer architecture for intentional resource discovery”, 2002.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services”, *IETF RFC 2474*, 1998.

[5] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors”, *Communications of the ACM*, vol.13, no.7, pp 422-426, 1970.

[6] G. Di Caro, and M. Dorigo, “AntNet: Distributed Stigmergetic Control for Communication Networks”, *Journal of Artificial Intelligence Research* 9, 1998.

[7] S. Chen, and K. Nahrstedt, “Distributed Quality-of-Service Routing in Ad Hoc Networks”, *IEEE Journal on Selected Areas in Communications*, vol. 17, No. 8, pp. 1488-1505, 1999.

[8] S. Cheshire, B. Aboba, E. Guttman, “Dynamic Configuration of IPv4 Link-Local Addresses”, *RFC 3927*, May 2005.

[9] T. Clausen Ed., and P. Jacquet Ed., “Optimized Link State Routing Protocol (OLSR) ”, http://www.ietf.org/rfc/rfc3626.txt, October 2003.

[10] S. S. Dhillon, and P. Van Mieghem, “Performance Analysis of AntNet algorithm”, (submitted to *Computer Networks*).

[11] Y. Ge, T. Kunz, and E.F. Roberts, “Proactive QoS Routing in Ad-Hoc Networks”, *2nd International Conference on Ad-Hoc Networks and Wireless*, Montreal, October 8-10, 2003.

[12] Y. Ge, L. Lamont, and L. Villasenor, “Hierarchical OLSR – A Scalable Proactive Routing Protocol for Heterogeneous Ad Hoc Networks”, IEEE Communication Magazine,July, 2005.

[13] P. Goering and G. Heijenk, “Service Discovery Using Bloom Filters”, *Proc. Twelfth Annual Conference of the Advanced School for Computing and Imaging*, Belgium, June 14-16, 2006.

[14] Z. J. Haas, and M. R. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, Internet Draft, draft-ietf-manet-zone-zrp-02.txt, June, 1999.

[15] J. Hoebeke, I. Moerman, and B. Dhoedt, “Analysis of decentralized resource and service discovery mechanisms in wireless multi- hop networks”, *Proceedings WWIC 2005*, Xanthi, Greece, May 11-13 2005.

[16] X. Hong, et al, “Scalable Ad Hoc Routing in Large, Dense Wireless Networks Using Clustering and Landmarks”, IEEE Internationl Conference on Communications, 2002.

[17] A. Jehangir, and S.H. de Groot, “A Security Architecture for Personal Networks”, (in submission).

[18] C. Karlof, N. Sastry, and D. Wagner, “TinySec: A Link Layer Security Architecture for Wireless Sensor Networks”, *The Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, Baltimore, Maryland, USA, November 3-5, 2004.

[19] V. Kanodia, C.Li, A. Sabharwal, B. Sadeghi, E. Knightly, “Distributed priority scheduling and medium access in ad-hoc networks”, *ACM Kluwer, Wireless Networks 8 (5)*, pp. 455-466, 2002.

[20] C.R. Lin and M. Gerla, “MACA/PR: an asynchronous multimedia multi-hop wireless network”, *Proceedings of IEEE Infocom’97*, 1997.

[21] W. Lu, A. Lo, I.G. Niemegeers, "Session Mobility Support for Personal Networks Using Mobile IPv6 and VNAT", 5th Workshop on Applications and Services in Wireless Networks (ASWN’05), Paris, 2005.

[22] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, “Search and Replication in Unstructured Peer-to-Peer Networks”, *Proc. 16th ACM International Conference on Supercomputing*, 2002.

[23] P. Van Mieghem, H. De Neve, and F. Kuipers, “Hop-by-hop Quality of Service Routing”, *Computer Networks*, vol. 37, No. 3-4, pp. 407-423, 2001.

[24] I. G. Niemegeers and S. M. Heemstra de Groot, “Research issues in ad hoc distributed personal networking,” *Wireless Personal Communications*, vol. 26, no. 2-3, pp. 149–167, August 2003.

[25] P. Van Mieghem, and F. Kuipers, “Concepts of Exact Quality of Service Algorithms”, *IEEE/ACM Transactions on Networking*, vol. 12, No. 5, pp. 851-864, 2004.

[26] C. Perkins, and E. Belding-Royer, “Quality of Service for Ad hoc On-Demand Distance Vector Routing”, *IETF Internet draft*, 14 October 2003.

[27] C. Perkins, E. Belding-Royer, and S. Das, “Ad Hoc On-Demand Distance Vector (AODV) Routing”, http://www.ietf.org/rfc/rfc3561.txt, July 2003.

[28] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.D. Tygar. “SPINS: Security protocols for sensor networks”, *The Seventh Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 16-21, 2001.

[29] S. C. Rhea and J. Kubiatowicz, “Probabilistic location and routing”, *Proceedings of INFOCOM 2002*, 2002.

[30] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm”, *IEEE Journal on Selected Areas in Communications*, vol. 17, No. 8, August, 1999.

[31] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, “Chord: A scalable Peer-To-Peer lookup service for inter- net applications”, *Proceedings of the 2001 ACM SIGCOMM Conference*, pp 149-160, 2001.

[32] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, “Service location protocol”, RFC 2165, 1997.

[33] Q. Wang, M. Abu-Rgheff and A. Akram, Design and evaluation of an integrated mobile IP and SIP framework for advanced handoff management, *ICC*, 2004.

[34] X. Yang and N. Vaidya, “Priority scheduling in wireless ad-hoc networks”, *Proceedings of the 3rd ACM International symposium on Mobile ad-hoc networking and computing*, Lausanne, Switserland, 2002.

[35] C. Zhu, and M. Scott Corson, “QoS routing for mobile ad hoc networks”, *Proc. IEEE Infocom,* vol. 2, pp. 958-967, 2002.

[36] P802.11E/D11.0, Draft amendment to standard for telecommunications and information exchange between systems – LAN/MAN Specific requirements – part 11: Wireless Medium Access Control and physical layer specifications: Medium Access Control Quality of Service Enhancements, February 2004.

[37] IEEE part 11: Wireless LAN Medium Access Control and Physical Layer specifications, Institute of Electrical and Electronics Engineers Inc., 1997.