University of Twente

Telefónica

Possibilities of Peer-to-Peer Technology in Network Management

Master Thesis

Author:	Martijn Frints	
Date:	December 1, 2006	
University:	University of Twente, The Netherlands	
Department:	Computer Science Design and Analysis of Communication Systems (DACS)	
Graduation Committee:	Dr. Ir. Aiko Pras	University of Twente
	Tiago Fioreze, M.Sc.	University of Twente
	Pablo Arozarena, B.Sc., MBA.	Telefónica Investigación y Desarrollo

Acknowledgements

This thesis is the result of my Master assignment performed at Telefónica Investigación y Desarollo, Madrid, Spain. It is the final part of the Telematics course at the faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) at the University of Twente, the Netherlands.

I would like to thank my supervisors Tiago Fioreze, M.Sc and Dr. Ir. Aiko Pras of the Design and Analysis of Communication System (DACS) chair at the University of Twente for their valuable comments and guidance which enabled me to constantly improve my thesis. In addition, I would like to thank my supervisor and manager Pablo Arozarena of Telefónica Investigación y Desarrollo for the trust, the freedom and the opportunities given to me.

I also would like to thank my fellow students and friends for their support and friendship throughout the years at this University. I consider myself lucky for having the pleasure of knowing all of you.

Finally I would like to thank my parents for giving me all the chances I had, for supporting me, for believing in my capabilities and for never doubting me.

Thank you.

Abstract

The traditional centralized network management approach has proven to be less suitable for managing large, dynamic environments due to scalability and flexibility issues. As a consequence, distributed network management approaches have been investigated in order to overcome these issues. The distributed management approach used in the Celtic Madeira project is based on peer-to-peer principles and designed for Wireless Mesh Networks. Based on experience gained in this project, the goal of this thesis is to show the possibilities of using peer-to-peer technology in the field of network management. As is going to be presented, this technology has promising advantages as, for example, lack of a single point of failure, improved scalability behaviour and robustness to changes in the topology are discovered, but there are still some areas (e.g. security) that require further attention.

Keywords: Peer-to-peer technology, network management, Wireless Mesh Networks, centralized and distributed management approaches

List of Abbreviations

CA	Certifying Authorities
CMIP	Common Management Information Protocol
IETF	Internet Engineering Task Force
MbD	Management by Delegation
MIB	Management Information Base
NBI	Northbound Interface
NMS	Network Management System
OASIS	Organization for the Advancement of Structured Information Standards
OSI	Open Systems Interconnection
OSS	Operations Support System
P2P	Peer-to-Peer
PGP	Pretty Good Privacy
PKI	Public Key Infrastructures
SNMP	Simple Network Management Protocol
SSID	Service Set Identifier
TMN	Telecommunications Management Network
UDDI	Universal Description, Discovery and Integration
WMN	Wireless Mesh Network
XML	eXtensible Markup Language

Contents

List of Figures

1 Introduction

Computer networks are getting larger and more heterogeneous. In order to reduce operating expenses, research on automated network management is required. In addition, the growing popularity of wireless communication like WiFi networking introduces new and more sophisticated management requirements. Ad hoc technology for example enables networks to be set up without existing infrastructure while providing a high degree of mobility for its users.

In the above context, the use of traditional, centralized management techniques suffers a number of disadvantages like static architectures and rigid standards, which make them less suitable for these large and dynamic environments.

Research on next-generation management technology is required, together with an overall framework for network-wide optimal configuration according to well-defined goals and constraints. Approaches based on network programming, management overlays and peer-to-peer computing, have been recently proposed to engineer management systems with improved scalability behaviour and that are robust regarding topology changes and failures.

This thesis researches the possibilities of applying peer-to-peer principles to overcome the drawbacks of present day management approaches, like scalability. It focuses on the work done in the Celtic Madeira project, in which a management solution based on peer-to-peer technologies has been designed for Wireless Mesh Networks. This management solution facilitates self-management and dynamic behaviour of nodes within the network.

1.1 Research questions

The main research question that this thesis aims to answer is formulated as follows:

> **What are the main advantages and disadvantages of using peer-to-peer technology in the field of network management?**

In order to answer this question, and to narrow down the field of research, a number of sub questions are formulated. These sub questions guide the research process and will be answered in the conclusion in section 5.

1. What different approaches to network management exist?
2. In which category does network management based on peer-to-peer principles fit best?
3. Is it possible to set up a network management system based on peer-to-peer principles without pre-configuration or user intervention?
4. How can one acquire management information about the whole network from a single point in a completely distributed management framework?

1.2 Research approach

The approach describes the steps that will be taken to answer the research questions defined in the previous section 1.1.

1. The sub question "*What different approaches to network management exist?*" will be answered by conducting a literature study. With the information acquired in this study, a categorization of network management approaches will be established.
2. The sub question "*In which category does network management based on peer-to-peer principles fit best?*" will be answered by combining the information acquired for the previous question with information about peer-to-peer principles.
3. The sub question "*Is it possible to set up a network management system without pre-configuration or user intervention*" will be answered by using personal experience acquired in the design, implementation and testing process, and by evaluating the Madeira prototype.

4. The sub question "*How can one acquire management information about the whole network from a single point in a completely distributed management framework?*" will be answered by developing an extension for the Madeira framework. This will then be implemented and integrated in the Madeira prototype.

1.3 Intended audience

This work is intended for readers with an interest in network management. Basic knowledge about networks and management of networks is required.

1.4 Structure of this thesis

This document is structured as follows. Chapter 2 first provides a basis on how to categorize network management approaches, followed by a description of current approaches according to this categorization. An introduction into peer-to-peer technology and its relationship with ad-hoc networks is given in chapter 3. Chapter 4 starts with a general description of the Madeira peer-to-peer management solution, followed by the design of the Northbound extension and the Operations Support System. It also contains an elaboration on the advantages and disadvantages of the presented approach. Finally, in the conclusion in chapter 5, the research question will be answered and some ideas on future work are presented.

2 State of the art

A network management system (NMS) consists of managers, agents and dual-role entities with both manager and agent capabilities. In conventional systems, a manager is in charge of performing management functions whereas an agent normally performs simple tasks such as acquiring data. Nowadays, however, agents are performing tasks with more responsibilities, blurring the clear distinction between managers and agents. Because of this, a definition separating manager functions from agent functions is difficult if not impossible. Therefore, [ScQu00] rather defines a manager as an entity that needs to communicate with other entities to perform its tasks whereas an agent can execute its assigned tasks alone. For dual-role entities, the same definition holds: in agent role, it performs its tasks on its own whereas in management role, it needs to cooperate with agents. This can be seen in Figure 1, also derived from [ScQu00].

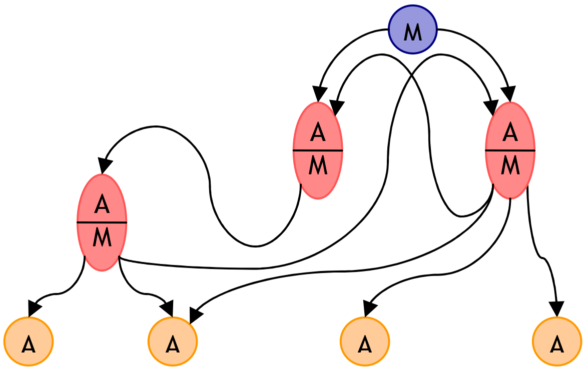


Figure 1: General model of a management approach

For this thesis, the same classification of network management systems will be used as in [ScQu00], where an overall distinction of systems is made based on the level of distribution of management. Let *m* be the number of managers in a network with *n* network elements, i.e., agents and managers:

1.	$m = 1$:	centralized management;
2.	$1 < m << n$:	weakly distributed management;
3.	$1 << m < n$:	strongly distributed management;
4.	$m \approx n$:	cooperative management

As can be seen, this classification is based purely on the number of managers in the network, and the ratio between this number and the total number of network elements. A different method is used by [ChLi02], where network management approaches are divided based on the mobility of management functionalities. This leads to a similar total of four categories:

1. client-server approach, where one centralized NMS polls *n* network elements;
2. hierarchical static approach, that consists of one top level manager, *k* mid-level managers (or dual-role entities), and *n* network elements. Each mid-level manager manages a separate subnet with an average of *n*/*k* elements;
3. weak mobility approach, where the NMS distributes code to specific network elements where the code is executed. Typically, the results are transmitted back to the NMS;
4. strong mobility approach, where the NMS distributes code to one or more network elements. Unlike the weak mobility approach, these agents can travel among different network elements to fulfil their task.

The following section contains a more detailed exploration of the first four categories, given by [ScQu00], and their relationship with the classification made by [ChLi02]. It has to be said that the line between the different approaches is blurry. It is difficult to make a clear distinction between the different approaches. The characterisation should thus be seen as ‘typical’ rather than ‘strict’.

2.1 Centralized approach

2.1.1 Characteristics

Communication networks are traditionally managed by centralized management systems. As mentioned before, a centralized management system typically contains one manager. This manager may control a potentially large number of network elements by manipulating local management agents [ScQu00][Sub00]. One can view these systems as client-server approaches where the network management system is a client to these management agents [ChLi02]. The figure below gives an example structure.

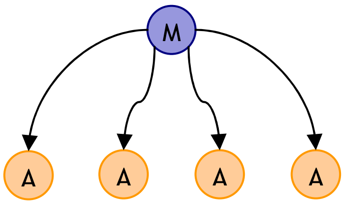


Figure 2: Example for a centralized approach

According to the categorization of management systems made above, a pure centralized approach consists of exactly one manager which manages multiple agents. It is interesting to notice that such a system does not contain dual-role entities.

Most of today's computer networks are managed by Simple Network Management Protocol, or SNMP. The first release, SNMPv1, is described by the Internet Engineering Task Force (IETF) in [RFC1157] in 1990 and became hugely popular. Already in 1993, almost every IP device had an embedded SNMP agent [MaZn97]. SNMP standardized network management, eliminating the need for multiple, incompatible, management platforms. The NMS polls the agents on a regular basis to acquire information. It can also modify settings in and receive trap messages from the agents. All the management 'intelligence' is concentrated in the NMS, whereas an agent can be seen as merely a simple data collector. SNMPv1 has been superseded by SNMPv3 [RFC3410] which has, among others, improved security features.

2.1.2 Disadvantages

The usage of a centralized network management approach has a considerable number of disadvantages. The most important ones are described below:

- Using a central entity that is in charge of polling all network elements has a negative effect on the scalability. A large amount of nodes will lead to a large polling load of the network. Especially links near the NMS will have to handle all these requests and responses [LiTh03]. Furthermore, since all management logic is concentrated in the NMS and the agents simply collect the data, only the NMS can analyse all the data. As proven by [ChLi02], both a traffic bottleneck and a processing bottleneck at the NMS will then be unavoidable.

- Using a single server that is in charge of network management and contains all management logic introduces a single point of failure. Naturally, redundancy can be created by server replication for example, but this is usually a costly operation [AnDa95].

- In situations where the central entity becomes unavailable, due to link failures or a (temporal) outage for example, management operations will not be executed. Centralised architectures are not well suited to support disconnected operation [LoOl00].

The main problem in the centralized approach is scalability. The increase of the size of networks in the 1990s and the wide availability of the IP stack on network devices influenced the need for other approaches [MaZn98].

2.2 Weakly distributed approach

2.2.1 Characteristics

The main characteristic of weakly distributed models is that network management processing is divided over a number of top-level and mid-level managers. These mid-level managers are in charge of managing a smaller part of the network. Typically there is minimal communication between mid-level managers on the same level, as can be seen in Figure 3 below. Delegating management functions elsewhere decreases the polling and computational load per manager and the communicational load on the network [ScQu00][MaZn98]. Similar to the aforementioned centralized approach, agents are considered to act as data collectors.

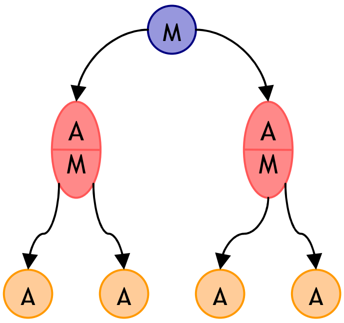


Figure 3: Example for a weakly distributed approach

Initial and well established frameworks in the telecommunication networks such as the OSI System Management (OSI-SM) use the Common Management Information Protocol (CMIS/CMIP). This communication protocol between the management system and the managed system can support both a centralized model with one manager that processes all data, and a hierarchical model with mid-level managers [ChLi02]. OSI System Management is also used as the base for the Telecommunications Management Network (TMN) [Sub00]. TMN was defined by ITU-T as a framework for the management of communications networks in ITU-T M.3000 recommendation series.

TMN adopts the manager-agent paradigm for the communication between systems and network equipment. TMN is divided into five layers, depicted in the pyramid in Figure 4 [PrBe99]:

- Business Management takes care of the management of the whole enterprise. It can be seen as goal setting rather than goal achieving. This makes Business Management more strategic and tactical management rather than operational management like the other layers described below
- Service Management is in charge of managing aspects that are directly observed by the users of the telecommunication network. It addresses topics like customer care, service development and operation, Quality of Service management, accounting, etc.
- Network Management provides management services that are related to the interaction between multiple pieces of equipment. This layer can generate a complete network view, create dedicated paths, detect faults and optimize network performance.
- Element Management takes care of vendor specific management functions, which are hidden from the Network Management layer above. For example, it can detect equipment errors and perform measurements on power consumption, temperature and resource usage.
- Network Element provides agent services, mapping the physical aspects of the equipment to the TMN framework.

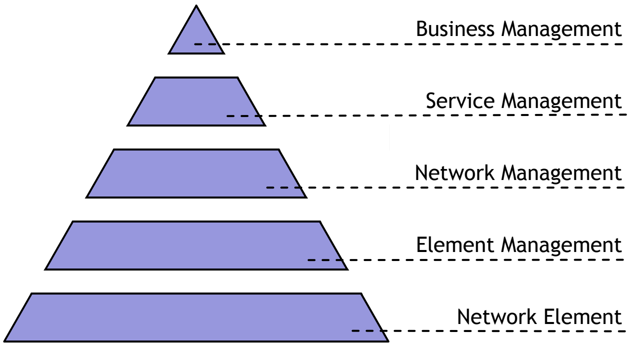


Figure 4: TMN Pyramid

Models as TMN imply a hierarchical structure with the degree of generalisation increasing to the top and, vice versa, the degree of specialisation increasing to the bottom layers. Every layer has a defined interface to its neighbouring layers.
Fault events travel upwards from the network elements and are filtered and correlated by the management layers. Configuration Management is command-response based, with commands being propagated down from service management to network elements and responses going in the opposite direction.

The Distributed Management Working Group of the IETF has developed an architecture where a main manager is able to delegate control to several distributed management stations, improving the scalability [LoOl02]. These distributed managers allow management operations to be executed even when the main manager is (temporarily) unavailable. If access to this central entity is not possible, each distributed manager may handle critical situations locally [LoOl02]. The delegation of management functions to distributed managers is offered by the Script MIB. These management functions are defined as executable code, or scripts.

2.2.2 Disadvantages

Although the weakly distributed approach eliminates some of the disadvantages of the centralized approach, it does not solve everything:

- Using mid-level managers has a positive effect on scalability when comparing the weakly distributed approach to the centralised approach. Each mid-level manager is responsible for a small part of the network. Although this causes less processing and traffic bottlenecks, the benefits have to be weighted against the cost of deploying enough mid-level managers [ChLi02].

- Although multiple managers are used, the hierarchical static configuration still implies a single point of failure at the top. Using multiple top-level, redundant, managers could be expensive, similar to centralized approach as indicated [AnDa95].

- Lack of robustness. If the connection between an agent and a manager is lost, the agent cannot take action in case of an emergency [MaZn98]. Because of this problem, weakly distributed models are less suitable for large, dynamic network environments.

2.3 Strongly distributed management

2.3.1 Characteristics

In the previous two approaches, agents have been functioning as simple data collectors. As was identified considerable time ago by, among others, [Well96], network devices can do much more than just running a simple agent, and could even be capable of managing themselves if given the opportunity. The Management by Delegation (MbD) framework, defined by Goldszmidt in [GoYe95], was the first to demonstrate the potential of task distribution [MaZn97]. Developed to overcome the key limitations of other models, it enables pushing management functions closer to target managed entities and, in doing so, decreasing the management traffic on a central point, and distributing the management decisions along the managed network. These management functions can be transferred and remotely executed by managed agents, which gives them more autonomy. Management tasks are therefore no longer only performed in the upper layers. Figure 5 shows that midlevel manager communicate with each other to accomplish management tasks, which can be seen as an important difference with the weakly distributed management approach.

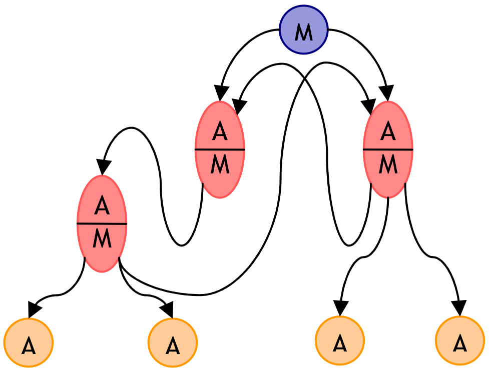


Figure 5: Example for a strongly distributed approach

The MbD approach is an example of weak mobility [ChLi02]. After execution, a delegated agent does not migrate to another host. In a strong mobility approach, code, execution state and maybe even data can move from one host to another. This is the case in Mobile Agent technology [KoXu02]. Mobile Agents are small software programs that ‘travel’ through the network. These agents decentralize processing and control by remotely performing certain management tasks [BiGu04]. They can be launched into an unstructured network and travel around to gather information, making them suitable for rapidly evolving networks [Sam04]. The IETF Script MIB, already mentioned in the weakly distributed management section above, can also be used in strongly distributed management systems, by employing scripts as agents for example [ScQu00].

2.3.2 Disadvantages

Although strongly distributed management solves a number of problems that exists in centralized and distributed approaches, it has some drawbacks:

- According to [ChLi02], Mobile Agents have the potential to perform well for monitoring purposes, but the additional overhead required for strong security is substantial. This need for a highly secure agent execution environment results in performance and functional limitations, as indicated in [ScEy00]. This secure execution environment could for example restrict access to certain resources, reducing the agent’s capabilities [HaCh95].

- Although agents are more than mere data collectors, they are in a way ‘unintelligent. They receive and execute programs from a manager, but they do not know the goal that the manager is pursuing [MaZn98].

2.4 Cooperative management

2.4.1 Characteristics

As the name suggests, cooperation is the main characteristic of the cooperative management. A cooperative model contains a large amount of dual-role entities. These entities, acting both as managers and agents, work together to fulfil management tasks. In general, a cooperative model has no, or at least a small amount of, entities that are just simple agents as data collectors. An important characteristic of this model is the fact communication is not restricted to a fixed hierarchy.

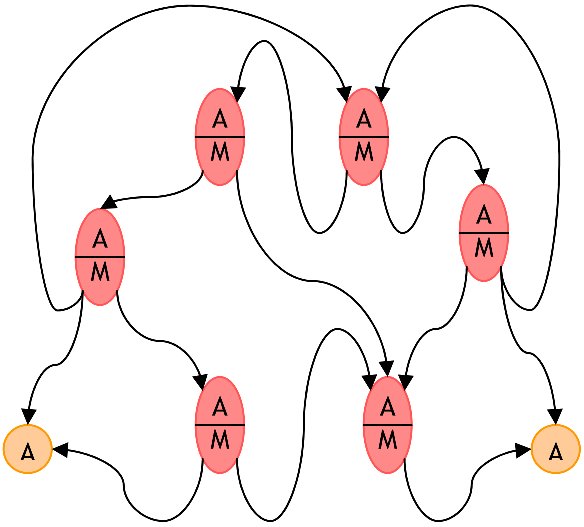


Figure 6: Example for a cooperative management approach

An important disadvantage of strongly distributed management is the fact that agents perform their tasks without knowing the goal. As pointed out in [MaZn98], cooperative management aims to solve this problem by using ‘intelligent agents’ and informing them about the goal while expecting agents to know how to achieve it. An exact definition of such an agent is difficult to determine. In both [MaZn98] and [TsSo00], definitions of the characteristics of intelligent agents in literature are summarized. Because of a lack of consensus on the exact definition of such agents, [MaZn98] identified the following core properties for intelligent agents:

- <u>reactive</u>: an intelligent agent can respond in a timely fashion to changes in its environment;
- <u>pro-active, goal-oriented</u>: an intelligent agent should be able to take initiative to achieve its goals, rather than solely react to external events;
- <u>autonomous</u>: an intelligent agent is supposed to be able to operate without direct intervention of humans. Moreover it should have some kind of control over its actions and internal state;
- <u>cooperative (communicative, coordinating)</u>: intelligent agents cooperate and communicate with other intelligent agents to achieve their goals;
- <u>temporally continuous</u>: an agent is a continuous running process. This feature distinguishes intelligent agents from mobile agents that are mentioned in the strongly distributed approach in section 2.3. Unlike mobile agents, intelligent agents can travel between network elements to fulfil their tasks, making them an example of the strong mobility approach [ChLi02].

2.4.2 Disadvantages

The following disadvantages have to be considered concerning cooperative management:

- The higher degree of ‘intelligence’ in cooperative management approaches makes such systems more complex to implement than the previous models [MaZn98].

- When network entities are able to autonomously perform management tasks and capable of reacting on changes in their environment, this increases the requirements on available resources as processing power and memory, for example. A result could be an increase of the cost per device [MaZn98].

- Intelligent agents are mobile and need a secure execution environment. Similar to the strongly distributed approach, these requirements could lead to performance and functional limitations indicated by [HaCh95] and [ScEy00].

Comparing the four approaches mentioned above, the centralised approach contains important disadvantages concerning scalability and robustness. The weak mobility approach solves these problems to some extent, but still contains important disadvantages as a single point of failure. The strong mobility approach contains better characteristics concerning scalability and robustness, but at the cost of increased security requirements. In a cooperative approach, these requirements might even be higher because of a bigger autonomy and high degree of intelligence, making these systems more complex than the other models. In the following section, peer-to-peer technology, which follows some principles of strongly distributed management and cooperative management approaches, is going to be presented.

3 Peer-to-Peer technology

3.1 Introduction

Peer-to-Peer networking has become more and more popular the last few years. This popularity is mainly fuelled by the use of file-sharing applications like Napster [Nap00], Gnutella [Gnu00] and Kazaa/FastTrack [Sha02]. Besides file-sharing also other solutions are possible using Peer-to-Peer principles. A well-known example is Skype [BaSc06], an application that uses a proprietary Peer-to-Peer VoIP protocol, but also distributed storage solutions like OceanStore [RhEa03] are being researched.

These peer-to-peer based solutions form (virtual) overlay networks on top of existing network infrastructure. Typically, there is no direct link between this underlying infrastructure and the overlay network. Neighbouring peers in the overlay network are thus not necessarily neighbours in the lower layer.

As already mentioned in the previous section, the centralised management approached is considered to be least suitable option for large scale and dynamic environments. Although the weakly distributed approach has better characteristics, it still has important disadvantages concerning flexibility, scalability and robustness. Peer-to-peer systems are generally characterised by their large scale and the continuous joining and leaving of nodes [OvPo02][Män05]. The decentralised nature and the promising behaviour concerning scalability and robustness make peer-to-peer technology an interesting area to explore its possibilities for the field of network management.

3.2 Peer-to-Peer basics

A number of definitions exist on what peer-to-peer networking exactly is. Instead of the client-server model, nodes in a peer-to-peer system can talk directly to each other without interference of a central entity. According to [DrRo01], peer-to-peer systems can be characterized as distributed systems in which all nodes have identical capabilities and responsibilities and all communication is symmetric. In [Scho01], Schollmeier indicated that some define peer-to-peer networks as a collection of heterogeneous distributed resources which are connected by a network, whereas others define it simply as the opposite of Client/Server architectures.

In a P2P system, the nodes have a significant or total degree of autonomy from central servers. As pointed out by [Shi00], P2P systems enable the utilization of previously unused resources such as storage, cycles or content, for example by tolerating and working with the variable connectivity of numerous devices.

An overall characteristic of a peer-to-peer network is that the nodes can send and receive information in a way that makes them both servers and clients, or ‘servents’. In both [Scho01] and [LuCa03], a distinction is made between pure peer-to-peer networks and hybrid peer-to-peer networks, as described below.

- Pure P2P architectures are completely decentralized. There is no central server or router. Each node can issue and respond to requests, or route requests to other nodes.
- In Hybrid P2P architectures, more types of nodes exist. The leaf nodes are nodes with an information need or information resource. In other words, they can provide information to or request information from other leaf nodes. Another type of nodes, super peers, has a more ‘server-like’ role in the network. These nodes provide regionally centralized services to the network in order to improve the routing of information requests. In [LuCa03], these nodes are called directory nodes or ultra peers. Each directory node provides directory services for portions of the network and directory nodes work in a cooperative manner to cover the whole network.

Comparing these two different peer-to-peer architectures, similarities arise with the strongly distributed and cooperative management approaches. Similar to nodes in these two management approaches, peers have a high degree of autonomy. The lack of hierarchy in pure peer-to-peer systems resembles the communication structure in a cooperative management approach, while the usage of super peers is more similar to a strongly distributed management approach. Figure 7 gives an example of both types of peer-to-peer networks.

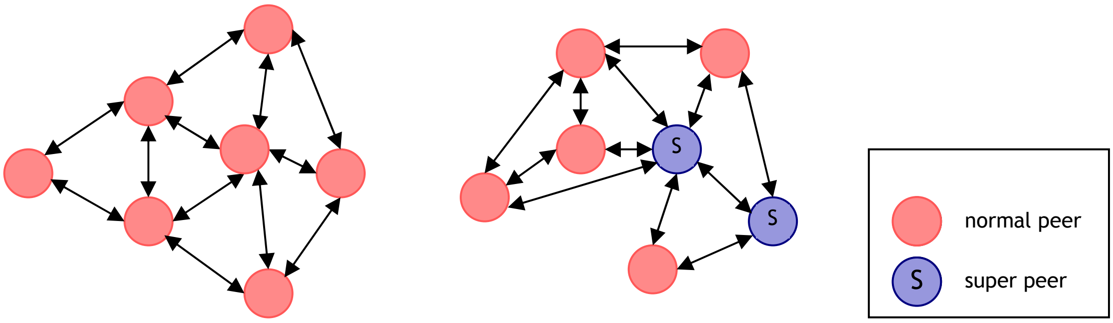


Figure 7: Pure peer-to-peer (left) and hybrid peer-to-peer (right)

3.3 Related work

Peer-to-peer principles are being researched in multiple network management projects, indicating the interest in combining both technologies.

- The Ambient Network (AN) program, co-funded by the European Union through the 6th Framework Programme, aims to provide fast network cooperation between different networks in order to provide users with the services they want, irrespective of their location. An Ambient Network is defined as a connectivity network with a number of similar characteristics like mobility and security for example. Peer-to-peer technology is used to organize the ambient networks into a dynamic hierarchical structure creating a hierarchical overlay network. An overlay is a set of peers belonging to one Ambient Network, and is represented by a super peer. This super-peer is responsible for negotiations with other overlays. Super-peers may form new peer-groups at a higher hierarchy level. Each AN can offer specific management services to other domains, like Quality-of-Service or security [BrGa05].
  Although the idea of this overlay network with super-peers is quite similar to the clustering overlay with cluster heads in Madeira, the AN project has a bigger scope than Madeira. The AN program is focused on the cooperation between different networks, while Madeira provides a distributed management solution for wireless mesh networks.

- Also in research on management of optical networks, peer-to-peer technology has gained attention. End-to-end optical circuits can lead to advanced services like bandwidth on demand and optical virtual private networks [PiJu06]. [PiJu06] proposes a solution to facilitate user-driven provisioning requests within one or more domains, where users or applications can manage/control or even own network resources like bandwidth. A management domain is defined as a collection of managed objects which are grouped to meet organizational requirements according to geography, technology, policy or other structure, typically for the purpose of providing control in a consistent manner. Each domain is managed by a Distributed Optical Manager, or DOM. DOMs are comparable to traditional mid-level managers. Peer-to-peer technology is used to forward information between DOMs. The DOMs form a structured overlay network based on Distributed Hash Tables, or DHTs. It can be used as a distributed storage and lookup service. Like a conventional hast table, a DHT stores (key, value)-tuples, but does so in a highly scalable, decentralized and fault-tolerant manner [Pos03].
  In the approach taken by [PiJu06], the DHT stores information on managed objects (users, services, etc.) by applying the hash function on the name, obtaining the key, and storing this key together with the address. The DOMs form a ring together in which each DOM is responsible for a specific range of keys. A DOM maintains a table of neighbour DOMs. This finger table is used to determine the DOM to send a query to.

The number of DOMs in each table and number of steps to find the responsible DOM for a specific object is logarithmic with respect to the total number of nodes. Compared to flooding techniques, DHTs give strong guarantees on routing and message delivery.

The combination of peer-to-peer technology with network management seems promising. An area in which this combination could also proof very interesting due to several similarities is Wireless Mesh Networks. The following section will introduce Mobile Ad-hoc Networks in general and these Wireless Mesh Networks in particular. It will also explain the similarities between these networks and peer-to-peer technology.

3.4 Mobile Ad-hoc Networks

Ad-hoc mobile communication is characterised by its self organising connectivity with no pre-existing infrastructure [Pos03]. Mobile ad-hoc networks (or MANETs) are self configuring wireless networks where each terminal can send, receive and route information [ScGr02]. They hold similarities with peer-to-peer systems, since both offer networking functionality in a completely decentralised environment, as also indicated in [HuSa03]. This decentralised environment makes mobile ad-hoc networks an interesting research area to explore the possibilities of a decentralised management system based on peer-to-peer principles.

Stations forward data to each other. Even when two stations are not within direct transmission range, information can be sent from one station to another, using neighbouring nodes as relay stations [ScGr02]. The main advantage of this kind of network is that it lacks the need of an infrastructure. Because of this flexibility, ad-hoc technology enables network setup in locations where communication would otherwise be impossible, like after a disaster in which existing infrastructure is damaged [Pos03].

The lack of infrastructure also has its disadvantages. Since data is relayed by intermediate nodes, secure communication could be necessary. Because the lack of a central entity, authentication of communication partners is not easy or might even be impossible to guarantee. Besides this security issue, the mobility of nodes results in a dynamic topology. When links between nodes change, routing tables should be updated. Since each node can act as a router, these updates have to be announced to all of them to ensure a correct network view at all time in every node. This causes scalability problems for ad-hoc networks with a large number of nodes combined with high mobility patterns. Combined with the scarce available bandwidth, optimized routing algorithms are needed [ScGr02] [Pos03]. To reduce traffic and to ensure scalability, it is important that also the network management system makes effective use of the underlying routing protocol. Section 3.4.1 describes the basics of the different types of routing algorithms currently available.

3.4.1 Ad-hoc routing algorithms

A number of different routing algorithms have been developed for ad-hoc networks. They can be divided into a number of different categories which will be briefly described below.

Proactive (table driven) routing
When running a proactive routing algorithm, each node has complete knowledge of the network topology. Changes in this topology are broadcasted to all other nodes in the network. Because of this knowledge, routes can be established very fast. However, when the number of nodes grows, or mobility is high, the high frequency of topology updates makes this solution less scalable [ScGr02].

Examples are:

- Optimised Link State Routing (OLSR) [JaMu01]
- Destination-Sequenced Distance-Vector (DSDV) [PeBh94]

Reactive (on-demand) routing
A reactive routing protocol does not send topology updates. A route is determined ‘on-demand’. When a node needs to send information to another node, it will flood a route request through the network [ScGr02]. This makes this protocol suitable for small networks

with high mobility of nodes for example. In large networks, this solution is only suitable when few route requests are generated, since the route requests could result in flooding the complete network [Pos05].

Examples are:

- Dynamic Source Routing (DSR) [JoMa01]
- Ad-hoc On-Demand Distance Vector (AODV) [PeRo99]

Hybrid solutions
A combination between proactive and reactive routing algorithms is called a hybrid solution, which could yield better results than pure proactive or reactive algorithms. However, they still have scalability issues since they still rely on flooding and link updates, making them less suitable for very large networks [Pos05].

Examples are:

- Sharp Hybrid Adaptive Routing Protocol (SHARP) [RaHa03]
- Zone Routing Protocol (ZRP), where the scope of the proactive procedure is limited to the local neighbourhood and route discovery is done on-demand [KoVa00].

Hierarchical (cluster based) routing
Instead of routing between individual nodes, it can also be based on routing between clusters of nodes. When a node wants to communicate with a node outside of its cluster, it typically sends the data to its cluster head, which will forward it to the next level. This is repeated until a cluster head of the destination node is in the same cluster, and the data can travel down to it [Pos05][IbMa04]. A difference between this approach and the abovementioned Zone Routing Protocol is that with ZRP, a routing zone is defined for each node separately, and zones of neighbouring nodes overlap [HaPe02].

Examples are:

- Hierarchical State Routing (HSR) [PeGe99]
- Hierarchical OLSR (HOLSR) [GeLa05]

Location based routing
Another way of improving the routing performance is to include location information in the protocol. Some routing algorithms have been developed that use geographic coordinate information provided by GPS receivers. Data is subsequently forwarded to geographic closer nodes until it reaches its destination [Pos05][KoVa00].
Besides using GPS information to determine the location, also virtual coordinates can be used to guide the routing process. By using the distance to a set of well-known nodes in the network, the location can be approximated. This approach is based on the Landmark hierarchy for routing, where these well-known nodes are known as landmarks [ChMo02][Pos05].

Examples are:

- Location Aided Routing (LAR), a DSR based approach where route requests are constrained to a geographic area for efficiency, using GPS for coordinate information [KoVa00]. In the worst case, LAR still floods the network to discover the current location of a destination, reducing the scalability [Pos05].
- Distance Routing Effect Algorithm for Mobility (DREAM) works similarly to LAR, but location changes are proactively flooded throughout the network, again reducing the scalability [Pos05].
- Landmark based routing protocols as LANMAR [PeGe00b], Fisheye State Routing [PeGe00a], L+ [ChMo02] and Beacon Vector Routing (BVR) [FoRa04].

3.4.2 Wireless Mesh Networks

A Wireless Mesh Network (WMN) is a subtype of ad-hoc networks. A WMN consists of mesh routers and mesh clients. Besides the normal gateway functions, mesh routers can forward packets on behalf of other nodes, creating an ad-hoc network and providing connectivity for clients and other routers. When a router in the network is connected to the Internet, it can act as a gateway, providing Internet connectivity for all nodes [AkWa05]. Figure 8 below shows an example of a wireless mesh network with two nodes acting as such Internet gateways. Nodes can either be connected by their wireless or their wired interfaces.

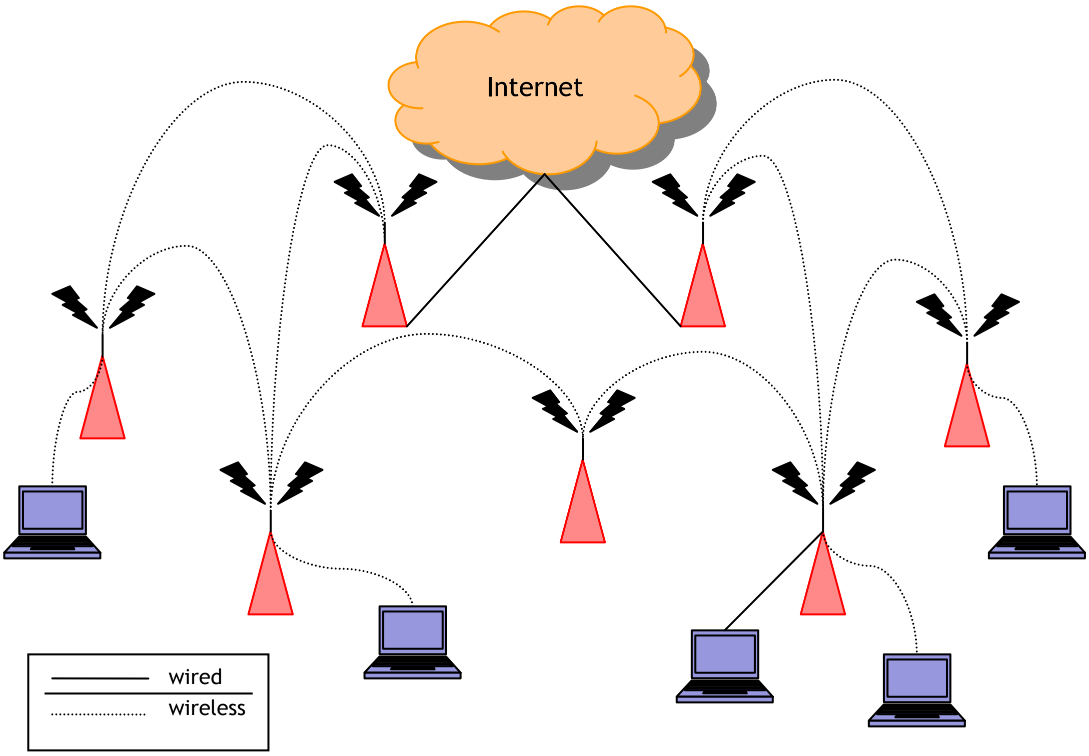


Figure 8: Example wireless mesh network

Since a wireless mesh network is a specific type of an ad-hoc network, an ad-hoc routing protocol is typically used. Mobility of mesh routers is very low, or even non-existing, which leads to few route updates. They form the backbone of the network. For inter-router communication, a reactive routing protocol would thus seem less appropriate, since topology changes are less frequent.

The similarities between peer-to-peer technology and mobile ad hoc networks have already been indicated in section 3.4. Both provide networking connectivity in a decentralized nature. For Wireless Mesh Networks, a specific type of mobile ad hoc network, the feasibility of a decentralised, peer-to-peer management system is explored in the Celtic Madeira project, which will be described in more detail in the following section.

4 Peer-to-Peer in Management for Wireless Mesh Networks

In the previous section, Mobile Ad-hoc Networks and Wireless Mesh Networks are introduced, and their similarities with peer-to-peer technology. This section focuses on the work performed in the Celtic Madeira project [Mad06], which aims at the development of a peer-to-peer based management solution for such Wireless Mesh Networks. Section 4.1 contains an overview of the general principles in this approach, derived from [ArFr06]. For this management solution, a Web Services based northbound interface is developed. A description of this Madeira Northbound Interface can be found in section 4.2, followed by a description of the Operations Support System in section 4.3. Finally, the advantages and disadvantages of the Madeira approach will be given.

4.1 Introduction

In an attempt to overcome the shortcomings of the traditional management approaches to face the challenges of next generation telecommunication networks, Madeira aims to develop a new management framework based on peer-to-peer networking concepts, designed for Wireless Mesh Networks. Furthermore, it provides technologies for a logically meshed Network Management System to facilitate self-management and dynamic behaviour of nodes within the network. Madeira is designed to be able to use any underlying routing protocol to create a flexible solution. It also uses Policy Based Management Paradigm [RoHo03] that pursues the separation of management logic from the actual applications. This logic is then specified as a set of rules or policies that can be dynamically fed into the management system, allowing a change of its behaviour without the need of changing the application or even restarting it. Besides this architectural framework, the Madeira project provides interface protocols, standards and a reference software implementation. Ultimately, by enabling the management of network elements of increasing numbers, heterogeneity and transience, the Madeira approach should reduce the Operating Expenses, or OPEX.
It investigates the feasibility of distributing management responsibilities among peer nodes, and focuses on Fault and Configuration Management functional areas and, especially, on the way they can co-operate to solve management problems

4.1.1 Overlay Management Network

The approach of Madeira is completely different than in traditional management systems. It encompasses a much flatter structure that is based on peer-to-peer (P2P) principles. The management functions are executed in P2P aware Adaptive Management Components (AMCs), which correspond directly to the Network Elements (NEs). By using a well-defined peer-to-peer interface, these AMCs can communicate with each other, creating an Overlay Management Network. Figure 9 gives a graphical representation of this overlay.

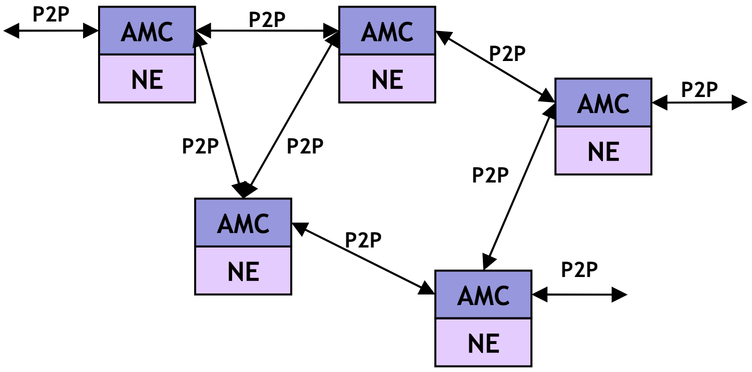


Figure 9: Overlay Management Network using AMCs

As mentioned before, an Adaptive Management Component (AMC) fulfils the network management functionality of a peer in a P2P network. It can exchange management information between peer management applications in a network element or management node. One or more AMCs interact with each other to perform a specific network management application. Moreover, an AMC has the ability to import (and export) functionality to perform specific tasks.

4.1.2 Madeira components

In order to perform the management tasks, each AMC requires a variety of services. These are provided by the Madeira platform. The separation of the Madeira Management System between AMC and platform is depicted in Figure 10. The AMC covers the management specific parts for a particular scenario, while the platform provides all the generic functionality required to run tasks in a P2P environment. Separating this functionality enables Madeira to adapt to changing scenarios and requirements in an efficient way.

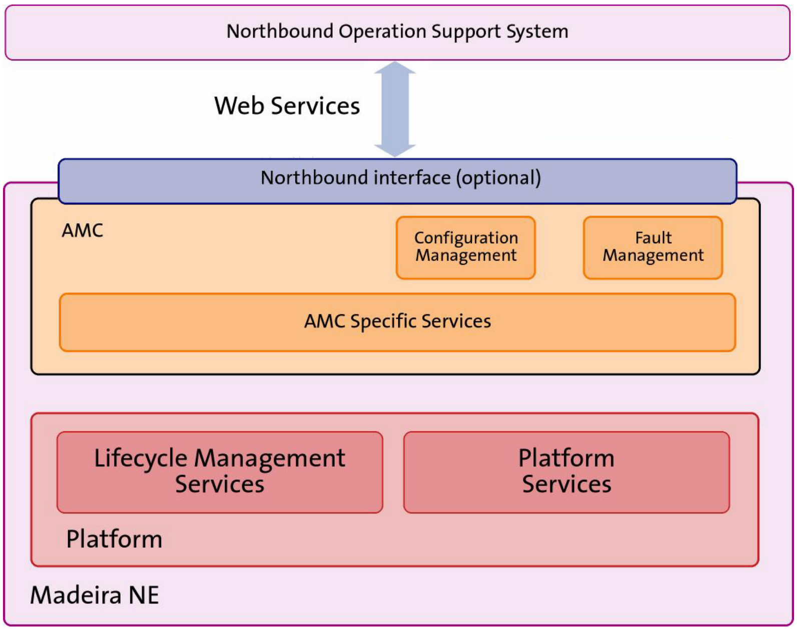


Figure 10: Madeira components

Figure 10 above shows the internal Madeira components, and their relationship with the Northbound Interface. The following groups of services are available in an AMC:

- The Northbound Interface is an optional interface that will be described in more detail in section 4.2. The Northbound Interface enables a higher layer, external Operations Support System (OSS) to communicate with Madeira via Web Services. The OSS can, for example, retrieve information like network topology, events or alarms. The design of this interface is part of the scope this thesis.
- The Configuration Management and Fault Management layer contains the specific network management applications. They provide the ability to set-up the network, react to faults and other CM and FM related tasks.
- The AMC Specific Services offer a base for the Network Management Applications. It mainly provides services to communicate with other AMCs. This can be either publish-subscribe based, or a direct peer-to-peer connection to another AMC.

As mentioned before, the AMC performs all tasks that are directly related to network management. To execute these tasks, the AMC needs services that are offered by the platform. The platform takes care of all additional functionality that is needed in a peer-to-peer environment. Its capabilities are divided into two groups of services:

1. The Lifecycle Management Services take care of the management of AMC containers. In more detail, this group offers the following services:

- The Lifecycle Service can perform start/stop/restart operations on all modules loaded by the AMC.
- The Code Distribution Service enables dynamic loading of application logic/data into the AMC. AMCs have a minimum "bootstrap" configuration to function in the network. Additional functionality can be imported as required with this service.
- The Security Service provides all aspects of security and authentication from a platform perspective.

2. The Platform Services offer the following services, which are specific for the peer-to-peer environment:
    - The Notification Service is a basic event notification service, based on a standard publish-subscribe service. It enables AMCs to subscribe to certain event types.
    - The Directory Service is a directory of AMCs with their roles and capabilities. It keeps track of the physical one-hop neighbourhood of the NE and enables AMCs to be looked up.
    - The Connectivity Service provides reliable one to one communication between two AMCs. This point to point connection supports multi-hop P2P communication.
    - With the Persistency Service, AMC data can be stored for retrieval across restarts. It supports permanent (local) storage of application-defined data.
    - The Grouping Service can dynamically form AMC groups for a given management function. It provides application partitioning for AMCs of similar roles or capabilities.

4.1.3 Clustering

The Grouping Service provided by the Platform Services offers the ability to create groups of AMC that perform a specific management function. These groups are called management clusters. Each management cluster contains exactly one network node that acts as the cluster head. This cluster head is responsible for coordination of and topology publishing for its cluster. The clustering principle makes Madeira a hybrid peer-to-peer solution, where the cluster heads can be seen as "super peers". The other nodes are "normal peers", as described in section 3.2. An example of the clustering of nodes is given in Figure 11, in which cluster heads are depicted as black nodes.

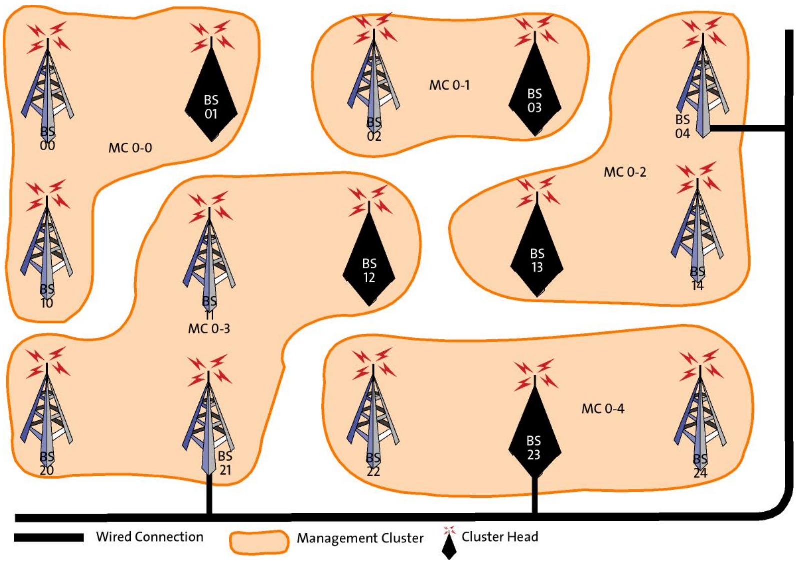


Figure 11: Management Clusters in the Wireless Meshed Network

Different levels of clustering can exist, which leads to a clustering hierarchy. A node in a level *n* cluster is the cluster head in a lower level *n-1* cluster. Such a clustering hierarchy is shown in Figure 12, and corresponds to the example given in Figure 11 above.

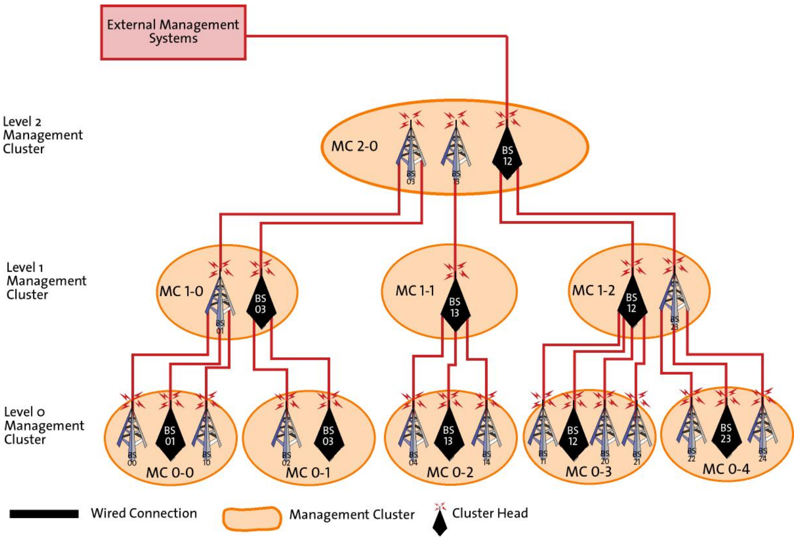


Figure 12: Management Cluster hierarchy

A level “n” cluster head receives information from his cluster members and, subsequently, receives information from the lower “n-1” clusters since each member is a cluster Heads of a level “n-1” cluster. The top level cluster head is responsible for the complete network. This makes this node ideal for publishing, for example, the topology of the complete network and events or alarms to a higher layer Operations Support System through the Northbound Interface. However, since all nodes can exchange all information with each other, this is not obligatory.

The use of this clustering concept is of great importance for the scalability of the management application. It creates the ability to divide a large network into smaller, dynamic groups to perform certain management tasks.

4.1.4 Policies

In Madeira, policies are used to tune the behaviour of the management application. These policies are introduced to the system by the Northbound Interface and automatically distributed among the appropriate AMCs. Certain policies can be present during initialisation. To give an example of this rather abstract principle, the structure of the management clusters mentioned above is based on policies. They can, for example, control the number of nodes per cluster, but they can also contain criteria for the election of cluster head, such as memory resources, load etc.

Policies are also used in the fault reporting function of Madeira. Alarms can travel up the hierarchy. On each layer, alarm information is correlated and a decision is made, based on information in the policies, if the alarm is important enough to be forwarded to a higher layer. This process is repeated until the top of the hierarchy is reached. In other words, it is not compulsory that every alarm that occurs in a low level will appear in the top level. Only alarms that appear in the top level will be forwarded to an external Operations Support System.

4.1.5 Classification

Madeira creates a clustering hierarchy as explained above. In short, a cluster head is responsible for its cluster, and receives, for example, events and alarms from the members. This results in the fact that the top level cluster head is responsible for the complete network. Alarms and events are analysed and correlated at each level, after which they can be forwarded to the next level. This concept of different layers, and the absence of simple, pure agents, eliminated the centralized approach given in section 2.1, since that approach uses a single manager that polls a large number of simple agents. In the cooperative management approach described in section 2.4, a hierarchy is typically not present. There is no top level manager that is responsible for the whole network. Given this fact, the Madeira approach cannot be seen as a form of cooperative management.

In other words, the Madeira peer-to-peer approach would belong to one of the two distributed approaches. The clear distinction between the weakly distributed and strongly distributed approach is, like many other categorizations, difficult to make. However, in this thesis, the Madeira approach is considered as a strongly distributed management approach:

- There are no nodes functioning as simple data collectors. Nodes in Madeira are not polled by a manager as is the case with pure agents. All nodes are basically equal and cooperate in clusters to fulfil management tasks.
- When connection with a cluster head is lost, a Madeira node will keep functioning. The management application will try to fulfil its task, and will trigger re-clustering. It could even become a cluster head itself. The same principle applies when the top level cluster head fails. Automatically, a new clustering hierarchy well be set-up, ensuring a fully functional management application again.
- The use of policies approaches the goal-oriented nature described in the cooperative approach. Thus, this property is more closely related to the strongly distributed than to the weakly distributed approach.

4.2 Northbound Interface

The Northbound Interface offers support for a higher layer Operations Support System (OSS) or another Network Management System to access Madeira's network management functionality. From this point on, the Operations Support System as well as the Network Management System shall be referred to as OSS.

To ensure an open environment and to enable cross-platform communication and interoperability, the communication between Madeira and an external OSS is based on Web Services. According to W3C, a Web Service is "*a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.*" [W3C].

To maintain the readability of this document, a more detailed explanation on Web Services can be found in Appendix A – Relevant Web Services standards.

4.2.1 Design

The design of the Northbound Interface is based on the functional and non-functional requirements given in Appendix B – NBI Requirements. These requirements are developed from higher level Madeira requirements, and are included as background information for the design choices made below.

The external interface of the Northbound Interface is based on Web Services, as mentioned earlier. This interface is described in WSDL (Web Services Description Language). The WSDL specification is an XML description of the public interface to the Web Service. It contains information like protocol bindings, message formats and available operations and messages.

Request/Response communication

The Northbound Interface supports 'normal' request/response type of communication, which is initialised by the OSS. This kind of communication is intended to enable the OSS to perform simple management tasks as introducing (updated) policies or disabling base stations, and to acquire management information on, for example, the cluster topology.

The communication is based on XML SOAP messages sent by the OSS to the NBI. All available methods are described in the WSDL file. This file also contains information on the data that is contained in both request and response message. In other words, XML style sheets have been developed and included in the WSDL file to describe the management clusters, policy files and physical topology information for example.

Notifications

Besides communication initialised by the OSS, the Northbound Interface also has to support sending notifications in case of an event or alarm. The OASIS Web Services Notifications (WSN) standard provides this functionality. This standard is also included in the OASIS Web Services Distributed Management (WSDM) standard. As can be read in Appendix A – Relevant Web Services standards, WSDM is developed to manage entities in a network via Web Services. However, due to a combination of factors, it was chosen to use only the OASIS Web Services Notifications standard:

- The novelty of the Madeira solution in general, and the prototype in particular, required a large degree of flexibility on the information exchanged between the Northbound Interface and the OSS;
- The available software supporting WSDM was inferior to the software supporting WSN.

The topics an OSS can subscribe to are directly linked to the available information in the Madeira management system. Notifications from both Configuration Management and Fault Management applications are received by the Northbound Interface, converted into Web Services Notifications messages, and distributed to all subscribed consumers.

Configuration Management event notifications are Madeira specific. They concern the creation and deletion of management clusters, as well as the addition and removal of members in these clusters for example. The structure of the notifications sent to the OSS is thus specifically designed to include this information.
For Fault Management notifications, the ITU X.733 Alarm Report Function [ITU92] standard has been used. In other words, the fields in the Fault Management alarm notification that is sent from the NBI to an OSS are compliant to the ITU X.733 standard.

Discovery of NBI node

The network node that is running the Northbound Interface functionality can dynamically change as a result of network reconfiguration. Due to this, some functionality should be in place so the OSS can discover the new address. The following two approaches are available:

1. **UDDI approach**
   Madeira will use an external UDDI (Universal Description, Discovery and Integration) registry to store the NBI address. It is an XML registry in which information on Web Services can be stored [UDD06]. The address of the external UDDI repository has to be included in Madeira, possibly by some manual preconfiguration. In the UDDI, the NBI will publish an "Organization" with a specific name. This organization can have multiple "Services". After creating a service with a specific name, a "Service Binding" is created which contains the address of where the NBI can be reached. For an external entity to discover this address, the organization and service names have to be known. Figure 13 shows the relationship between Northbound Interface, OSS and UDDI.

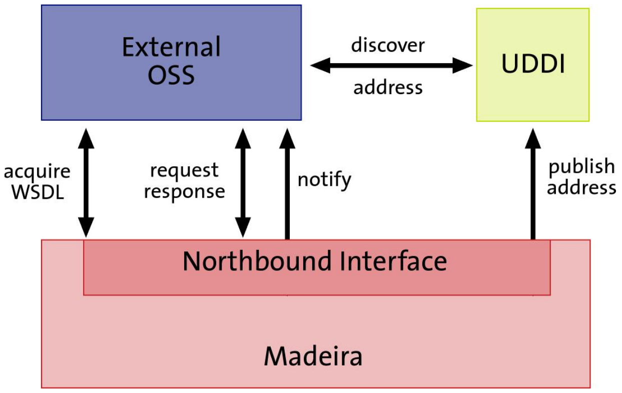


Figure 13: Usage of Web Services for the Northbound Interface

2. **Peer-to-Peer approach**

Because the use of the UDDI registry is not a peer-to-peer solution, another solution is proposed. With this solution, the OSS can contact any node in Madeira with a Web Services SOAP request. The receiving node will forward this request to its cluster head by using Port Forwarding. This process will be repeated until it reaches the top level cluster head, which will reply with a SOAP message containing its address. This response will travel down the clustering hierarchy until the original node sends it to the OSS. Subsequent requests from the OSS will be done directly to the NBI node. In this approach, the OSS has to be aware of the address of at least one node in the network. Because of time constraints, this approach is not further developed.

The OSS has to be able to detect a change of the NBI node. When such a change is detected, the OSS can contact the UDDI registry in order to acquire the new address. For this detection, a polling mechanism can be used. When a timeout occurs, the detection process of the new NBI node is restarted. This can be either by using the UDDI registry, or by using the peer-to-peer solution.

Internal structure

The Northbound Interface propagates instructions from the OSS to the platform, and data from the platform back to the OSS. The internal structure is depicted in Figure 14.

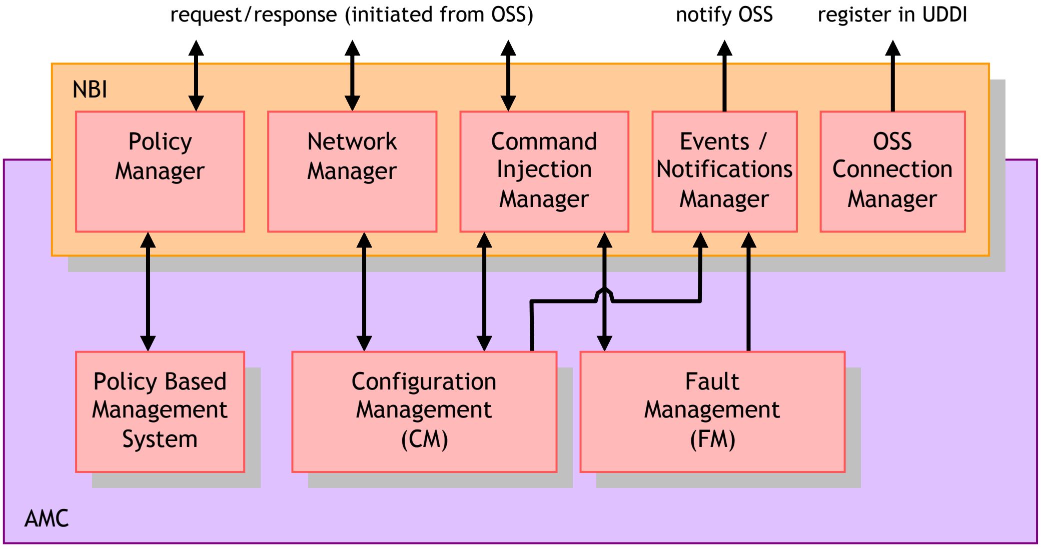

request/response (initiated from OSS)
notify OSS
register in UDDI
NBI
Policy Manager
Network Manager
Command Injection Manager
Events / Notifications Manager
OSS Connection Manager
Policy Based Management System
Configuration Management (CM)
Fault Management (FM)
AMC

Figure 14: NBI Architecture

As is shown in this picture, the Northbound Interface offers the following services, divided into five groups:

1. The Policy Manager can be used to add, change or remove policies to the Madeira Management System.
2. The Network Manager provides information about the network topology and management clusters.
3. The Command Injection Manager receives commands from the external OSS, forwards them to the appropriate Madeira components for further processing and sends back the corresponding response to the OSS. The OSS can use the Command Injection Manager to disable Base Stations or to acquire a list of currently active alarms for example.
4. The Events/Notifications Manager receives alarms and events from Madeira, and converts these messages to a format readable by the OSS. Notifications will be sent to every subscribed consumer. This means that multiple Operations Support Systems can monitor the Madeira Management System.
5. The OSS Connection Manager enables the OSS to (re)discover the Northbound Interface node. It publishes for example the address of the Northbound Interface in

an external UDDI registry. It also responds to polling messages sent by the OSS to check connectivity with Madeira.

4.2.2 Implementation

The prototype of the Madeira management system is developed and implemented by the different members of the Madeira consortium. Since the chosen programming language is Java, the Northbound Interface was also to be implemented in this language. Section "Implementation" in Appendix A – Relevant Web Services standards contains information on the available software that offers support for the necessary Web Services standards. As stated there, support for Web Services in general, and the OASIS Web Services Notifications standard in particular, is given by Apache Pubscribe. This web application needs a web container or web application server to run. The recommended solution, Apache Tomcat, was well documented and proved to work correctly. Thus, for the implementation of the Northbound Interface, the following solution is chosen:

- Apache Tomcat v5.5, web container
- Apache Pubscribe v1.1, web application with support for Web Services including Web Services Notifications

To ensure an efficient implementation of the Northbound Interface within the Madeira prototype, both had to run in the same Java Virtual Machine. Thus, instead of starting Apache Tomcat as a standalone application, it had to be started during runtime by the Madeira software, which proved to be a challenge but was accomplished by David Ortega, Madeira project participant and colleague at Telefónica.

The basis of the Northbound Interface is its WSDL file. It specifies all the available methods, including message formats and protocols. It also contains the necessary information on the notification topics, as well as the structure of the notification messages. This file is then automatically converted into Java class files. After implementing the functionality in these files, the application was deployed as a Web Service.

4.3 Operations Support System

The Northbound Interface only responds on Web Services requests. It does not provide a graphical interface. In order to show the capabilities of the Madeira management system, and to create a comprehensible testing environment, a Madeira-specific Operations Support System (or OSS) is created. This system can issue Web Services requests to the Madeira node that runs the Northbound Interface. This NBI node is usually the top level cluster head.

4.3.1 Design

As mentioned in section 4.2, the NBI uses 'normal' Web Services request/response communication to reply on requests from an OSS. The OSS then acts as a client, while the NBI acts as server. The NBI also offers also a publish/subscribe mechanism which enables an external OSS to subscribe for certain topics and receive notifications, compliant to the OASIS Web Services Notifications standard [WSN06]. To be able to receive notifications, the OSS has to listen on a predefined address. Although it is not necessary to implement a Web Service to receive response, this is the preferred approach. Apache Pubscribe [Pub06] offers such a service when it is run in 'consumer mode'.

Because of these different communication mechanisms, the Madeira OSS consists roughly of three different components. Each of these components will be described in more detail below. Figure 15 gives a graphical representation on the communication structure between NBI and OSS.

1. Apache Pubscribe in consumer mode
2. MySQL database
3. JSP pages

Apache Pubscribe in consumer mode

Apache Pubscribe is a web application. This web application has to be executed in a web container of application server. It is recommended to use Apache Tomcat as its web container.

Apache Pubscribe enables the OSS to be a Web Service that can receive and handle notifications. For this, Pubscribe has to be in ‘consumer mode’, implementing the “Notify” method. This is the method that is invoked by the Web Services request sent by the Madeira NBI.

The basis of the Madeira OSS Web Service is the WSDL (Web Service Description Language) file. This file describes the interface and contains for example the available methods, data types, and supported transfer mode.

Upon receiving a notification, the OSS will analyse the contents. After determining if it is an FM alarm or a CM event, the OSS checks the MySQL database if the alarm or event already exists. It will then either update the information or insert a new entry.

MySQL database

The MySQL database contains the notification information. It has one table for FM alarms, and one table for CM events. The database is filled by the OSS Web Service that runs in Pubscribe.

The JSP pages also access the database. They can acquire the information and delete notifications.

JSP pages

The JSP pages offer the user access to Madeira. They provide a graphical user interface that can be reached by an Internet browser using HTML over HTTP. The JSP pages provide an easy access to the services offered by the Madeira NBI. For the user, they are the starting point for all the request/response based methods. Besides, they depict the notification information stored in the MySQL database.

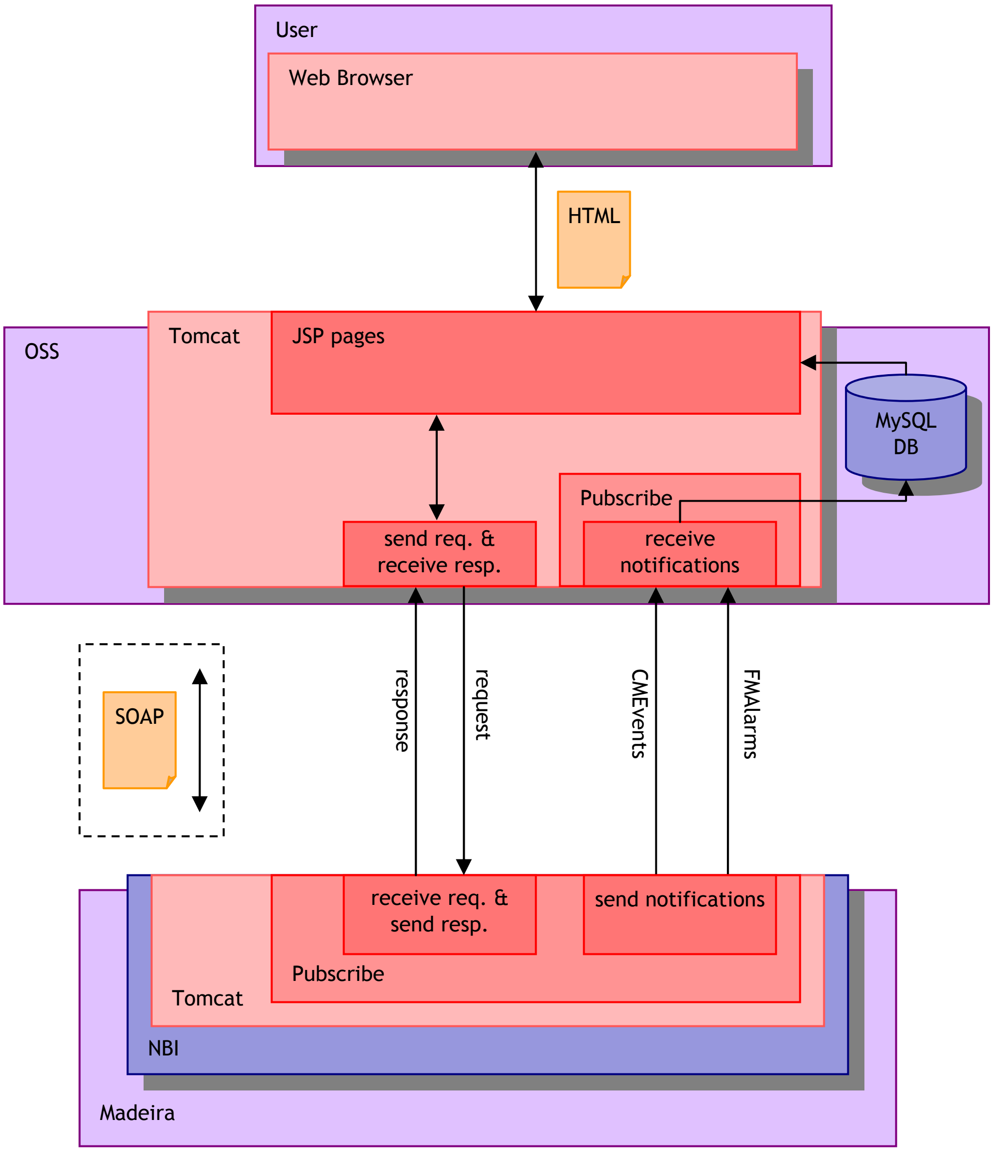


Figure 15: Communication structure between NBI and OSS

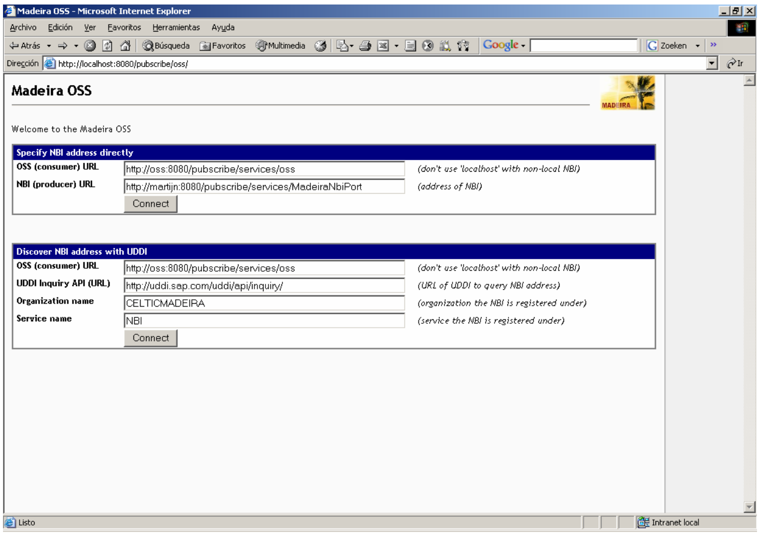
4.3.2 Using the OSS

Initial screen

The OSS can be reached using a web browser. When opening the initial page, shown in Figure 16, the OSS offers two ways to connect to the Northbound Interface of the Madeira management network:

1. Entering the address of the NBI directly. This address is the Web Service Endpoint and typically has the following value: “*http://<IPADDRESS>:8080/pubscribe/services/MadeiraNbiPort*"
2. Using the UDDI. When using this method, additional information is required like the organization and service names under which the NBI address is registered.

For both methods, the address of the OSS should be entered. This address is used to indicate the NBI where notifications should be sent to.


Madeira OSS - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia Google Zoeken
Dirección
MADEIRA
Listo
Intranet local

http://localhost:8080/pubscribe/oss/

Madeira OSS

Welcome to the Madeira OSS

Specify NBI address directly

OSS (consumer) URL: http://oss:8080/pubscribe/services/oss (don't use 'localhost' with non-local NBI)

NBI (producer) URL: http://martijn:8080/pubscribe/services/MadeiraNbiPort (address of NBI)

Connect

Discover NBI address with UDDI

OSS (consumer) URL: http://oss:8080/pubscribe/services/oss (don't use 'localhost' with non-local NBI)

UDDI Inquiry API (URL): http://uddi.sap.com/uddi/api/inquiry/ (URL of UDDI to query NBI address)

Organization name: CELTICMADEIRA (organization the NBI is registered under)

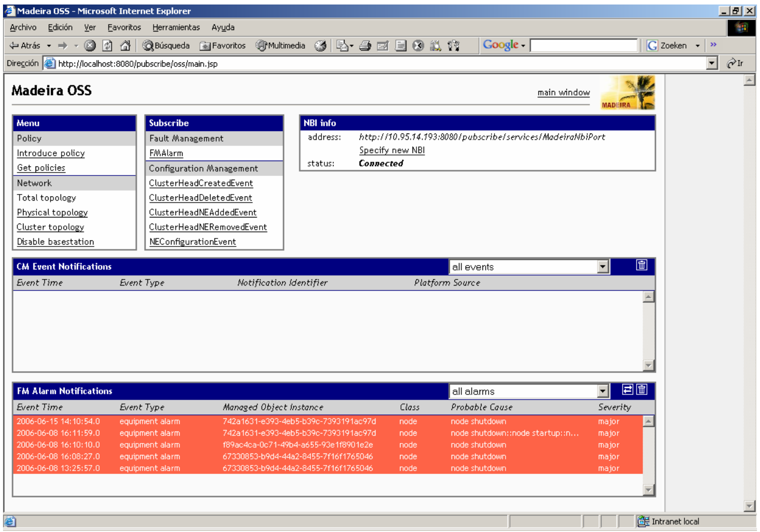
Service name: NBI (service the NBI is registered under)

Connect

Figure 16: Madeira OSS – Initial screen

Main window

After clicking connect, the OSS will try to connect to the NBI, possibly by contacting the UDDI first. When the connection was successful, information on subscribed topics is acquired from the NBI, after which the user can go to the main window, shown in Figure 17.



http://localhost:8080/pubscribe/oss/main.jsp

Madeira OSS

NBI info

address: *http://10.95.14.193:8080/pubscribe/services/MadeiraNbiPort*

Specify new NBI

status: ***Connected***

CM Event Notifications all events

Event Time	*Event Type*	*Notification Identifier*	*Platform Source*

FM Alarm Notifications all alarms

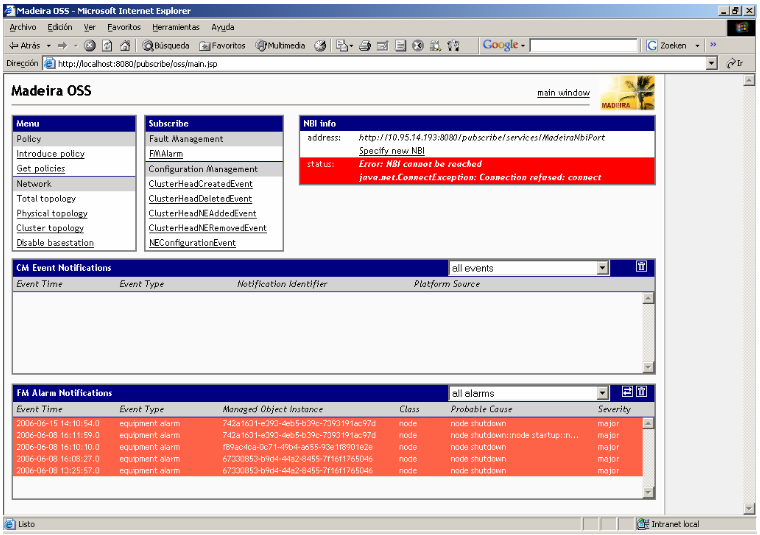
Event Time	*Event Type*	*Managed Object Instance*	*Class*	*Probable Cause*	*Severity*
2006-06-15 14:10:54.0	equipment alarm	742a1631-e393-4eb5-b39c-7393191ac97d	node	node shutdown	major
2006-06-08 16:11:59.0	equipment alarm	742a1631-e393-4eb5-b39c-7393191ac97d	node	node shutdown::node startup::n...	major
2006-06-08 16:10:10.0	equipment alarm	f89ac4ca-0c71-49b4-a655-93e1f8901e2e	node	node shutdown	major
2006-06-08 16:08:27.0	equipment alarm	67330853-b9d4-44a2-8455-7f16f1765046	node	node shutdown	major
2006-06-08 13:25:57.0	equipment alarm	67330853-b9d4-44a2-8455-7f16f1765046	node	node shutdown	major

Figure 17: Madeira OSS – Main window

The main window shows the following information:

- Available methods
  The user can acquire and send policies, get the physical and cluster topology and disable base stations.
- Available topics to subscribe to for notifications
  The user can subscribe for the FM alarms and CM events. When the OSS is subscribed for a certain topic, this is marked with a checkmark (✓ -icon) in the main window.
- Connection status
  It shows information on the connection status. If the NBI does not respond, it is depicted here. Information on possible UDDI failures is also shown. The OSS calls the KeepAlive method every 30 seconds. When connection is lost, it tries to reconnect every 10 seconds. When UDDI is used, it consults the registry every 10 seconds before polling the NBI. It is also possible to manually select a new NBI address. An example screen on connection loss is shown in Figure 18.

- CM events
  Information on the CM events present in the database. All events can be removed by clicking on the 🗑-icon. Clicking on an event will give detailed information on it.
- FM alarms
  Information on the FM alarms present in the database. All alarms can be removed by clicking on the 🗑-icon. Clicking on an alarm will give detailed information on it. It is also possible to synchronize the alarms by acquiring the active alarms. To do this, click on the ⇄-icon.



address: http://10.95.14.193:8080/pubscribe/services/MadeiraNbiPort

status: Error: NBI cannot be reached
java.net.ConnectException: Connection refused: connect

Event Time	Event Type	Notification Identifier	Platform Source

Event Time	Event Type	Managed Object Instance	Class	Probable Cause	Severity
2006-06-15 14:10:54.0	equipment alarm	742a1631-e393-4eb5-b39c-7393191ac97d	node	node shutdown	major
2006-06-08 16:11:59.0	equipment alarm	742a1631-e393-4eb5-b39c-7393191ac97d	node	node shutdown::node startup::n...	major
2006-06-08 16:10:10.0	equipment alarm	f89ac4ca-0c71-49b4-a655-93e1f8901e2e	node	node shutdown	major
2006-06-08 16:08:27.0	equipment alarm	67330853-b9d4-44a2-8455-7f16f1765046	node	node shutdown	major
2006-06-08 13:25:57.0	equipment alarm	67330853-b9d4-44a2-8455-7f16f1765046	node	node shutdown	major

Figure 18: Madeira OSS – Loss of connection with NBI

Policies

The OSS offers the ability to influence the behaviour of Madeira by altering the policies.

- Introduce policy
  The introduce policy window enabled a user to dispatch policies to the Madeira management system. Removing a certain policy is done by submitting an inactive policy with the same identifier.
- Get policies
  The get policies window will show all the available policies in the Madeira management system.

Topology

Madeira offers two different kinds of topology. The cluster topology depicts the clustering hierarchy while the physical topology gives information on interfaces, routes, neighbours and connected wireless equipment.

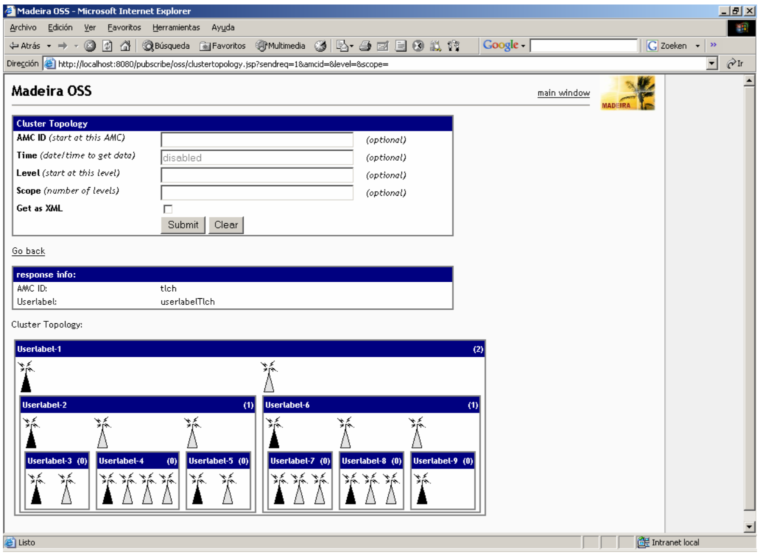
Cluster Topology

The cluster topology window gives you a number of fields you can fill in. All of them are optional. If all fields are left blank, this results in acquiring the total cluster topology by starting from the top of the clustering hierarchy.

- The AMC ID field specifies the AMC ID (i.e. the node) from which the topology information should start.
- The Time field is disabled in this prototype of Madeira
- Level indicates at which level the topology information should start.
- The Scope indicates how many extra levels should be included. When the scope is 0, only one level will be sent.
- Get as XML indicates if the response should be represented as an XML file.

The clustering hierarchy will be depicted as a figure. It will show the number of nodes per cluster, and the name of the cluster. Clicking on a certain node will give you a number of options:

- get the physical topology. Clicking on this option on a certain node will get the physical topology for all nodes under the selected node in the clustering hierarchy.
- disable base station. Clicking this option will present a screen to disable the selected base station.
- acquire 2 extra layers of cluster topology (only available on cluster heads). When you click on a cluster head, you can select this option to acquire 2 extra layers of clustering information, starting at that point in the clustering hierarchy. This enables users to 'walk' through the cluster topology.


Madeira OSS - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia Google Zoeken
Dirección
Ir
main window
MADEIRA
Submit Clear
Go back
Listo
Intranet local

http://localhost:8080/pubscribe/oss/clustertopology.jsp?sendreq=1&amcid=&level=&scope=

Madeira OSS

Cluster Topology		
AMC ID *(start at this AMC)*		*(optional)*
Time *(date/time to get data)*	disabled	*(optional)*
Level *(start at this level)*		*(optional)*
Scope *(number of levels)*		*(optional)*
Get as XML	□	

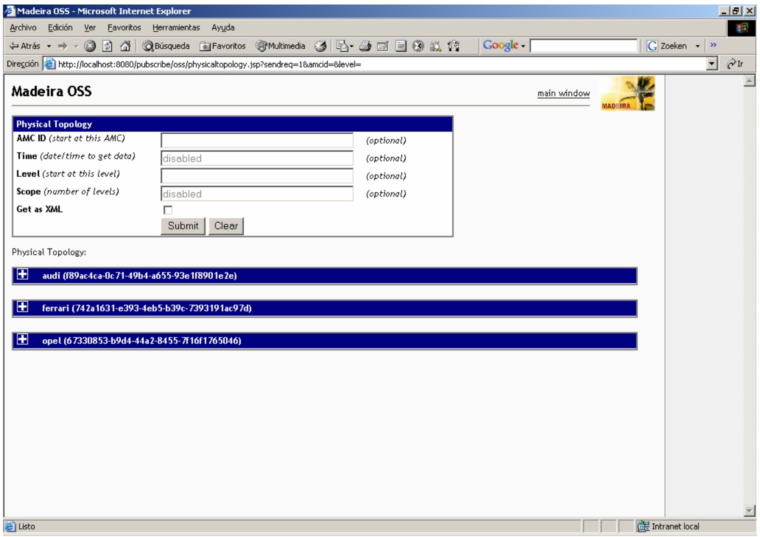
response info:	
AMC ID:	tlch
Userlabel:	userlabelTlch

Cluster Topology:

Userlabel-1 (2)
Userlabel-2 (1)
Userlabel-6 (1)
Userlabel-3 (0)
Userlabel-4 (0)
Userlabel-5 (0)
Userlabel-7 (0)
Userlabel-8 (0)
Userlabel-9 (0)

Figure 19: Madeira OSS – Cluster Topology

Physical Topology
The physical topology window gives you a number of fields you can fill in. All of them are optional. If you do not fill in any field, you will receive the total physical topology, starting from the top of the clustering hierarchy. The meaning of the fields is identical to the Cluster Topology.


Madeira OSS - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia Google Zoeken
Dirección
Ir
main window
MADEIRA
Submit Clear
Listo
Intranet local

http://localhost:8080/pubscribe/oss/physicaltopology.jsp?sendreq=1&amcid=&level=

Madeira OSS

Physical Topology		
AMC ID *(start at this AMC)*		*(optional)*
Time *(date/time to get data)*	disabled	*(optional)*
Level *(start at this level)*		*(optional)*
Scope *(number of levels)*	disabled	*(optional)*
Get as XML	□	

Physical Topology:

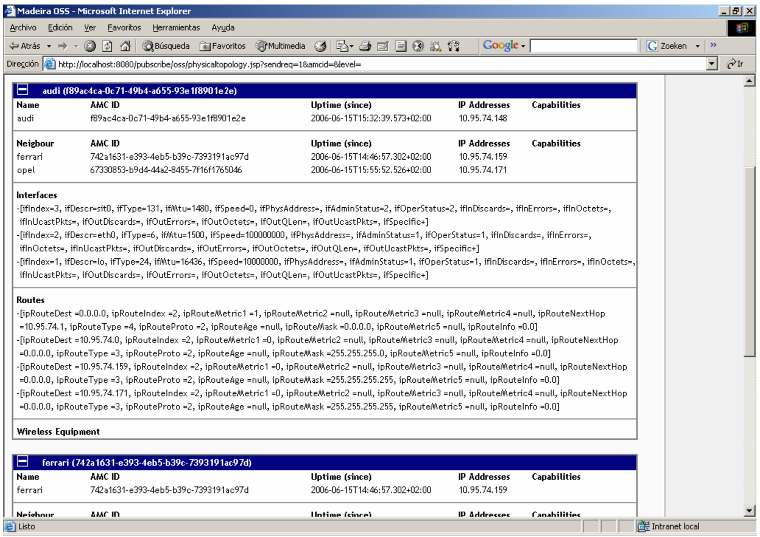
⊞ audi (f89ac4ca-0c71-49b4-a655-93e1f8901e2e)

⊞ ferrari (742a1631-e393-4eb5-b39c-7393191ac97d)

⊞ opel (67330853-b9d4-44a2-8455-7f16f1765046)

Figure 20: Madeira OSS – Physical Topology (collapsed)

The physical topology will be shown as a collection of tables as depicted in Figure 20. Each table represents a single node, and is 'collapsed'. Clicking on the ⊞-icon will expand the table and show all the information on that node. For each node, information on uptime, addresses and neighbours is shown, as well as information on the interfaces, routes and connected wireless equipment. Clicking on the ⊟-icon will collapse the table again. The expanded view is shown in Figure 21.


Madeira OSS - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia
Google Zoeken
Dirección http://localhost:8080/pubscribe/oss/physicaltopology.jsp?sendreq=1&amcid=&level=
Ir
Listo
Intranet local

audi (f89ac4ca-0c71-49b4-a655-93e1f8901e2e)

Name	AMC ID	Uptime (since)	IP Addresses	Capabilities
audi	f89ac4ca-0c71-49b4-a655-93e1f8901e2e	2006-06-15T15:32:39.573+02:00	10.95.74.148	

Neigbour	AMC ID	Uptime (since)	IP Addresses	Capabilities
ferrari	742a1631-e393-4eb5-b39c-7393191ac97d	2006-06-15T14:46:57.302+02:00	10.95.74.159	
opel	67330853-b9d4-44a2-8455-7f16f1765046	2006-06-15T15:55:52.526+02:00	10.95.74.171	

```
Interfaces
-[ifIndex=3, ifDescr=sit0, ifType=131, ifMtu=1480, ifSpeed=0, ifPhysAddress=, ifAdminStatus=2, ifOperStatus=2, ifInDiscards=, ifInErrors=, ifInOctets=,
 ifInUcastPkts=, ifOutDiscards=, ifOutErrors=, ifOutOctets=, ifOutQLen=, ifOutUcastPkts=, ifSpecific+]
-[ifIndex=2, ifDescr=eth0, ifType=6, ifMtu=1500, ifSpeed=100000000, ifPhysAddress=, ifAdminStatus=1, ifOperStatus=1, ifInDiscards=, ifInErrors=,
 ifInOctets=, ifInUcastPkts=, ifOutDiscards=, ifOutErrors=, ifOutOctets=, ifOutQLen=, ifOutUcastPkts=, ifSpecific+]
-[ifIndex=1, ifDescr=lo, ifType=24, ifMtu=16436, ifSpeed=10000000, ifPhysAddress=, ifAdminStatus=1, ifOperStatus=1, ifInDiscards=, ifInErrors=, ifInOctets=,
 ifInUcastPkts=, ifOutDiscards=, ifOutErrors=, ifOutOctets=, ifOutQLen=, ifOutUcastPkts=, ifSpecific+]

Routes
-[ipRouteDest =0.0.0.0, ipRouteIndex =2, ipRouteMetric1 =1, ipRouteMetric2 =null, ipRouteMetric3 =null, ipRouteMetric4 =null, ipRouteNextHop
 =10.95.74.1, ipRouteType =4, ipRouteProto =2, ipRouteAge =null, ipRouteMask =0.0.0.0, ipRouteMetric5 =null, ipRouteInfo =0.0]
-[ipRouteDest =10.95.74.0, ipRouteIndex =2, ipRouteMetric1 =0, ipRouteMetric2 =null, ipRouteMetric3 =null, ipRouteMetric4 =null, ipRouteNextHop
 =0.0.0.0, ipRouteType =3, ipRouteProto =2, ipRouteAge =null, ipRouteMask =255.255.255.0, ipRouteMetric5 =null, ipRouteInfo =0.0]
-[ipRouteDest =10.95.74.159, ipRouteIndex =2, ipRouteMetric1 =0, ipRouteMetric2 =null, ipRouteMetric3 =null, ipRouteMetric4 =null, ipRouteNextHop
 =0.0.0.0, ipRouteType =3, ipRouteProto =2, ipRouteAge =null, ipRouteMask =255.255.255.255, ipRouteMetric5 =null, ipRouteInfo =0.0]
-[ipRouteDest =10.95.74.171, ipRouteIndex =2, ipRouteMetric1 =0, ipRouteMetric2 =null, ipRouteMetric3 =null, ipRouteMetric4 =null, ipRouteNextHop
 =0.0.0.0, ipRouteType =3, ipRouteProto =2, ipRouteAge =null, ipRouteMask =255.255.255.255, ipRouteMetric5 =null, ipRouteInfo =0.0]

Wireless Equipment
```

ferrari (742a1631-e393-4eb5-b39c-7393191ac97d)

Name	AMC ID	Uptime (since)	IP Addresses	Capabilities
ferrari	742a1631-e393-4eb5-b39c-7393191ac97d	2006-06-15T14:46:57.302+02:00	10.95.74.159	
Neighbour	AMC ID	Uptime (since)	IP Addresses	Capabilities

Figure 21: Madeira OSS – Physical Topology (expanded)

4.4 Advantages

In the process of cooperating in the overall design, implementation and testing of the Madeira management solution in general and the Northbound Interface in particular, a number of advantages of the peer-to-peer approach to network management were revealed.

Distributed approach without a fixed, central entity

Once the underlying networking infrastructure is set-up, and IP connectivity is established, the Madeira platform provides services that enable the deployment of network management applications in a distributed fashion. No fixed, central entity is necessary for Madeira to fulfil its management tasks. The nodes in the Madeira network management solution communicate with each other and automatically create a hierarchical management overlay, which can be used by management applications, as proven by Configuration Management and Fault Management applications in the prototype.

By eliminating this fixed, central entity, there is no single point of failure. Nodes communicate with each other as peers. They automatically form clusters and dynamically choose a cluster head that is responsible for forwarding information to and from the cluster.

Scalability

To improve the scalability of the network management application, Madeira automatically creates management clusters. Each cluster has a cluster head, and all clusters form a clustering hierarchy. A cluster head of a level *n* management cluster is automatically a member of an *n+1* management cluster. Clusters are created dynamically. When more nodes are added to the network, new clusters are formed and, if necessary, more layers of clustering are added.

The use of this clustering hierarchy can be seen as using mid-level managers, since the network is divided into smaller section for which an entity is responsible. As indicated by [ScQu00] and [MaZn98], this approach has good scalability behaviour compared to a centralised approach since it decreases the computational load per manager and the communicational load on the network. An advantage of the Madeira approach is that clusters and layers are created automatically and dynamically, increasing the flexibility of the network management system [ScQu00].

Manageable from one point

The node at the top of the clustering hierarchy, the top level cluster head, is responsible for the whole network. This node will receive the correlated notifications that are to be forwarded to external Operations Support Systems (OSSs). This node typically runs the Northbound Interface. This interface enables such an external OSS to acquire management information and to perform simple management tasks on the complete network from one single point. Moreover, an OSS can subscribe on notification topics based on the OASIS Web Services Notification standard (see Appendix A – Relevant Web Services standards for more information). When the NBI receives an event or alarm notification from Madeira, this notification will then be transformed and forwarded to all subscribed consumers.

Robust
The dynamic clustering principle of Madeira is already described above. An advantage of the flexible hierarchy is the robustness against changes in topology. In a dynamic environment, nodes can be added or removed constantly. When a node is added, it will try to join an existing management cluster. If this is impossible, it will form its own cluster and become part of the clustering hierarchy. When a node is removed, either intentionally or because of a failure, two different scenarios exist:

- Removal of a cluster head, which is solved by re-clustering and choosing a new cluster head;
- Removal of a cluster member, which does not have a big impact on the hierarchy. Note that this case concerns cluster members on the lowest level. Cluster members on higher hierarchy levels are cluster heads in lower levels.

Depending on the policies, adding and removing of nodes can cause notifications that are forwarded to higher levels.

Open standards and platform independence
The Madeira Northbound Interface uses Web Services to communicate with external Operations Support Systems. The usage of Web Services ensures cross-platform communication [W3C]. Its platform independence combined with the use of open standards enables external entities to easily communicate with the Madeira management system. The Northbound Interface uses well-defined standards like SOAP, UDDI (optional), WSDL and OASIS Web Services Notifications, which are all explained in Appendix A – Relevant Web Services standards.

4.5 Issues and disadvantages

Besides the advantages mentioned in the previous section, a number of issues were also exposed in the design, implementation and testing process. These issues are described in this section.

Setting up the infrastructure
In practice, setting up the wireless mesh infrastructure requires some pre-configuration. This was also encountered while testing the Madeira prototype. On all nodes, the same wireless channel and SSID (Service Set Identifier, to identify the wireless network) had to be chosen. In addition, IP addresses had to be configured, including the gateway address of the node that was connected to the Internet. Only after providing IP connectivity, the Madeira prototype could be executed and tested.

Security
The current design of the Madeira framework does not contain any security measures since this was outside the scope of the project. However, security (e.g. encryption, authentication, non-repudiation) is an important aspect in network management in order to ensure, for example, correct management information. The current design cannot determine malicious nodes. These nodes could send wrong event or alarm notifications for example. This creates the need for a security mechanism in which legitimate peers exchange correct information in a secure manner. However, the lack of a central entity has its consequences on the security aspects of the management application. For example, Public Key Infrastructures (PKIs) typically rely on centralized architectures, where entities authenticate each other via a trusted Certificate Authorities (CAs). This centralized approach is less suitable for peer-to-peer based systems as indicated by [LuZe02] and [AbDa04]. Another solution thus needs to be researched.

Clustering is not based on proximity

Madeira is developed to support different underlying routing mechanisms. The prototype relies on IP connectivity, and uses a simple program which informs each peer of the presence of other peers. It goes without saying that this solution is not very scalable, since every peer is informed of all other peers in the network.

This knowledge is used to create the management clusters. When a node joins the network, it will try to join an existing (level 0) management cluster. If this is not possible, for example when all management clusters already reached their maximum number of peers, the node will form a new management cluster. When a second node joins the network, it will also try to join an existing cluster, and thus will join the newly founded cluster. Unfortunately, information on proximity is not included in this process. This would mean that a (level 0) management cluster can be spread over the complete network. Given the fact that a cluster member communicates with its cluster head (e.g. in case of alarm or event notification), this would cause unnecessary traffic in an environment where bandwidth is already scarce. Figure 22 (left) gives an example of how clustering could look like when node G joined the network and created a new cluster. The figure on the right shows how this cluster 3 looks like after node H joined the network. Dark nodes are cluster heads.

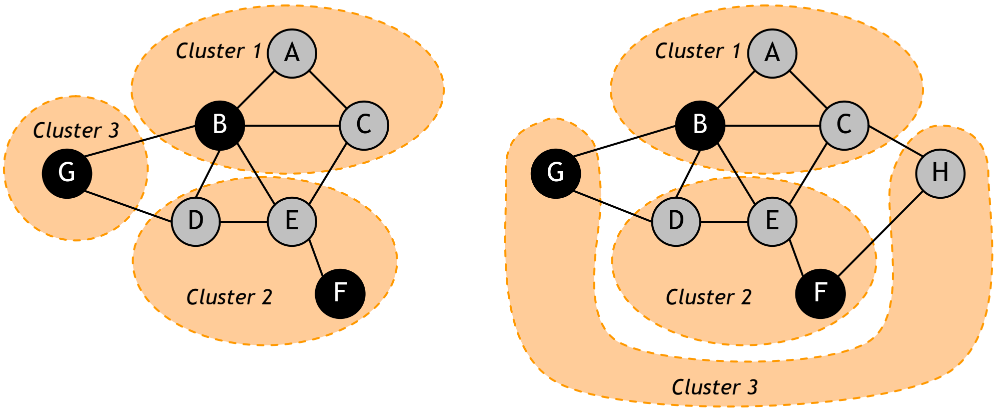


Figure 22: Clustering example

Hierarchical structure

The hierarchical structure of the clustering has a number of advantages already mentioned in the previous section. Although there is no central point of failure due to the flexibility of the clustering algorithm, the top level cluster head is responsible for the whole network. Alarms, for example, travel up the hierarchy and, depending on the policies, could reach this top node. This is inherent to the hierarchical structure, but could eventually lead to traffic or processing bottlenecks when the amount of nodes in the network increases, as indicated by [ChLi02].

5 Conclusion and Future work

5.1 Conclusion

This thesis focused on peer-to-peer principles in the field of network management. After describing different management approaches, peer-to-peer technology was introduced. Peer-to-peer networks have similarities with mobile ad hoc networks, as both provide network connectivity in a decentralized nature. For a specific type of mobile ad hoc network, i.e. a Wireless Mesh Network, the possibilities of a decentralised, peer-to-peer management system are explored, based on experience gained in the Celtic Madeira project. In this international project, peer-to-peer principles are used to develop a completely decentralised network management solution.

The main research question stated in the introduction has been used as a basis for this thesis. Combined with the sub research questions, it offered a guideline for the overall research process. In this conclusion, first the sub questions will be answered, followed by the main research question.

The first sub question, "*What different approaches to network management exist?*", is answered in section 2, where four different management approaches are distinguished. Traditional network management is typically based on a *centralized management approach*, in which a central manager is the only responsible for managing the complete network. Due to the poor scalability of the centralized approach, *weakly distributed management approaches* were developed, which resulted in the introduction of midlevel managers. A midlevel manager acts as both a manager and an agent, and is responsible for managing a part of the network. In both these approaches, agents act as simple data collectors. In the third approach, the *strongly distributed management approach*, agents can execute management tasks themselves and report the results back to the (midlevel) managers, creating a more flexible solution. Finally, in the *cooperative management approach*, the hierarchical communication structure is removed. Moreover, instead of simply executing management tasks, agents 'know' the goal of the task.

For the second sub question, "*In which category does network management based on peer-to-peer principles fit best?*", the four different management approaches are compared with peer-to-peer networks in general and the approach taken in Madeira in specific. Peers in a peer-to-peer network have a high degree of autonomy. Combined with the flexibility to changes in the topology and the decentralized nature, this eliminates the centralized and weakly distributed approaches because of their lack of robustness and the fact that nodes (or agents) act as simple data collectors. As indicated in section 3.2, hybrid peer-to-peer systems make use of super peers. These super peers have a more 'server-like' role in the network since they provide regionally centralised services. They introduce a form of hierarchy, which resembles the strongly distributed approach. In pure peer-to-peer systems, communication is not bound to any hierarchy, which has more similarities to the cooperative management approach. The approach taken in Madeira should be seen as a strongly distributed approach due to the different levels of clustering. In this clustering hierarchy, one node is assigned as top level cluster head. This node is thus responsible for the whole network.

The answer to the third question, "*Is it possible to set up a network management system without pre-configuration or user intervention?*", is based on experience with the Madeira prototype. To enable communication between nodes in a network, IP connectivity is typically required. In a Wireless Mesh Network, this results in some manual pre-configuration (e.g. wireless channel, SSID and IP configuration). After establishing IP connectivity, it is possible to set-up a network management system without pre-configuration or user intervention, as is proven by the Madeira prototype.

To answer the fourth question "*How can one acquire management information about the whole network from a single point in a completely distributed management framework?*", a Northbound extension to the Madeira prototype has been developed. A description of this interface can be found in section 4.2. This so called Northbound Interface enables external entities, like Operations Support Systems, to acquire management information and perform simple management tasks (e.g. disable base-stations or introduce policies) on the whole network from a single point. It is based on Web Services to ensure cross-platform communication. This interface supports 'normal' request/response based communication as well as publish/subscribe based communication, all based on open standards. The node that runs the Northbound Interface can change due to reconfiguration. For the discovery of the node that runs the Northbound Interface, two solutions have been proposed. Due to time constraints, only the UDDI-based solution has been integrated in the prototype.

The answer on the main research question, "*What are the main advantages and disadvantages of using peer-to-peer technology in the field of network management?*" is based on experience gained while cooperating in the Celtic Madeira project, combined with the acquired knowledge researching and answering the sub questions. The following main advantages are discovered:

- No central, single point of failure. Peer-to-peer technology enables the development of a strongly distributed network management approach or a cooperative management approach. Pure peer-to-peer systems do not inhibit hierarchy, which resembles the cooperative management approach where communication is also not bound to a specific hierarchy. Hybrid peer-to-peer systems on the other hand make use of super peers and thus have more similarities with strongly distributed approaches. In both types of peer-to-peer networks, no fixed, central entity is necessary. Super peers can for example be dynamically elected, as is the case in the clustering algorithm in the Celtic Madeira approach.

- Scalability. Peer-to-peer principles can be used to divide the network into smaller clusters. The clustering overlay as used in the Celtic Madeira approach automatically divides the network in small groups to increase the scalability. Each cluster has a cluster head that is responsible for that cluster. In addition, a clustering hierarchy is created. Events and alarms are correlated at each level and, depending on the policies, are forwarded to a higher level. Also Distributed Hash Tables, which can be used to store and lookup data in a completely distributed fashion, provide a highly scalable environment.

- High autonomy. In both strongly distributed and cooperative network management approaches, agents are more than simple data collectors. They are capable of performing network management tasks themselves, and in the case of cooperative management, they even 'know' the goal of the tasks they are performing. This high degree of autonomy is similar for peers in a peer-to-peer network.

- Robust to changes in the topology. Peer-to-peer systems are typically designed to be able to cope with changes in the topology. These changes occur when nodes join or leave the network. In the case of Celtic Madeira, the clustering algorithm is specifically designed to cope with these situations. Changes in this topology automatically trigger re-clustering, ensuring a correct functioning management overlay.

- The usage of Web Services has proven to be a feasible solution to provide external entities access to a decentralized management system. The Northbound Interface, which has been developed for the Celtic Madeira project, enables external entities to acquire management information and perform simple management tasks. In combination with the clustering hierarchy, this interface makes the distributed management system accessible and manageable from one point. Moreover, using Web Services for the Northbound Interface ensures cross-platform communication while the usage of open standards increases the interoperability.

- As also proven by the Celtic Madeira approach, the decentralized nature of peer-to-peer systems can be used to create a distributed management system for decentralized network environments as Mobile Mesh Networks.

Besides these advantages, the following disadvantages were discovered:

- Security. As is the case in both strongly distributed approaches and cooperative management approaches, both approaches have high security requirements. Detecting malicious nodes, encrypting communication, authentication and authorization of users are aspects that have to be taken into account to ensure a secure environment and correct management information. The lack of a central entity in peer-to-peer systems creates difficulties in the field of security. For example, typical Public Key Infrastructures use centralized, trusted Certificate Authorities to authenticate users. For a completely decentralized environment, other solutions need to be researched. These solutions could be more complex.

- A specific disadvantage of the clustering algorithm in the Celtic Madeira approach is the fact that clustering is not based on proximity. The clustering algorithm in Madeira is developed to work on top of any routing protocol. However, due to the scarce availability of bandwidth in Wireless Mesh Networks, effective usage of the routing protocol is also important for the management application. In the current approach, a management cluster could be spread over the whole network. Since a cluster member communicates with its cluster head, it is expected that network traffic could be reduced when ensuring nodes in a cluster to be close to each other.

- Although the hierarchical structure in a strongly distributed management approach improves the scalability compared to the centralised or weakly distributed management approach, one node remains responsible for the complete network. When the size of the network increases, causing more alarm/event notifications for example, traffic and processing bottlenecks could occur at the top level cluster head.

- As already mentioned as a disadvantage in cooperative management approaches, the high degree of autonomy, and the ability to execute management tasks and cooperate with other nodes in the network increases requirements on available resources as processing power and memory for example.

Concluding, the usage of peer-to-peer principles seems promising. They can be used to create strongly distributed and cooperative management approaches, which have better characteristics concerning robustness and scalability then centralized or weakly distributed management approaches. The lack of need for a fixed, central entity in peer-to-peer technology eliminates a single point of failure. The high autonomy of nodes ensures nodes can cooperate with each other to perform management tasks without the help of a central manager. The Celtic Madeira project proved the feasibility of using peer-to-peer principles for the management of Wireless Mesh Networks. When IP connectivity is established, Madeira can automatically create a dynamic, hierarchical management overlay. Moreover, Web Services proved to be an appropriate solution to enable external entities to communicate with a distributed management system because of the platform independence and the usage of open standards. One of these standards is the OASIS Web Services Notification standard, which enables external entities to receive information in a publish/subscribe manner. However, there are a number of issues and disadvantages in using peer-to-peer principles. An important aspect is security. Typical Public Key Infrastructures rely on central entities. When these entities are not present, other solutions are to be researched. The following section contains ideas and elaborations on areas where further research could be aimed on.

5.2 Future work

Incorporate Distributed Hash Tables (DHTs)

Structured peer-to-peer overlay networks based on Distributed Hash Tables (DHT) have gained attention of the research community, given the different algorithms and implementations that have been developed [Pos03][AbDa04][Män05][PiJu06]. A DHT can be used as a distributed storage and lookup service. Even a notification (publish-subscribe) infrastructure has been developed, based on a DHT [RoKe02]. Also for network management, the usage of DHT-based overlays could be useful:

- Store user data. When a user tries to access the management system (or, in case of Madeira, the NBI), the DHT could be accessed to see if the user is authorized to acquire management information or perform management tasks.
- The DHT could also be used as a distributed storage solution to store historic information on, for example, topology or alarms.
- Sending and receiving notifications is already possible [RoKe02], so developing a distributed management solution based on DHTs might also be a possibility. Naturally, research is needed to discover the feasibility.

Peer-to-peer security

A disadvantage mentioned in section 4.5 concerned the security in a completely decentralised peer-to-peer environment. Research is required to ensure a secure environment. In fact, trust in peer-to-peer and ad-hoc networks is an area that has already gained attention of the research community, as proven by [LuZe02], [SiLi03], [AbDa04] and [BeKu04]. For example, it is possible to use a decentralized Public Key Infrastructure (PKI) which is maintained by the participants. In other words, No central controls or Certifying Authorities (CAs) are used. [AbDa04] distinguishes three main subclasses of decentralized PKIs:

1. Web-of-trust
   In a web-of-trust, participants know the public keys of some other peers. The participant can use the knowledge of these other peers to certify the public key of other peers. This model is used in PGP [PGP06] for example. A problem with this approach is that the strength of a trust chain is determined by its weakest link.
2. Statistical approach
   In statistical approaches, the public key information from many peers is obtained and, if it is possible to form a quorum, the public key is considered authentic.
3. Hybrid approach
   In these approaches, public key information is obtained from many peers, after which a weighted quorum is formed. This quorum depends on the relative trust level of the information providing peers.

[AbDa04] presents a solution based on a statistical approach. It uses structured P2P overlays using Distributed Hash Tables, or DHTs (already mentioned in section 3.3). This is considered a better solution than the web-of-trust which uses flooding, and fails to use the collective knowledge of the whole population. Such a system might be implemented in a network management system.

Madeira specific

The Madeira approach is developed for Wireless Mesh Networks. However, the concept of creating a distributed hierarchical clustering overlay in a fully automated manner might also be useful in other types of networks. Research is needed to discover in which situations such a decentralized clustering concept could be required, and if some changes are necessary in the algorithm when it is applied in fixed networks.

Madeira is designed to work on top of any routing protocol. Although this ensures a high degree of flexibility, it also has a disadvantage already explained in section 4.5 concerning wide-spread clusters. A solution for this problem could be to increase the cooperation between the clustering algorithm and the underlying routing protocol. However, research is needed to discover which routing protocol would perform best. Some initial ideas: as already mentioned in section 3.4.1, a proactive routing protocol is less suitable with a high number of nodes. This is the same for the reactive routing protocols when a large amount of route requests are generated because of the flooding-nature of these requests. A hierarchical, cluster-based routing algorithm could be a good place to start. This could enable one-on-one mapping of ‘routing-cluster with Madeira management clusters, which could optimise bandwidth usage in forwarding management information.

Because of re-clustering due to topology changes, section 0 introduces two different methods to discover the NBI node. The UDDI approach uses an external UDDI registry, which does not qualify as a peer-to-peer solution. In the second approach, no central entities are used. Research is needed to discover the feasibility and efficiency of this approach.

Research into peer-to-peer security has already been mentioned. However, also for the Northbound Interface, security is an important element. Aspects as encryption, authentication and authorization for example are necessary to ensure management tasks are performed by authorized users and management information is only acquired by approved entities. A starting point could be the usage of the OASIS Web Services Security standard [WSS06], but there are more solutions like creating an encrypted channel using an SSL (Secure Sockets Layer) connection. For more information on securing Web Services, please refer to [Sun03] and [IbMi02].

To discover if the Northbound Interface could be a possible bottleneck, testing is required to discover the limitations of number of requests per second, the number of notifications per second, maximum number of consumers and the maximum size of the requests/notifications for example. Load balancing might be required to solve possible problems. This could be achieved by letting another node run the NBI if the top level cluster head cannot handle the load. Another solution could be by allowing multiple instances of the Northbound Interface on different nodes. Requests could then be handled by the NBI with the lowest load. Allowing multiple NBIs might, however, create difficulties in storing subscription information for notification topics. This might be solved by storing this information in a distributed fashion, for example by using a structured P2P overlay based on Distributed Hast Tables.

Another improvement for the Northbound Interface could involve using a different XML parser. As indicated in [ElPa02], the flexibility and interoperability of the SOAP protocol has been partly achieved at the expense of run-time performance. To improve this performance, it is important to (1) handle communication as efficient as possible, and to (2) choose a parser that is suitable for the type of XML documents used in the Web Service applications [ElPa02]. In the current prototype, the default Apache Xerces parser is used. Performance testing is needed to discover which other parser would perform better. In [MaFe05], a number of guidelines are developed to evaluate this performance, and suggestions are made to make improvements.

References

[AbDa04] Abere, K., Datta, A. & Hauswirth, M. (2004). A decentralized public key infrastructure for customer-to-customer e-commerce. *International Journal of Business Process Integration and Management, Volume X, No X, 2004.*

[AkWa05] Akyildiz, I., Wang, X. & Wang, W. (2005). Wireless mesh networks: a survey. *Computer Networks 47. pp. 445-487.*

[AnDa95] Anderson, T.E., Dahlin, M.D., Neefe, J.M., Patterson, D.A., Roselli, D.S., Wang, R.Y. (1995). Serverless Network File System. In *Proceedings of the 15th Symposium on Operating System Principles. ACM.*

[ArFr06] Arozarena, P., Frints, M., Collins, S., Fallon, L., Zach, M., Serrat, J. & Nielsen, J. (April 2006). Madeira: A peer-to-peer approach to network management. *WWRF16, 26-28 April 2006, Shanghai, China.*

[BaSc06] Baset, S.A. & Schulzrinne, H. (April 2006). An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *Proceedings of the INFOCOM '06 (Barcelona, Spain, April 2006).*

[BeKu04] Bearly, T & Kumar, V. (2004). Expanding Trust Beyond Reputation in Peer-To-Peer Systems. In *Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04).*

[BiGu04] Bivens, A. & Gupta, R. (2004). Scalability and performance of an agent-based network management middleware. *Int. J. Network Mgmt, 14, pp. 131–146.*

[BrGa05] Brunner, M., Galis, A., Cheng, L., Andres Colas, J., Ahlgren , B., Gunnar, A., Abrahamsson, H., Szabo, R., Csaba, S., Nielssen, J., Gonzalez Prieto, A., Stadler , R., Molnar, G. (October 2005). Towards Ambient Networks Management. *Second International Workshop on Mobility Aware Technologies and Applications (MATA 2005), Montreal, Canada, October 17-19, 2005.*

[ChLi02] Chen, T. & Liu, S. (2002). A Model and Evaluation of Distributed Network Management Approaches. *IEEE Journal on Selected Areas in Communication, 28(4), pp. 850-857.*

[ChMo02] Chen, B. & Morris, R. (March 2002). L+: Scalable Landmark Routing and Address Lookup for Multi-hop Wireless Networks. *MIT LCS Technical Report 837, March, 2002.*

[DrRo01] Druschel, P. & Rowstron, A. (2001). PAST: A large-scale, persistent peer-to-peer storage utility. *Proceedings of the 18th ACM Symposium on Operating Systems Principles, pp. 188-201.*

[ElPa02] Elfwing, R., Paulsson, U. & Lundberg, L. (2002). Performance of SOAP in Web Service environment compared to CORBA. In *Proceedings of the Ninth Asia-Pacific Software Engineering Conference, 2002.*

[FoRa04] R. Fonseca, S. Ratnasamy, D. Culler, S. Shenker & I. Stoica, Fonseca, R., Ratnasamy, S., Culler, D., Shenker, S. & Stoica, I. (May 2004). Beacon Vector Routing: Scalable Point-toPoint in Wireless Sensornets. *Technical Report IRBTR - 04-12, Intel Research Berkeley, May, 2004.*

[GeLa05] Ge, Y., Lamont, L. & Villasenor, L. (August 2005). Hierarchical OLSR – S scalable proactive routing protocol for heterogeneous ad hoc networks. In *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), Montreal, Canada, August 22-24, 2005.*

[Gnu00] Gnutella (2000). The Gnutella protocol specification v0.4. *Retrieved October 2, 2006, from http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.*

[GoYe95] Goldszmidt, G. & Yemini, Y. (June 1995). Distributed Management by Delegation. In *Proceedings of the 15th International Conference on Distributed Computing Systems. IEEE Computer Society, Vancouver, British Columbia, June*

1995.

[HaCh95] Harrison, C.G., Chess, D.M. & Kerschenbaum, A (1995). Mobile Agents: Are they a good idea? *IBM Research Report, IBM Research Division, Yorktown Heights, NY.*

[HaPe02] Haas, Z., Pearlman, M. & Samar, P. (July 2002). The Zone Routing Protocol (ZRP) for Ad Hoc Networks. *IETF Internet Draft.*

[HuSa03] Hu, Y.C., Saumitra, M.D. & Pucha, H. (May 2003). Exploiting the synergy between peer-to-peer and Mobile Ad Hoc Network. In *Proceedings of HotOS-IX, Hawaii, pp. 37-42.*

[IbMa04] Ibriq, J. & Mahgoub, I. (July 2004). Cluster-based routing in wireless sensor networks: Issues and challenges. In I*nternational Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), San Jose, California, USA, July 2004, pp. 759-766.*

[IbMi02] IBM Corporation & Microsoft Corporation (April 2002). Security in a Web Services World: A Proposed Architecture and Roadmap. *A joint whitepaper from IBM Corporation and Microsoft Corporation.*

[ITU92] ITU (1992). Recommendation X.733 – Systems Management: Alarm Reporting Function.

[JaMu01] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A. & Viennot, L. (December 2001). Optimized link state routing protocol for ad hoc networks. In *Procedings of Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. IEEE International. pp. 62 – 68.*

[JoMa01] Johnson, D., Maltz, D., Broch, J. (2001). DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Ad Hoc Networking, pp.139–172, 2001.*

[KoVa00] Ko, Y. & Vaidya, N. (July 2000). Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. *In Wireless Networks Volume 6, Issue 4 (July 2000) pp. 307-321, Kluwer Academic Publishers, Hingham, MA, USA.*

[KoXu02] Kona, M. & Xu, C. (2002). A Framework for Network Management using Mobile Agents. In *Proceedings of the Parallel and Distributed Processing Symposium, 2002, pp. 227-234.*

[LiTh03] Li. L., Thottan, M., Yao, B. & Paul, S. (2003). Distributed Network Monitoring with Bounded Link Utilization in IP Networks. Bell Labs. *IEEE Infocom 2003.*

[LoOl00] Lopes, R. & Oliveira, J. (2000). On the use of mobility in distributed network management. In *Proceedings of the 33rd Hawaii International Conference on System Sciences.*

[LuCa03] Lu, J. & Callan, J. (2003). Content-Based Retrieval in Hybrid Peer-to-Peer Networks. *Proceedings of the Twelfth International Conference on Information and Knowledge Management. (CIKM-03). ACM Press.*

[LuZe02] Luo, H., Zerfos, P., Kong, J., Lu, S. & Zhang, L. (2002). Self-securing Ad Hoc Wireless Networks. In *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02).*

[Mad06] Celtic Initiative (n.d.). Celtic Madeira project website. *http://www.celtic-madeira.org. Last visited: January 10, 2006.*

[MaFe05] Machado, A. & Ferraz, C. (2005). Guidelines for performance evaluation of web services. In *Proceedings of the 11th Brazilian Symposium on Multimedia and the web, Pocos de Caldas - Minas Gerais, Brazil, 2005, pp. 1-10.*

[Män05] Mäntylä, J. (2005). Scalability of peer-to-peer systems. *Seminar on Internetworking, Spring 2005.*

[MaZn97] Martin-Flatin, JP. & Znaty, S. (March 1997). Annotated Typology of Distributed Network Management Paradigms. *Technical Report SSC/1997/008, SSC, EPFL,* *Lausanne Switzerland, March 1997.*

[MaZn98] Martin-Flatin, JP., Znaty, S. & Hubeax, JP. (August 1998). A survey of distributed network and system management paradigms. *EPFL, SSC, Lausanne, Switzerland.*

[Nap00] Napster (April 2000). The Napster protocol. *Retrieved October 2, 2006, from http://opennap.sourceforge.net/napster.txt.*

[OvPo02] Overeinder, B.J., Posthumus, E. & Brazier, F.M.T. (2002). Integrating peer-to-peer networking and computing in the AgentScape framework. In *Proceedings of the 2nd IEEE International Conference on Peer-to-Peer Computing, pp. 96-103.*

[PeBh94] Perkins, C., Bhagwat, P. (1994). Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proceedings of ACM SIGCOMM#94, London, UK, 1994, pp. 234-244.*

[PeGe00a] Pei, G., Gerla, M. & Chen, T-W. (2000). Fisheye state routing in mobile ad hoc networks. In *Proceedings of ICDCS Workshop on Wireless Networks and Mobile Computing. April 2000, pp. D71-D78.*

[PeGe00b] Pei, G., Gerla, M. & Hong, X. (August 2000). LANMAR: Landmark based routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of IEEE/ACM MobiHOC 2000, Boston, MA, August 2000, pp. 11-18.*

[PeGe99] Pei, G., Gerla, M., Hong, X. & Chiang, C.-C. (1999). A Wireless Hierarchical Routing Protocol with Group Mobility. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 1999. WCNC.*

[PeRo99] Perkins, C., Royer, E. (February 1999). Ad¤hoc On¤Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.*

[PGP06] PGP (n.d.). PGP website. *http://www.pgpi.org. Last visited: October 19, 2006.*

[PiJu06] Pinart, C. & Junyent, G. (April 2006). Integration of peer-to-peer strategies and SOAP/XML for inter-domain, user-driven provisioning in an ASON/GMPLS network. In *OSA Journal of Optical Networking, Vol. 5, Nb. 4, pp. 246-262, April 2006.*

[Pos03] Post, A. (August 2003). Towards a scalable ad hoc network infrastructure. In *Proceedings of the First IRIS Student Workshop (ISW'03) Boston, MA.*

[Pos05] Post, A. (May 2005). Strata: A simple lightweight ad hoc communications infrastructure. *Masters Thesis, Rice University, Houston, TX, USA.*

[PrBe99] Pras, A. & van Beijnum, B.J.F. (1999). Introduction to TMN. *CTIT Technical Report 99-09.*

[Pub06] Apache (n.d.). Apache Pubscribe website. *http://ws.apache.org/pubscribe/. Last visited: June 15, 2006.*

[RaHa03] Ramasubramanian, V., Haas, Z., Sirer, E. (June 2003). SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), Annapolis, Maryland, June 2003.*

[RFC1157] Case, J., Fedor, M., Schoffstall, M. & Davin, J. (May 1990). A Simple network Management Protocol (SNMP), IETF, RFC1157.

[RFC3410] Case, J., Mundy, R., Partain, D., Stewart, B. (December 2002). RFC3410: Introduction and Applicability Statements for Internet-Standard Management Framework. *IETF, RFC3410.*

[RhEa03] Rhea, S., Eaton, P., Geels, D., Weatherspoon, H., Zhao, B, & Kubiatowicz, J. (March 2003). Pond: the OceanStore Prototype. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST '03), March 2003.*

[RoHo03] Romijn, W., Hoekstra, G., Serrat, J. & Grampin, E. (2003). Enhancing Network Management by Applying Policy Management Principles, *Bell Labs Technical*

Journal, Vol.8, Nº 1, 151-156, 2003.

[RoKe02] Rowstron, A., Kermarrec, A., Castro, M. & Druschel, P. (November 2001). SCRIBE: The design of a large-scale event notification infrastructure. In *Proceedings of 3rd International Workshop on Networked Group Communication (NGC2001), UCL, London, UK, November 2001.*

[Sam04] Samaras, G. (2004). Mobile Agents: What about them? Did they deliver what they promised? Are they here to Stay? In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04).*

[ScEy00] Schoder, D. & Eymann, T. (2000). The Real Challenges of Mobile Agents. *Communications of the ACM, 43(6).*

[ScGr02] Schollmeier, R., Gruber, I. & Finkenzeller, M. (May 2002). Routing in Mobile Ad Hoc and Peer-to-Peer Networks. A Comparison. In P*roceedings of Networking 2002. International Workshop on Peer-to-Peer Computing, Pisa, It*aly. May 19-24, 2002.

[Scho01] Schollmeier, R. (2001). A definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. *First International Conference on Peer-to-Peer Computing, pp. 101-102.*

[ScQu00] Schönwälder, J., Quittek, J. & Kappler, C. (2000). Building Distributed Management Applications with the IETF Script MIB. *IEEE Journal on Selected Areas in Communication, 18(5), pp. 702-713.*

[Sha02] Sharma, A., (September 2002). The FastTrack Network. *PC Quest Magazine, India. Retrieved October 2, 2006, from http://www.pcquest.com/content/p2p/102091205.asp.*

[Shi00] Shirky, C. (2000). What is P2P … and what isn't. *Retrieved February 13, 2006, from http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html.*

[SiLi03] Singh, A. & Liu, L. (September 2003). TrustMe: Anonymous Management Trust Relationships in Decentralized P2P Systems. In *Proceedings of the Third International IEEE Peer-to-Peer Computing, September 2003.*

[Sub00] Subramanian, M. (2000). Network Management: An introduction to principles and practice. *Addison-Wesley, 1995.*

[Sun03] Sun Microsystems (2003). Securing Web Services — Concepts, Standards, and Requirements. *White Paper.*

[TsSo00] Tsatsoulis, C. & Soh, L. (2000). Intelligent Agents in Telecommunication Networks. *Computational Intelligence in Telecommunication Networks.*

[UDD06] OASIS (n.d.). OASIS UDDI Specification. *Retrieved June 14, 2006, from http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm.*

[W3C] W3C (n.d.). W3C Web Services Architecture website. *http://www.w3.org/TR/ws-arch/. Last visited: August 21, 2006.*

[Well96] Wellens, C. & Auerbach, K. (1996). Towards Useful Management. *The Simple Times, 4(3).*

[WSN06] OASIS (n.d.). OASIS Web Services Notification website. *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn. Last visited: October 11, 2006.*

[WSS06] OASIS (n.d.). OASIS Web Services Security website. *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss. Last visited: October 19, 2006.*

Appendices

Appendix A – Relevant Web Services standards

Standards

Description

According to W3C, a Web Service is "*a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.*" [W3C]. The figure below depicts the typical communication structure when using Web Services.

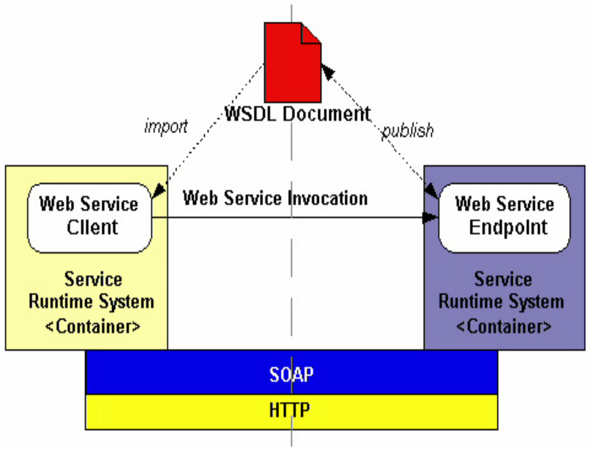


Figure 23: Web Services communication

SOAP

When using Web Services, XML based messages are exchanged between two Web Services Endpoints. The SOAP protocol is an XML messaging protocol [SOA03]. A SOAP message typically consists of a header-part that contains some addressing information and a body-part that contains the actual message contents. The header and body are packed into an envelop. For the transport of these SOAP messages, the HTTP protocol is mostly used. Other protocols as FTP or SMTP for example can also be used, but are less common.

Web Services Description Language, or WSDL

To describe the public interface of a Web Service, the WSDL (Web Services Description Language) has been developed. This XML based language describes how a client can communicate with the Web Service [WSD01]. It mainly contains information on message formats, operations and protocol bindings. The structure of a WSDL document is as follows:

- name space imports
- types (defines message formats for example)
- messages
- portType (defines operations)
- binding (defines protocol bindings)
- service (defines service)

Universal Description, Discovery and Integration, or UDDI

UDDI is an XML-based registry in which businesses worldwide can list themselves on the Internet. It is an open industry initiative that enables businesses to publish services they offer, including instructions on how these services can be accessed [UDD06].
Combined with SOAP and WSDL, these three technologies are considered to be the basis of Web Services. It is designed to be interrogated by SOAP messages and to provide access to WSDL documents describing the protocol bindings and message formats required to interact with the web services listed in its directory.

Web Services Resource Framework, or WSRF

The Web Service Resource Framework (WSRF) is an OASIS standard. Normally, a Web Service by itself is stateless. In other words, it does not retain data between invocations of clients. This is a disadvantage which limits the functionality of Web Services. Workaround for this problem exists, for example by using an external database to store the information and having the Web Service access this database.

Instead of using workarounds to acquire this 'stateful' functionality, WSRF provides a clean set of methods which allow data to be stored and retrieved in the Web Service. This data is stored as "ResourceProperties". These ResourceProperties are declared in the WSDL file. WSRF provides methods to set and acquire the value of a certain ResourceProperty [WSR06a]. The following operations are available by default:

- GetResourceProperty, get a single ResourceProperty, based on the qname (namespace and ResourceProperty name);
- GetMultipleResourceProperties, get multiple ResourceProperties at the same time;
- SetResourceProperties, set the value of a single ResourceProperty;
- QueryResourceProperties; acquire a list of all the available ResourceProperties.

The piece of 'code' below is an example of the ResourceProperties declaration in a WSDL file. It contains two different properties, ConnectionStatus and LastLogin. Both properties are defined in the 'types' namespace, hence the "types:...".

```
<element name="ResourceProperties">
<complexType>
        <sequence>
                <element ref="types:ConnectionStatus" />
                <element ref="types:LastLogin" />
        </sequence>
</complexType>
</element>
```

Web Services Notifications, or WSN

Since Web Services are request/response based, where requests are initiated by the client and the server took care of the response, no support for notifications was initially present. The OASIS Web Services Notification [WSN06] standard offers this ability in a publish/subscribe way. It enables clients to subscribe for a certain topic and the server to send notification messages to all subscribed consumers.

WSN builds on top of WSRF. The topics a client can subscribe to are defined as ResourceProperties. Upon subscribing, a client can specify a subscription expiry time.

Notifications are SOAP messages that are sent to a certain Web Services Endpoint. The address of this endpoint is specified by the client on subscribing for a topic. [WSN06] recommends that the consumer also implements a simple Web Service to receive and handle the notification messages. This Web Service then has to implement a method called "Notify".

The 'normal' notification approach is described in the WS-BaseNotification specification. However, WSN also contains the WS-BrokeredNotification specification in which notifications that were created by other entities are reproduced by the NotificationProduces.

An example scenario: A client subscribes on a certain topic (i.e. ResourceProperty). When the value of this ResourceProperty is changed, the server will send a notification indicating the old and new value of the property to all subscribed consumers.

Another Web Services specification that offers support for notifications is Web Services Eventing [WSE04]. This specification is mainly developed by Microsoft and IBM and is quite similar to the OASIS standard. At the time of writing, WS-Eventing is supported by Microsoft .Net. Organizations from both standards are currently working together to "*align the two specifications and reduce potential for overlap and incompatibilities*" [WSE06].

Web Services Distributed Management, or WSDM

The OASIS Web Services Distributed Management [WSD06] consists of two sets of specifications:

- Web Services Distributed Management: Management Using Web Services (MUWS) [MUW04a] defines how management of any resource can be accessed via Web Services protocols

- Web Services Distributed Management: Management Of Web Services (MOWS) [MOW04] specifications defines how the above mentioned MUWS can be used to manage Web Services endpoints. It can be seen as a domain specific application of MUWS: management of Web Services, using Web Services. Therefore, MOWS will remain out of the scope of this thesis.

The main idea of MUWS is that every manageable entity in the network is a Web Service Endpoint. In other words, every entity implements a Web Service that supports MUWS. This allows a user (e.g. a management system or a real-life user) to perform management tasks, acquire management information and subscribe for notifications. The Web Service can be implemented by the resource itself, or by an agent.

MUWS builds heavily on WSRF and WSN specifications. The WSRF ResourceProperties represent actual properties of the managed resource. To inform the manager of changes for example, it uses WSN notifications.

The notifications sent by MUWS have a number of predefined fields. These fields are defined in the MUWS part 1 specification [MUW04a]

• ReportTime	Date and time when event was reported	(optional)
• EventID	Primary identifier of even	(required)
• SourceComponent	Identification of source of event	(required)
• ReportedComponent	Identification of reporter	(required)

Besides these fields, any other XML content can be added. The MUWS part 2 specification [MUW04b] contains some additional elements that could be included.

MUWS contains a number of predefined topics. These topics are declared in [MUW04c].

Implementation

The Apache Software Foundation Web Services Project [Apa06] focuses on open source software implementation of different areas in Web Services. Apache Axis [Axi06] is an implementation of the SOAP standard, and exists in a Java and C++ version. It is a web application that needs a web container or a web application server to run. The recommended solution is to use Apache Tomcat as its web container.

Apache has also developed a Java-based build tool called Ant. With the help of Ant, it is possible to create all necessary Java classes based on a WSDL file. In other words, when one specifies different message types etc. in the WSDL, Ant will create the corresponding Java source files.

An implementation of the Web Service Resource Framework is provided by the Apache WSRF [WSR06b] project. It is built upon Apache Axis, so naturally it offers SOAP support. It also incorporates the Ant-based WSDL-to-Java generator. It runs as a similar web application as Axis in a web container like Apache Tomcat.

Depending on the information contained in the WSDL file, Apache WSRF supports all operations described above. By changing the WSDL file, one or more of these operations can be disabled.

An implementation in Java of the OASIS Web Services Notification standard in general and the WS-BaseNotification specification in particular is provided by the Apache Pubscribe project [Pub06]. At the time of writing, the latest version is Apache Pubscribe 1.1. Since it uses ResourceProperties as topics, Apache Pubscribe is built on top of Apache WSRF. In other words, installing the Apache Pubscribe web application enables support for SOAP messages and the Web Services Resource Framework.

Apache Pubscribe can be run in two ‘modes’: producer and consumer mode. In producer mode, Apache Pubscribe acts as the server to which clients can subscribe. It exports topics that can be found in the WSDL description. Apache Pubscribe takes care of the subscription process and notification process. The Web Service that runs in Apache Pubscribe only has to tell it to send a certain notification to all subscribed consumer on a specific topic.

The second mode is the consumer mode. In consumer mode, Apache Pubscribe exports the “Notify” method which is called when a notification is received. What happens after receiving a notification is up to the programmer.

A Java implementation of the OASIS WSDM specification is provided by the Apache Muse project [Mus06]. It is built on top of Apache WSRF and Apache Pubscribe. However, at the time of writing and prototype development, the Muse project used Apache Pubscribe 1.0.

References

[Apa06] Apache Web Services Project website. *http://ws.apache.org/. Last visited: June 15, 2006.*

[Axi06] Apache Axis website. *http://ws.apache.org/axis/. Last visited: June 15, 2006.*

[MOW04] OASIS (2004). OASIS WSDM MOWS 1.0 Specification. *Retrieved June 14, 2006, from http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0.pdf.*

[Mus06] Apache Muse. *http://ws.apache.org/muse/. Last visited: 14 June 2006.*

[MUW04a] OASIS (2004). OASIS WSDM MUWS 1.0 Part 1 Specification. *Retrieved June 14, 2006, from http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf.*

[MUW04b] OASIS (2004). OASIS WSDM MUWS 1.0 Part 2 Specification. *Retrieved June 14, 2006, from http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf.*

[MUW04c] OASIS (2004). OASIS WSDM MUWS 1.0 Part 2 – Topics. *Retrieved June 14, 2006, from http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2-events.xml.*

[Pub06] Apache Pubscribe website. *http://ws.apache.org/pubscribe/. Last visited: June 15, 2006.*

[SOA03] W3C (June 2003). SOAP Version 1.2 Part 1: Messaging Framework. *Retrieved June 13, 2006, from http://www.w3.org/TR/soap12-part1/.*

[Tom06] Apache Tomcat website. *http://tomcat.apache.org/. Last visited: June 15, 2006.*

[UDD06] OASIS (n.d.). OASIS UDDI Specification. *Retrieved June 14, 2006, from http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm.*

[W3C] W3C (n.d.). Web Services Architecture. *Retrieved June 13, 2006, from http://www.w3.org/TR/ws-arch/.*

[WSD01] W3C (March 2001). Web Services Description Language. *Retrieved June 13, 2006, from http://www.w3.org/TR/wsdl.*

[WSD06] OASIS Web Services Distributed Management website. *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm. Last visited: June 14, 2006.*

[WSE04] IBM (August 2004). Web Services Eventing specification. *Retrieved June 14, 2006, from ftp://www6.software.ibm.com/software/developer/library/ws-eventing/WS-Eventing.pdf.*

[WSE06] IBM Web Services Eventing website. *http://www-128.ibm.com/developerworks/library/specification/ws-eventing/. Last visited: September 18, 2006.*

[WSN06] OASIS Web Services Notification website. *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn. Last visited: June 14, 2006.*

[WSR06a] OASIS Web Services Resource Framework website. *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsr. Last visited: June 14, 2006.*

[WSR06b] Apache WSRF website. *http://ws.apache.org/wsrf/. Last visited: June 14, 2006.*

Appendix B – NBI Requirements

This section lists specific high-level requirements on the Northbound Interface.

Functional Requirements

From OSS to Madeira

ReqNBI-F-1: Topology Information
Madeira shall provide topology information to external OSS. It shall include:

- Information about base stations present in the network, including capabilities, resources, etc.
- Physical' links between the base stations
- Connection to the backhaul network
- Information about wireless equipment connected to base stations

ReqNBI-F-2: Clustering Information
Madeira shall provide clustering information to external OSS. It shall include:

- Management clusters
- Cluster heads
- Cluster hierarchy

ReqNBI-F-3: Simple Configuration Management Actions
NBI shall allow external OSS to perform some simple configuration management actions within the MADEIRA platform, such as:

- Disable a Base Station

ReqNBI-F-4: Policies Management
External OSS shall be able to control/configure the use of policies within the Madeira network. NBI shall accept commands related to the policies management, such as:

- Introduce Policy, which enables the operator to add new policies, change existing policies, and remove or disable policies
- Get an overview of all policies that are present in Madeira

ReqNBI-F-5: Get Active Alarms
Madeira is able to perform self management. An external OSS is not necessary to perform management tasks. This means that an OSS does not always have to be connected to Madeira. This, however, implies the need for a method in order to enable the OSS to acquire the currently active alarms, that could have occurred in the period there was no OSS connected.

ReqNBI-F-6: Discover NBI address
The OSS has to implement functionality in order to discover the address of the NBI. Furthermore, the OSS has to be able to detect a change of NBI node, for example when a reconfiguration is triggered in Madeira.

From Madeira to OSS

Notifications are required by the Madeira NBI in order to convey status changes, alarms, etc. in an asynchronous fashion to an external OSS. One or more OSSs will be able to subscribe for events in Madeira. Once an OSS has subscribed to Madeira, it will receive notifications sent by the management system. These notifications are the final correlated alarms from FM and events from CM applications.

ReqNBI-F-7 Notifications about alarms
Information about final alarms shall be delivered to the OSS. The operator looking at this alarm shall know all relevant information regarding the element or elements affected by it. It will include information on base station removal, disabling and failure for example. Additional information includes the affected base station, reporting base station, severity and probable cause.

ReqNBI-F-8 Notifications about network reconfiguration
Information about reconfiguration of the network shall be delivered to the OSS. It also indicates the reason for reconfiguration (base station addition, removal, etc.).

ReqNBI-F-9 Notifications about clustering
Information about the management clusters in the network shall be delivered to the OSS. This could include changes in cluster heads for example.

ReqNBI-F-10 Notifications about wireless equipment
Madeira shall provide information about wireless equipment being connected to / disconnected from the network.

ReqNBI-F-11: Isolated network regions
Madeira shall provide sufficient information in order to enable an OSS to discover possible isolated network regions that are present in the Network.

Other

This section includes requirements that are not related to either of the two sections above

ReqNBI-F-12: Publish NBI address
Madeira shall provide a solution in order to enable the OSS to discover the address of the NBI. It will publish this address in an external UDDI registry. Furthermore, an additional peer-to-peer solution will be investigated.

ReqNBI-F-13: Security
Madeira will offer security mechanisms in order to assure secure and authorized communication between an OSS and the NBI.

Non-Functional Requirements

In principle, all of the non-functional requirements listed in [REQ] are also relevant for the Madeira NBI. In the following, the highest priority requirements considered in NBI design are listed.

ReqNBI-NF-1: Scalable
The NBI should work independent of the number of NEs present in the network, the number of requests being processed, the number of notifications being processed and the number of OSSs that are subscribed.

ReqNBI-NF-2: Handle requests
The NBI will be able to handle requests from one or more OSSs and pass information back to them.

ReqNBI-NF-3: Handle notifications
The NBI will be able to handle notifications sent from Madeira and forward the information to all subscribed OSSs.

ReqNBI-NF-4: Openness
The NBI will use open standards as much as possible to increase the interoperability and the ease of use of the Madeira management system.

ReqNBI-NF-5: Recover from network reformation
The NBI will automatically restart on the top level cluster head in case of network reformation.

ReqNBI-NF-6: Testing
The NBI can be tested against all functional and non-functional requirements.

ReqNBI-NF-7: Architecture independent
The NBI can run on any computer whose OS supports Java.

ReqNBI-NF-8: OSS reference implementation
A OSS reference implementation will be provided that is able to use all aspects of the NBI.

Summary and priorities

The summary below includes priority information on all requirements described above

OSS to Madeira		**Priority**
ReqNBI-F-1	Topology Information	high
ReqNBI-F-2	Clustering Information	high
ReqNBI-F-3	Simple Configuration Management Actions	medium
ReqNBI-F-4	Policies Management	high
ReqNBI-F-5	Get Active Alarms	low
ReqNBI-F-6	Discover NBI address	medium

Madeira to OSS		**Priority**
ReqNBI-F-7	Notifications about alarms	high
ReqNBI-F-8	Notifications about network reconfiguration	high
ReqNBI-F-9	Notifications about clustering	high
ReqNBI-F-10	Notifications about wireless equipment	medium
ReqNBI-F-11	Isolated network regions	medium

Other		**Priority**
ReqNBI-F-12	Publish NBI address	medium
ReqNBI-F-13	Security	low

Non-Functional		**Priority**
ReqNBI-NF-1	Scalable	high
ReqNBI-NF-2	Handle requests	high
ReqNBI-NF-3	Handle notifications	high
ReqNBI-NF-4	Openness	high
ReqNBI-NF-5	Recover from network reformation	high
ReqNBI-NF-6	Testing	medium
ReqNBI-NF-7	Architecture independent	medium
ReqNBI-NF-8	OSS reference implementation	high