Self-organising Management Overlays for Future Internet Services

Lawrence Cheng[1], Alex Galis[1], Bertrand Mathieu[2], Kerry Jean[1], Roel Ocampo[1,3], Lefteris Mamatas[1], Javier Rubio-Loyola[4], Joan Serrat[4], Andreas Berl[5], Hermann de Meer[5], Steven Davy[6], Zeinab Movahedi[7], Laurent Lefevre[8]

[1]University College London, UK {a.galis, l.cheng, k.jean, r.ocampo, l.mamatas}@ee.ucl.ac.uk
[2]France Telecom, France { bertrand2.mathieu@orange-ftgroup.com}
[3]University of the Philippines, Philippines {roel@eee.upd.edu.ph}
[4]Universitat Politècnica de Catalunya, Spain {jrloyola, serrat}@tsc.upc.edu
[5]University of Passau, Germany {andreas.berl, demeer@uni-passau.de}
[6]Waterford Institute of Technology, Ireland {sdavy@tssg.org}
[7]Université Pierre et Marie Curie, France {zeinab.movahedi@lip6.fr}
[8] INRIA RESO, France {laurent.lefevre@ens-lyon.fr}

Abstract. Networks are becoming service-aware implying that all relevant business goals pertaining to a service are fulfilled, and also the network resources are used optimally. Future Internet Networks (FIN) have time varying topology (e.g. such networks are envisaged in Autonomic Internet [1], FIND program [2], GENI program [3], FIRE program [4], Ambient Networks [5], Ad-hoc networks [6]) and service availability and service context change as nodes join and leave the networks. In this paper we propose and evaluate a new self-organising service management system that manages such changes known as the Overlay Management Backbones (OMBs). The OMB is a self-organising solution to the problem space in which each OMB node is dynamically assigned a different service context task. The selection of OMB nodes is conducted automatically, without the need of relatively heavy-weighted dynamic negotiations. Our solution relies on the scalability and dynamicity advantages of Distributed Hash Tables (DHTs). This system is needed to select continuously, automatically, and dynamically a set of network nodes, to become responsible for collecting the availability information of service context in the changing network. This solution advances the state of the art avoiding dynamic negotiations between all network nodes reducing management complexity and cost for bandwidth-limited environments.

Keywords: Self-organised management, Autonomic Internet, Distributed hash tables, Peer-to-Peer

1. Introduction

Recently, the use of structured and unstructured Peer-to-Peer (P2P) networks for supporting multimedia services in the Internet, such as P2P streaming, has attracted a great deal of attention. Unstructured P2P networks, such as Gnutella [7], BitTorrent [8], Freenet [9], and more, organise peers in a random graph in flat or hierarchical manners (e.g. SuperPeers layer) [10]. In contrast, structured P2P networks, such as Content Addressable Networks (CANs) [11], Pastry [12], Chord [13], and more, assign keys to peers, organising them and mapping data items to such keys. However, the possibility of

utilising the scalable and decentralisation features of P2P techniques [14] for solving service management challenges in large-scale, heterogeneous networks with time varying topology such as Future Internet Networks (FINs) has not been fully realised.

FINs consist of nodes that are heterogeneous end user devices such as mobile phones, MP3 players, PDAs, servers, etc., that are sharing services through wired or bandwidth-limited channels. FIN nodes [1][2][3] are dynamic, that they may join in and leave a service domain (i.e. P2P services, connectivity services, etc) at any time. Services reside on nodes; thus, as nodes join and leave the network, service availability in the network changes. The major challenge in developing a common control space for service management in dynamically changing, bandwidth-limited networks is to identify the availability of different types of services that are currently available to end users. Note that an end user service may be implemented by composing several services.

In order to determine the availability of different types of services in the network, as with WebServices [15], a consistent and distributed context service directory is needed. This service directory should be distributed allowing the system to be scalable. This paper presents a novel protocol, known as the Overlay Management Backbones (OMBs), which assigns different service availability monitoring tasks to peers in a dynamically changing network in an automatic and self-organising manner [16] hence the title of this paper. This protocol involves selecting and continuously maintaining a set of nodes in the network to act as distributed service directories. These service directories keep track of the different types of services that are currently available in the network, and are able to re-direct consumers or other management entities to access the services that are needed to implement more tailor-made service(s) that consumers want. Our solution achieves these goals, by utilising structured P2P systems i.e. Distributed Hash Tables (DHTs).

The Section 2 of this paper provides background information of the concepts addressed in this paper. Section 3 describes the OMB protocol and Section 4 presents some evaluation results. Finally Section 5 concludes the paper and gives some further work.

2. Background

2.1 Views on Autonomic Internet /Future Internet Networks

Networks are becoming service-aware. Service awareness means not only that all digital items pertaining to a service are delivered but also that all business or other relations pertaining to a service offer are fulfilled and the network resources are optimally used in the service delivery. In addition, the network's design is moving towards a different level of automation and self-management. The solution envisaged in [1][17] is based upon an optimised network and service layers solution which guarantees built-in orchestrated reliability, robustness, mobility, context, access, security, service support and self-management of the communication resources and services. It suggests a transition from a service agnostic Internet to service-aware network, managing resources by applying Autonomic principles as depicted in Figure 1. In order to achieve the objective of service-aware resources and to overcome the ossification of the current Internet, [1][17] aims to develop a self-managing virtual resource overlay that can span across heterogeneous networks and that supports service mobility, security, quality of service and reliability. In this overlay network, multiple virtual networks could co-exist on top of a shared substrate with uniform control. Ambient networking [5] addresses also the needs of future mobile and wireless systems as well as providing innovative solutions for

"fixed-mobile convergence". The main feature of an Ambient Network is an Ambient Control Space (ACS), which can be used to integrate and interoperate seamlessly any existing networks [18][5].

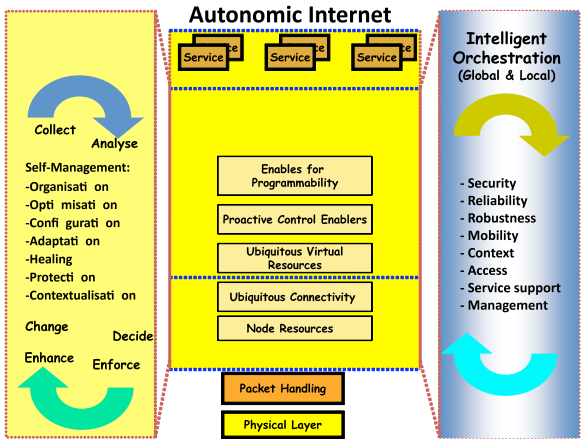


Figure 1 – Autonomic Internet

2.2 Requested Features: Self-management and Context-awareness

Currently, network management faces many challenges: complexity, data volume, data comprehension, changing rules, reactive monitoring, resource availability, and others. Self-management research started in 1989 [16] and it aims to automatically perform these tasks. The first main aim of Self-management systems is that they manage complexity, possess self-knowledge, continuously tune themselves, adapt to unpredictable conditions, prevent and recover from failures and provide a safe environment [19][20]. The second main aim [19] of Self-management systems is that they exhibit self-awareness properties, in particular self - optimisation; - organisation; -configuration; - adaptation/contextualisation; - healing; - protection.

On the other hand, the term "context aware" was first used in [21], which referred context as locations, identities of nearby people and objects and changes to those objects. In [22], the term of context was defined as locations, identities of the people around the user, the time of day, season, temperature, etc. Other refinements of the definition consider aspects of context [23] like: where you are, who you are, and what resources are nearby. In [24], context was defined to be the subset of physical and conceptual states of interest to a particular entity. One key definition for service context was espoused in [25] and can be adapted for this paper as service or network context "*Any information, obtained implicitly or explicitly, that can be used to characterise the situation of an entity involved in an application or service, especially the information that refers to the constantly changing environment of an entity. That information must be relevant to a service or application. An entity can be a physical object such as a person, a mobile host, a physical link, or a virtual object such as an application, process or computational object that is relevant to the application or service involved*". Network context characteristics include i. Network description (e.g. network identity, location, access-types, coverage); ii. Network resources in general (e.g. bandwidth, supported services,

available media ports for media conversion, available Quality of Service (QoS), security levels provisioned); iii. Flow context characteristics: flows are a possible embodiment of the interaction between the user and networks [26]. Context information that characterizes these flows may be used to optimise or enhance this interaction including: the state of the links and nodes that transport the flow, such as congestion level, latency/jitter/loss/error rate, media characteristics, reliability, security; the capabilities of the end-devices; the activities, intentions, preferences or identities of the users; or the nature and state of the end-applications that produce or consume the flow. Service context characteristics include: i. Service profiles; ii. Service resources, iii. Service qualities (e.g. QoS), iv. Service execution environment characteristics, v. Service configuration characteristics, vi. Service life cycle characteristics.

In a FIN service management system, a service directory that tells one where to access different types of services, should be capable of determining the availability of service context in a dynamically changing network. New tailor-made services can be provided to consumers by dynamically locating sources of different service context.

2.3. DHTs: Overview and Challenges

The solution presented in this paper utilises DHTs whose essential concepts using the Content Addressable Networks (CAN)-DHT as the underlying technique are presented in this Section. The fundamental concepts of different DHT implements are the same. For more detailed information, readers are referred to [10].

The essential element of a DHT is its keyspace. It is represented in a 2D array and is split between the DHT member nodes. Assume that, initially, there is only one node in the FIN. This node, say, node A, owns the entire DHT keyspace (Figure 2a). When the next node (i.e. node B) attempts to join the DHT, the new joining node randomly computes a point in the DHT keyspace. Node A will pass to node B the portion of keyspace that contains the point selected by node B; hence, the original keyspace is split (Figure 2b).

The keyspace passing process, essentially, means that: i) node A keeps a reference in its record that a portion of its keyspace (which is represented as a 2D array) is now assigned to node B; ii) node A notifies node B about the size of the original 2D array (i.e. the entire key space) and the section of the 2D array that node B has control over. The same applies for other new joining nodes i.e. node D and node C (Figure 2c and Figure 2d). The DHT keyspace ownership for 4 nodes is shown in Figure 2d.

Figure 2 - Example CAN-DHT keyspaces

To locate an item using the established DHT, the identifier of the item is hashed. This will map to a point in the DHT keyspace. The node that owns a keyspace portion that covers the point is responsible for holding that item (or holding the address of which the item is located). For example, node C maps an item name to a keyspace point (i.e. point xxxx in Figure 2). This point is covered by the keyspace portion currently owned by node A. However, node C does not see beyond its immediate DHT overlay neighbours (i.e. node B and D). But it knows that the point is "somewhere at the left-hand corner of the DHT keyspace" [11]. It therefore sends a request (i.e. a request for the item of interest) to its immediate overlay neighbour that is in the correct direction, in this case, node D. Node D also knows that the point is somewhere at the left-hand corner of the DHT keyspace; thus, it will pass on the request to node A. Node A can service the request. This decentralise overlay routing feature is an important element of DHTs. It is this feature together with the mapping facility that makes DHTs scalable [10].

We have previously addressed [18] some of the inefficiencies of existing DHTs for wireless and mobile networks. In [27], we have discussed how a large area of wireless network can be covered by multiple DHTs that are bootstrapped to support individual nodes' characteristics and requirements enabling nodes to avoid unnecessary negotiations during the DHT setup and maintenance process. One piece of work that is closely related to the OMB protocol is the RDFPeers [28] that focuses on developing a distributed repository p2p system. It stores triples at three locations in an addressable network, and provides guarantee to quires should the triple exists. Our work in contrast focuses on reducing the need of extended negotiation for setting up the directory service for which a number of challenging issues need to be addressed.

A service directory is capable of collecting information on service availability in FINs, and disseminating the collected information to consumers. Assuming FIN services are pre-defined, perhaps the simplest solution would be to pre-appoint a node (or a set of nodes) in the network to collect and disseminate network-wide service availability information (similar to the idea of having one receptionist which redirects queries in an office). However, such system would be centralised and static, which would not suit the decentralised and dynamic nature of FINs. A challenging approach leads to the investigation of techniques for continuously and dynamically selection of a set of nodes, to act as service directories. These nodes should be capable of keeping track of the real-time availability status of different service context in the network.

The concept of having nodes in the network to carry out management-oriented tasks is similar to the SuperPeer concept in P2P networking [29][30][31], that a subset of peers – that are considered as more capable of carrying certain tasks (such as those with more power, more processing power, etc.) – are dynamically selected or elected to take up certain managerial responsibilities in the network. A set of nodes is preferred because this arrangement is more distributed and more robust: should one node fail, others are there to "backup". SuperPeer election generally requires peers to compete against each other in order to determine the most capable peers in the network.

In rapidly changing network environments with relatively limited-bandwidth, dynamic negotiations for SuperPeer election, and subsequently re-election are not desirable due to the increased overhead they present. The negotiation overhead is dependent on the number of participating nodes (i.e. the more nodes negotiating at one time, the more overhead). It also depends on node mobility (i.e. re-election takes place when nodes move in and out). Thus, high node mobility would, potentially, result in frequent SuperPeer re-

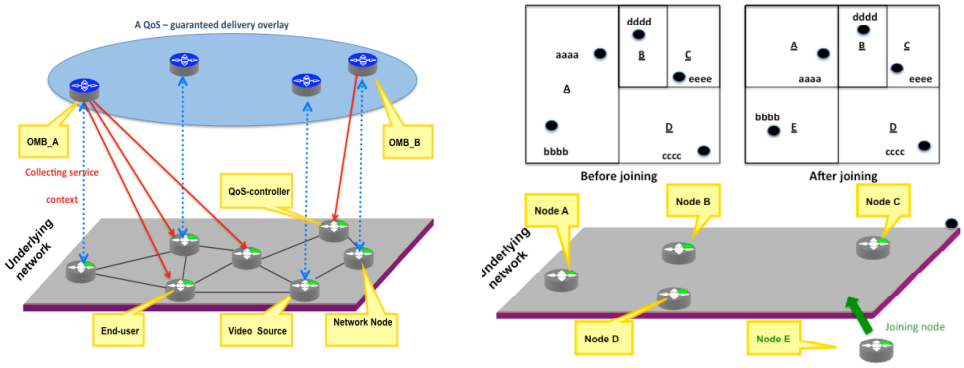
election. More importantly, SuperPeer re-election may result in a loop: negotiation → SuperPeer elected → new node joins → re-negotiation.

3 OVERLAY MANAGEMENT BACKBONE (OMB)

This section describes fundamental concepts of our solution, the OMB protocol.

3.1 System Overview

Our solution uses DHTs to determine which network nodes become service directories. These nodes are the OMB nodes. An OMB node is responsible for locating a particular type of service context, and disseminating the information to others.



(a) **(b)**

Figure 3 – (a) OMB example in a FIN, (b) Example DHT keyspace ownership

For example in Figure 3a, node OMB_A locates where the requested movie files is, and node OMB_B locates the QoS-controller(s) in the network. These nodes provide the information to support the implementation of the end-user's service. These OMB nodes, together with the node where the end-user resides, the node where the movie file is located, and the node where the QoS-controller is located, creates a service-specific overlay, i.e. a QoS-guaranteed movie-delivery-overlay, to serve the end-user's special needs. The challenge is to dynamically select a set of nodes to become the OMB nodes for different types of service context in the networks, and to re-select the OMB nodes if the selected nodes fail (i.e. when their power runs out, move out of range, etc.). The identities of these selected nodes must also be made known to all other nodes in the network, without additional overhead to suit the limited-bandwidth environment of FINs.

3.2. The OMB Protocol

Our approach towards selecting which node to become which service context locator, i.e. an OMB node that collects information of a particular type of service, is based on a mapping of service names using the DHT protocol. For example, if a DHT uses SHA-256, we will hash the name of the service using SHA-256, which will return a 256-bit keyspace identifier (Figure 4).

The DHT member node that owns the keyspace portion also contains the corresponding keyspace identifiers. This node will become the OMB node responsible for those particular types of service context. Figure 3b for example shows a simple example

DHT keyspace ownership in an FIN. Assuming there are initially four nodes in the network, each owns a portion of the DHT keyspace. The keyspace identifiers aaaa and bbbb are within the keyspace portion of node A, thus node A becomes the OMB node for SERVICE_01 and SERVICE_02. This means node A becomes responsible for collecting availability information of movies and QoS-controllers in the network. Similarly, node D owns point cccc, so node D is the OMB node for SERVICE_03, and so on. If node A wants to know which node is the OMB node for SERVICE_03, it maps the name of SERVICE_03 to the corresponding DHT keyspace identifier, i.e. cccc (Figure 3b). Then, by DHT overlay routing, it will (indirectly) route its request to node D, which is the OMB node for SERVICE_03. The advantage of this arrangement is obvious – without any form of (real-time) negotiation, we have achieved two objectives: to dynamically assign service context collection tasks to nodes, and to dynamically disseminate the information on "which node knows what" to others.

SERVICE_01 (locate movies):
SHA-256(SERVICE_01) → aaaa
SERVICE_02 (locate QoS-controllers):
SHA-256(SERVICE_02) → bbbb
SERVICE_03 (…):
SHA-256(SERVICE_03) → cccc

Figure 4 - Mappings between service names and DHT keyspace identifiers

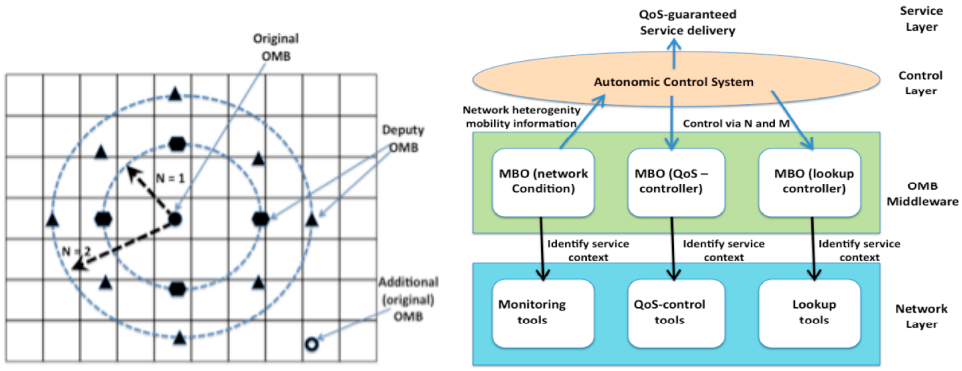
3.3. Addressing Node Mobility and Heterogeneity

To illustrate how our system addresses node mobility, assuming a node now joins the DHT (i.e. node E in Figure 3b). When it joins the DHT, it obtains a portion of the DHT keyspace from existing members of the DHT, using the standard/revised DHT protocols (explained in section 2.3). Assuming that it has obtained its keyspace portion from node A, it now owns the keyspace identifier bbbb (Figure 3b). Thus, we say that, node A has "transferred" its responsibility (of SERVICE_02) to node E. This is an importantly feature in our system – it enables balanced loading: the DHT joining process requires random keyspace partitioning; thus, statistically, keyspace ownership is balanced between all DHT nodes. Hence, the OMB load across all nodes in the DHT is also balanced. This enables the OMB protocol to be completely decentralised i.e. no one node is permanently responsible for collecting one type of service context. Also, when new nodes join, tasks are automatically transferred to the new joining node, which enables balanced loading in the network. Also note that, no other nodes in the network needed to be explicitly notified of the recent transferral of responsibility between node A and node E (in contrast to SuperPeer approaches where the identities of new SuperPeers must be explicitly published to peers in the network).

Now, suppose an end-user that resides on node C wants to know which node is responsible for SERVICE_02. Originally, before node E joins the network, node A is responsible for SERVICE_02. When node C maps the service name to keyspace identifiers (i.e. bbbb), it will route its request through DHT overlay routing. For example, it will send its request to node D. Because node D is an immediate overlay neighbour to node E, according to the DHT overlay routing, it knows that node E is likely to be responsible for SERVICE_02 [11]. Hence, node D will pass on the request to node E, which will service node C's request accordingly.

Note that, so far, the discussion assumes that all nodes are capable of carrying out the service context collection tasks that they are assigned to be responsible for. However, in reality, nodes are heterogeneous, they have different features and capabilities. Thus, not all nodes are capable of doing so.

The OMB protocol addresses this heterogeneity issue in the network by requiring the overlay neighbouring nodes to an OMB node (i.e. the original OMB node) to carry out the same service context collection task. These nodes are known as the deputy OMB nodes.



(a) **(b)**

Figure 5 – (a) Example original, deputy, and additional OMB nodes in the DHT; (b) FIN control space and network services

For example in Figure 5a, suppose the original OMB node is expected to be responsible for a particular type of service context. The nodes that own keyspace portions immediate neighbouring to the keyspace portion of the original OMB node (i.e. the ones marked with a hexagon) should carry out the same service context collection task as the original OMB node. These deputy nodes are useful because they act as the backups when: 1) the original OMB node is not capable of carrying out its task (e.g. it has little processing power); 2) the original OMB node is no longer capable of carrying out its task (e.g. it runs out of power); 3) the network is heterogeneous i.e. not all deputy nodes are capable of carrying out their tasks; thus a set of deputy nodes are selected so that they can "backup" the original OMB node.

If any of the OMB nodes become incapable, but they are queried by an end-user, the node will query its immediate overlay neighbours (which host the deputy OMB nodes) for the service context of interest. The incapable node also updates its cache with the service context obtained from its neighbours, and responds to the end-user as if it was the source of the result. Optionally, the incapable node or its deputy may also inform the end-user or an autonomic manager through its response that it is in fact an incapable node, and refer the end-user to one of its capable neighbours so that next time such entity can avoid contacting the incapable node.

Generally speaking, the larger the network, or the more heterogeneous the network, more deputy nodes would be needed. Also, if routing locality were optimised, that the overlay neighbourhood maps to physical network neighbourhood, the deputy nodes would tend to be physically near to each other, which would result in uneven load balancing. Thus, we introduce two factors, known as the neighbourhood scale N, and the network scale M, to control the scale of deployment of the OMB protocol.

The factor N determines the radius of deployment of the OMB protocol. For example, in Figure 5a, the radius of deployment is set to 1 and 2 respectively. If N=1, the immediate overlay neighbours to the original OMB node become the deputy OMB nodes (i.e. marked with a hexagon). If N=2, the immediate overlay neighbours to those deputy OMB nodes also become the deputy OMB nodes (i.e. marked with a triangle).

We have discussed that if routing locality is optimised, the deputy nodes tend to locate in nearby neighbourhood, which may result in uneven loading in the network. The M factor is designed to optimise the protocol when uneven loading happens due to optimised routing locality. M=1 when there is only one original OMB node (i.e. the one marked with a dark circle); M=2 when there is an additional (original) OMB node (i.e. the one marked with a hollow circle), and so on. The additional (original) OMB node is determined using a similar approach to the mapping logic as discussed in an earlier section, but by multiple hashing (Figure 6). Multiple hashing means:

1) Hash the service name (i.e. SERVICE_01) with, say, SHA-256, which gives a 256-bit keyspace identifier (#1). The node, which owns a keyspace portion of the DHT that covers this point, is the original OMB node.

2) If M=2, hash the result of step 1 (which was a 256-bit keyspace identifier) using SHA-256, which gives a new 256-bit keyspace identifier (#2). This new identifier refers to another keyspace point in the DHT keyspace (Figure 5a). The node that owns a keyspace portion of the DHT that covers this new point is the additional (original) OMB node.

When M=1:
SHA-256(SERVICE_01) → keyspace point of the original OMB node (#1)
When M=2:
SHA-256(SERVICE_01) → keyspace point of the original OMB node (#1)
SHA-256(SHA-256(SERVICE_01)) → keyspace point of the additional (original) OMB node (#2)

Figure 6- Multiple hashing

N and M may be used together to control the scale of deployment of the OMB protocol. The values for N and M are determined by the size of the network, and also the level of heterogeneity and mobility of the network (i.e. the failure rate). The level of heterogeneity and mobility are specific to the current conditions of the network, which means they are some form of service context. Thus, they are provided by some network condition monitoring services that are identified and located by OMB nodes. This means that, when a node joins the network, it queries the OMB node that is responsible for locating the network monitoring tools, and obtains the necessary information on N and M.

Figure 5b presents the design of the FIN control space, which utilises available service context in the networks to implement tailor-made end-user services. The OMB node, that is responsible for locating network monitoring tools, provides access to network heterogeneity and mobility information to the FIN (the information on where the monitoring tools are currently located in the network). In turn, the FIN control space uses this information to control the size of deployment of OMB nodes in the network (via N and M).

4. Evaluation

This section evaluates the OMB protocol in terms of scalability, efficiency and robustness. Key OMB features are analysed as compared with other approaches that

generally require dynamic negotiation (i.e. SuperPeer [10], RDFPeers [28]). The evaluation software was written in Java and run on a Linux box with an Intel Core2 CPU (1.83GHz) and 1G RAM. Our program essentially contains a 2D array representing a 2D CAN DHT keyspace, which are recursively allocated to joining peers.

4.1. Scalability Advantage of the OMB Protocol

We first compare how many messages would need to be exchanged in order identify the OMB node responsible for collecting information on the availability of a specific type of service context. In the SuperPeer approach, this would be to determine the responsible SuperPeer. We assume that the most suitable SuperPeer is the one with the most processing power. We used broadcast as the mechanism for negotiation in the SuperPeer approach [31][32].

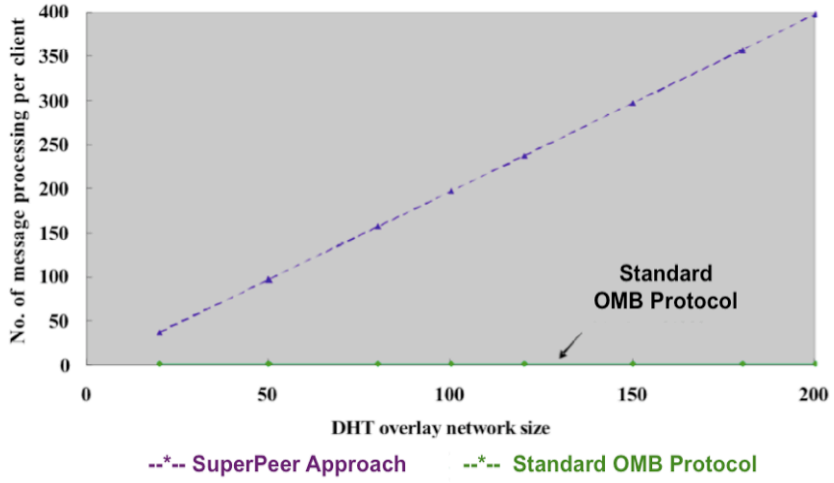


Figure 7- Scalability evaluation

From the results shown in Figure 7 the SuperPeer approach does not scale due to the number of messages needed to be exchange. Negotiation involves the use of a shared medium (i.e. broadcast), and the negotiation traffic would depend on the size of the entire network. The larger the network, the heavier the overhead. Note that if a new node joins, the same process would have to be repeated. On the other hand, the OMB protocol is much more scalable, dis-regard of the network size. This is achieved through the utilisation of the underlying DHT in the network. As for the storage requirements, a node needs only to maintain a keyspace map, and a list of its immediate physical neighbours. The latter is limited to the ad-hoc range of the device (e.g. a few nodes at most); as for the keyspace maintenance, the node only needs to maintain a list of its virtual neighbours. The degree of this scale of maintenance depends on how evenly fragmented the keyspace is: statistically, load balancing can be assumed in DHT (i.e. every node has equal chance to obtain any portion of the keyspace). Thus, we can safely assume that the keyspace would be divided evenly in the long term; hence, the number of virtual neighbours needed to be maintained by one node is limited.

4.2. Efficiency Advantage

Search efficiency is optimised in DHTs, in the sense that overlay hop count is limited. Standard DHTs do not optimise routing locality meaning that the overlay routing does not map with the underlying physical routing. Our previous work has designed a protocol that optimises DHT routing locality [18][27]. By utilising the underlying DHT in the OMB protocol, search efficiency to locate a particular OMB node is enhanced, too.

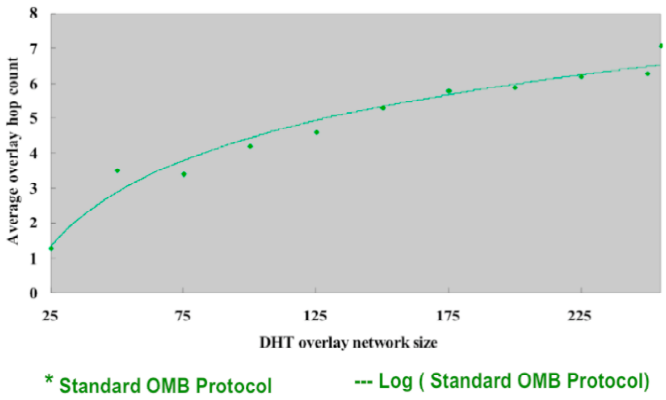


Figure 8 - Efficiency evaluation

The efficiency of the OMB protocol (N=0, hence only one OMB node per service context type) is evaluated by determining the average number of overlay hops needed by a randomly selected end-user node (which is also an overlay member) to reach a particular type of OMB node in the overlay, in networks of different sizes. We assume all nodes in the network are 100% capable of hosting any type of OMB node i.e. node failure rate=0% (see later for evaluation on robustness of OMB node). By setting only one OMB node per type in the network, the average hop count is most likely to be higher.

Figure 8 shows how the overlay hop count varies in overlay networks of different sizes. The OMB protocol is efficient because there are only a few overlay hops in between an OMB node and the end-user (~6 overlay hops for a 225-node network). We have discussed that our protocol is applicable when routing locality is optimised, by having additional (original) OMB nodes in the network. The slope of the curve gradually decreases as the size of network increases. Thus, the effect of a network with an increasing size has limited impact on the search efficiency of the OMB protocol.

4.3 Robustness Evaluation

Robustness (when a failed node is backup by others) evaluation is achieved by adjusting the values of N and M to maintain a certain number of active OMB nodes in the network. We set M=1, so that there is only one original OMB in the network. This OMB node may fail, so we adjust the value of N to increase robustness, which increases the number of deputy OMB nodes in the network. We have discussed that the value of N (and also M) depends on the failure rate, which in turn, depends on the network heterogeneity and mobility. Figure 9 shows that, in order to maintain a certain number of OMB nodes in the network (i.e. 10% of the total nodes in the network), N can be set to different values in

order to accommodate different failure rates. Therefore, the provisioning of N (and also M) in the OMB protocol provides the necessary facility to adjust the level of robustness.

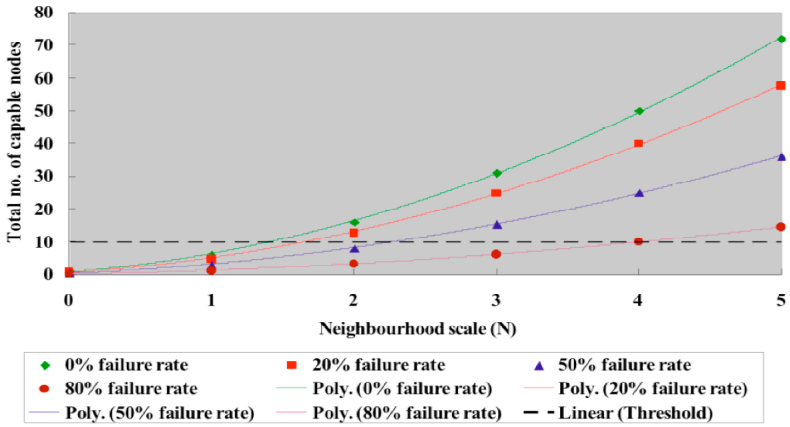


Figure 9- Robustness evaluation

The OMB protocol is designed for scalability, efficiency and robustness. The identities of the OMB nodes are determined and disseminated dynamically without any form of dynamic negotiation: the underlying DHT enables any users to efficiently locate which node is responsible for which type of service context. Also, consumers or autonomic managers may access service availability information via the original, deputy, and additional OMB nodes in the network, which means single point of failure is avoided. The end-user's request routes its way through the DHT to the (original) OMB node via DHT overlay routing. Thus, the request is intercepted and serviced by the first intercepting OMB node. This means that, the system is decentralised. Also, efficiency is enhanced because an end-user's request may be serviced by the nearest OMB node.

The load-balancing feature of the OMB protocol makes OMB more scalable and efficient: no one node in the network is responsible for collecting one type of service context. In fact, the entire system is distributed, that when new nodes join, responsibilities may be transferred to these nodes. Similarly, if an OMB node becomes an incapable node, its assigned responsibilities are automatically transferred to the backup nodes. All these are made possible without any form of dynamic negotiation between nodes.

5. Conclusions and Further Work

The key concept of Future Internet Network (FIN) is to develop a common control space, which enables service sharing across heterogeneous end-user devices through wired and wireless channels. In order to utilise service deployment in FINs, there is a need for a system that can dynamically determine the availability of different types of service context in the network. This implies that, some service directories, which indicate which service is available where, are needed in FINs. This service directory should have a self-organising mechanism that would efficiently and automatically determine a set of nodes in the network to carry out dedicated service context tasks. Existing similar approaches, such as the SuperPeer approaches, use real-time negotiations to assign tasks to the most suitable nodes in the network. However, due to node mobility, the negotiation process may result in a loop. Also, the higher the level of dynamicity in the network, the

more frequent the negotiation process may take place. Since negotiations are usually conducted through broadcast or multicast, frequent negotiation means frequent broadcast or multicast. Frequent use of a shared medium, particularly in bandwidth-limited environment, should be avoided.

This paper presented the OMB protocol, which enables automatic and self-organising service context task assignations in FINs. No dynamic negotiation is needed between peers even when the network has a high mobility. The OMB protocol features load balancing, by evenly distributing service context tasks to peers in the network through the utilisation of the underlying DHT in FINs. Our solution is completely decentralised and self-organised, and it is designed to be efficient and scalable. Enhanced robustness of the protocol is achieved by adjusting the values of N and M respectively, which controls the number of active OMB nodes in the network.

Future work includes: i. use of OMBs for different network management applications; ii. applying and using the OMB protocol for management of virtual networks; iii. quantification of the level of mobility/or network topology change; iii. assessing the impact of energy usage in the OMB selection; iv. synchronisation of deputies with the original nodes; v. defining a scalable and efficient mechanism for all nodes within the FIN control space to participate in request for tailored consumers' service proposals, assuming participant nodes have their own interests of association, service provisioning constraints, level of commitment and their own business objectives. The goal for this critical mechanism is to quickly converge towards an acceptable solution for the provisioning of the target service; vi. An Autonomic Network Programming Interface (ANPI) dedicated to autonomic services deployment in the networks is currently under development. It will be based on the OMB protocol to efficiently access to large decentralized service repositories.

Acknowledgments

Part of this work was undertaken in the context of the Autonomic Internet project [17] and Ambient Networks project [5], which are partially financed by the EU.

References

[1] Bassi, A., Denazis, S., Galis, A., Fahy, C., Serrano, M., Serrat, J., -"Autonomic Internet: A Perspective for Future Internet Services Based on Autonomic Principles" - IEEE 3rd Intl. Week on Management of Networks and Services Manweek 2007 / MACE 2007 2nd IEEE International Workshop on Modelling Autonomic Communications Environments, 29 October – 2 November, San José, California, USA

[2] Future Internet Design (FIND) Program - http://www.nets-find.net/

[3] Global Environment for Network Innovation (GENI) Program - http://www.geni.net/

[4] Future Internet Assembly (FIA)/ FIRE program –http://www.fi-bled.eu/ http://cordis.europa.eu/fp7/ict/fire/home_en.html

[5] Ambient Networks (ANs) Project, http://www.ambient-networks.org

[6] Mobile Ad-hoc NETworks (MANETs), http://www.ietf.org/html.charters/manet-charter.html

[7] Gnutella development forum, the gnutella v0.6 protocol. Available: http://groups.yahoo.com/group/the gdf/files/

[8] Bittorrent. Available: http://bitconjurer.org/BitTorrent/

[9] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. (1999) Freenet: A distributed anonymous information storage and retrieval system. Freenet White Paper.

[10] E. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, “A Survey and Comparison of Peer-to-Peer Overlay Network Schemes”, in IEEE Communications Survey and Tutorial, March 2004.
[11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, “A Scalable Content-Addressable Network”, in ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), San Diego, CA, USA, Aug 2001, pp. 161-172.
[12] Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", in the IFIP/ACM Middleware, Heidelberg, Germany, Nov 2001
[13] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", in the ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), San Diego, USA, Aug 2001
[14] Zach, M., Parker, D., Fahy, C., Carroll, R., Lehtihet, E., Georgalas, N., Marin, R., Serrat, J., Nielsen; Towards a framework for network management applications based on peer-to-peer paradigms; NOMS 2006, Vancouver, Canada
[15] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. and Orchard, D., "Web Services Architecture", W3C Working Group Note, W3C, Feb 2004
[16] Fehskens, A., “Monitoring Systems” 1st IFIP Integrated Network Management Symposium, Boston, May 1989
[17] Autonomic Internet Project –http:// www.ist-autoi.eu
[18] L. Cheng, R. Ocampo, K. Jean, A. Galis, C. Simon, R. Szabo, P. Kersch, R. Giaffreda, ”Towards Distributed Hash Tables (De)Composition in Ambient Networks”, 17th IFIP/IEEE Distributed Systems: Operations and Management (DSOM), Dublin, Ireland, Oct 2006
[19] J. Strassner, N. Agoulmine, and E. Lehtihet, FOCALE—A Novel Autonomic Computing Architecture, LAACS, 2006
[20] Curran, K., Mulvenna, M., Galis, A., Nugent, C. –”Challenges and Research Directions in Autonomic Communications”- International Journal of Internet Protocol Technology (IJIPT) - Vol. 2 No. 1; Jan 2007; SSN (Online): 1743-8217- ISSN (Print): 1743-8209
[21] Schilit, B., Theimer, M.; "Disseaminating Active Map Information to Mobile Hosts”, in IEEE Network 8(5), pp 22-32, IEEE 1994
[22] Brown, M., "Supporting User Mobility," International Federation for Information Processing 1996
[23] Schilit, B., Adams, N. and Want, R., "Context-Aware Computing Applications," in the IEEE Computer Society Workshop on Mobile Computing Systems and Applications, pp. 85-90, Santa Cruz, CA, IEEE 1994
[24] Pascoe, J., "Adding Generic Contextual Capabilities to Wearable Computers," in the 2nd International Symposium on Wearable Computers, pp 92-99, 1998
[25] Dey A., Salber, D., Abowd, G., "The Context Toolkit: Aiding the Development of Context-Enabled Applications," in the 1999 ACM Conference on Human Factors in Computer Systems (CHI'99), pp. 434-441, PA, ACM Press, May 1999
[26] , R., Galis, A., De Meer, H., Todd, C. “Flow Context Tags: Concepts and Applications,” NetCon05 Conference; Lannion, France 14-18 November 2005
[27] L. Cheng, K. Jean, R. Ocampo, A. Galis, P. Kersch, R. Szabo, “Secure Bootstrapping of Distributed Hash Tables in Dynamic Wireless Networks”, in the IEEE International Conference on Communications (ICC), Glasgow, UK, Jun 2007.
[28] M. Cai, M. Frank, “RDFPeers: A Scalable Distributed RDF Repository based on a Structured Peer-to-Peer Network”, 13th Intl. Conf. on World Wide Web, NY, USA, May 17-20, 2004
[29] M. Kleis, E. Lua, X. Zhou, “Hierarchical Peer-to-Peer Networks using Lightweight SuperPeer Topologies“, in the 10th IEEE Symposium Computers and Communications (ISCC), Cartagena, Spain, Jun 2005, pp. 143-148
[30] G. Jesi, A. Montresor, O. Babaoglu, “Proximity-Aware SuperPeer Overlay Topologies”, in the 2nd IEEE International Workshop on Self-Managed Networks, Systems & Services (SelfMan), Dublin, Ireland, Jun 2006

[31] A. Mizrak, Y. Cheung, V. Kumar, S. Savage, “Structured SuperPeers: Leveraging Heterogeneity to Provide Constant-Time Lookup”, in the IEEE Workshop on Internet Applications (WIAPP), San Jose, USA, Jun 2003
[32] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, D. Yao, “Optimal Peer Selection for P2P Downloading and Streaming“, in the IEEE Infocom, Miami, FL, March 2005