Replication of Routing Tables for Mobility Management in Ad Hoc Networks

Jose Costa-Requena
Nokia Mobile Phones
Nokia Corporation
Helsinki, Finland
Jose.Costa-Requena@nokia.com

Nicklas Beijar, Raimo Kantola
Networking Laboratory
Helsinki University of Technology
Helsinki, Finland
Nicklas.Beijar@hut.fi, Raimo.Kantola@hut.fi

Abstract— **Ad Hoc networking is a rather old technology that is gaining a lot of momentum in the research community. It is still an area under development and there are several proposals for the technology regarding routing protocols, addressing, interoperability, etc. Because of the extreme conditions where Ad Hoc networks should operate, it is envisioned that in medium to large scale networks, a hybrid proactive and reactive routing mechanism should be used. Therefore, considering the proactive routing protocols as the most suitable protocol for certain topologies in Ad Hoc networks, this paper proposes a specific layered structure to enable both technologies in a seamless manner. This approach supports both reactive (e.g. AODV, DSR) and proactive link state (e.g. OLSP) protocols. The proposed layered structure divides the routing problem into different parts, where one module performs the routing algorithm, another module takes care of replicating and synchronizing the routing tables, and a third one evaluates the routing information for triggering hand-over signals to the upper layers. The overall responsibility for selecting the appropriate routing schema and for initiating the roaming process resides on a proposed Context Sensitive Roaming Layer. This paper describes the interface and the functions implemented by the Roaming Layer. The paper also analyses replication using the Server Cache Synchronization Protocol (SCSP). Finally, the last part of the paper describes the demand for Ad Hoc networks, which is driven by applications supported with an appropriate billing or bonus mechanism.**

Keywords— ad hoc; link state; routing; service discovery; SCSP

I. Introduction

Ad Hoc networks are considered as the technical evolution from 4G towards Private Area Networks. In the Ad Hoc research area, multiple alternative routing protocols have been developed. These are either reactive (on demand routing protocols, e.g. AODV [1], DSR [2], TORA [3]), proactive (link state routing protocols, e.g., OLSR [4], FSR [5]), or combinations of these (e.g. ZRP [6]). Due to the uncertainty in selecting suitable routing protocols for the Ad Hoc infrastructure, we propose to split the problem into different building blocks. The selected philosophy for this approach is based on a hybrid solution that includes both reactive and proactive protocols.

The question is why to use a proactive protocol that stores and replicates the routing information among nodes within the Ad Hoc infrastructure. Reactive protocols, which do not require nodes to store link state information, are very suitable for Ad Hoc networks, where the topology is extremely dynamic. Moreover, the battery and bandwidth consumption required for keeping the routing tables up to date in the link state routing protocols is a significant drawback. The purpose of our study is to present the benefits of the link state protocols for certain aspects of the Ad Hoc infrastructure.

The aim of this paper is to define a modular design to support a large-scale link state based solution. The routing problem is divided into different parts: one consists of the routing algorithm itself, another contains a mechanism for replicating and synchronizing the routing tables, and a third one analyzes the received routing information for communication to upper layers (a proposed Context Sensitive Roaming Layer) in case it is necessary to trigger a handover to a new network interface. The proposed architecture considers a concept where some of the nodes with enough battery and resources (smart nodes) support the rest of the nodes with poor resources (dummy nodes). The capable or smart nodes run a link state protocol among them, while keeping a reactive behavior with the dummy nodes in the community.

The routing topology and information about the network elements are essential for the networks to function properly. In addition, information for locating the services available in the network is also crucial. Important elements provisioning services include DHCP [7] servers, DNS [8] servers, NAT [9], SIP registrars [10], etc. Service location is very challenging in Ad Hoc networks since nodes are frequently moving and changing their location or point of attachment. There are different proposals that advocate providing a service discovery mechanism at the session or application layer like JINI [11], UPnP [12], Salutation [13], OSGi [14], SLP [15], among others. On the other hand, there are alternatives that implement the service discovery mechanism at the link layer such as Bluetooth and SDP [16]. Based on previous studies about location services [17] we propose an alternative but more generic mechanism for network service location that can be used for any layer above the network layer.

Ad Hoc networks have special requirements in terms of update frequency, data size, and range of information involved. For providing some measurements as initial basis for evaluation, a Server Cache Synchronization Protocol (SCSP) [18] prototype is considered as the replication mechanism to update the terminal information among nodes. It does not preclude that SCSP is the optimal approach but SCSP is selected among others alternatives for analyzing its scalability in providing a synchronization and replication mechanism for routing information. The objective is that the replication module will provide the required API to the routing module. Therefore, an approach similar to SCSP optimized for Ad Hoc environments can be considered as a generic replication mechanism that works independently of the data content. Finally, to corroborate the SCSP performance in terms of scalability and flexibility some results obtained with a prototype are presented.

Section II describes the proposed architecture, the modular design and the API. Section III presents requirements for Ad Hoc networks and relevant features in terms of addressing, routing and interoperability. Section IV analyses different solutions that could be used for implementing the synchronization/replication module as part of the generic routing infrastructure. After comparing various alternatives, the Server Cache Synchronization Protocol (SCSP) is evaluated. The results help determining the appropriate algorithm for implementing the synchronization/replication mechanism for link state-based protocols in Ad Hoc networks. Section V describes some guidelines for implementing a suitable framework for Ad Hoc environments. Finally, conclusions and results are discussed.

II. ARCHITECTURE

Appropriate routing protocols for Ad Hoc networks are still under development. It remains uncertain, whether a distance based, a link state or a hybrid solution is optimal. Meanwhile, the best practice is to define a routing architecture that inter-operates with both distance vector and link state protocols. In the proposed architecture, the nodes are divided into smart and dummy nodes. The nodes that do not have sufficient resources for supporting the proactive link state protocol (dummy nodes) implement a reactive distance vector protocol. On the other hand, nodes with enough resources (smart nodes) implement the link state protocol and help the nodes with limited resources to accelerate the distance vector-based route discovery. In this approach, the nodes with enough capabilities for running link state protocols can assist the rest of virtual Ad Hoc community. The reason for proposing the link state mechanism is the need to keep some persistent data in the nodes to provide interoperability and mobility management, especially when the Ad Hoc network is part of another infrastructure (3G, 4G).

The classification into smart and dummy nodes corresponds to two main classes of devices found in large-scale Ad Hoc networks. Some nodes are small with a minimum of resources and with high mobility. Examples of such devices are mobile phones, PDAs and other equipment with Ad Hoc connectivity. On the other hand, Ad Hoc networks include relatively static nodes with more resources, for example servers or access points. Therefore, the separation of roles can improve performance and efficiency.

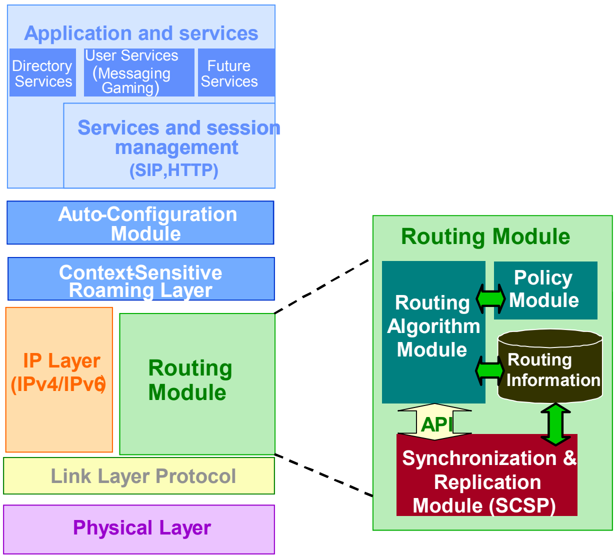


Figure 1. Routing module within the Ad Hoc node.

The routing problem is split into separate modules for the routing algorithm and the replication protocol. An API between the routing layer and the replication layer defines the communication between the modules. In addition to the routing information, the API provides information about neighboring nodes and link reliability. This allows the policy module to determine the frequency of updates and the information required for node discovery. It provides the possibility to decrease the update frequency as well as adapting the rate of refreshments and keep-alive messages depending on the environmental conditions. Figure 1, shows the routing module and its components embedded in the Ad Hoc node architecture.

Ad Hoc networks have a highly dynamic topology. Therefore, the API has to provide an Attach-Detach procedure. The Detach procedure allows the nodes to inform that they cannot continue contributing to maintain the link state information. Any of the smart nodes can become a dummy node in case that it runs out of battery or resources. Thus, a procedure is performed among the smart nodes to advertise either that they are able to participate as link state nodes (Attach) or that they are becoming dummy nodes (Detach).

By dividing the problem into separate modules, it is possible to utilize several routing schemes in the same nodes, and change the modules to maintain the optimal operation. The Context Sensitive Roaming Layer allows roaming between different networks according to the location or context.

III. REQUIREMENTS FOR AD HOC NETWORKS

This section identifies the different levels of requirements needed to provide a robust Ad Hoc infrastructure. These requirements are a set of the conditions that any routing mechanism should be compliant with to perform optimally.

A. Routing

The routing algorithm has to be flexible and support different technologies. Routing tables contain different address information such as MAC, IP or logical addresses [19] that are used by the link state protocol. Furthermore, the tables are extensible for adding new attributes if required by either the physical or the network layer. Thus, our proposal based on a modular structure in the implementation allows changing the routing algorithm without affecting the rest of the modules.

To provide modularity, the routing mechanism as a whole should be split into different layers. The routing mechanism should support backward compatibility with existing and forthcoming routing proposals. Furthermore, any proposal should make use of the existing protocols whenever possible and impose minimum changes or extensions on the existing ones. Any new approach has to minimize infrastructure installation costs and complexity.

The routing mechanism should provide the means for an Ad Hoc node to move between heterogeneous networks in a wide area. It could be used to attach the device anywhere and to enable the Ad Hoc nodes to work via any available access technology (802.11b WLAN, Bluetooth, etc.)

The routing module has to be capable of informing about the network entities available in the Ad Hoc network and providing a service discovery enabler to the Ad Hoc nodes. The information and capabilities of the network elements (DNS, DHCP, SLP, SIP Registrar, etc) are communicated as part of the link state information. The link information provides the awareness of the connectivity within the Ad Hoc networks. Additionally the routing module maintains information about the network capabilities. The information is kept as part of the routing data in a transparent manner to the application and the user.

When a session handover is performed at the network layer, the status is transparent to upper layers. A change of topological location will not have any impact on layers above the network layer other than a temporary loss of performance. The proposed routing mechanism should be able to handle the handover process during those changes, and perform the service discovery and node configuration transparently to the upper layers. For this functionality, requirements are minimum delays and minimum signaling load leading to minimum bandwidth consumption for non-user traffic.

A unified routing and network capabilities discovery mechanism avoids situations where distinct configuration mechanisms are unable to inter-operate with each other. Therefore, the addresses of servers that are most relevant for optimal performance of the infrastructure are part of the routing information that the nodes exchange.

B. Addressing

Addressing depends on the technology and requires self-descriptive formats and a flexible structure. This structure should be able to transport the MAC, IP, Bluetooth, and logical addresses in the link layer information. The replication/synchronization mechanism does not consider the content of the routing tables. It just considers the content of the cache entries, and creates the Hash or Cache Key and a time stamp to check the newest entries from the other nodes in the Ad Hoc group.

The link layer uses its own addressing mechanisms that are kept in the lower layers. Above the link layer, IP functionality is required to provide a common framework for existing applications and services.

The network layer keeps the routing and network feature information. It hides the underlying mechanism providing a transport platform while in the upper layer it will be seen as IP transport. Therefore, on top of this layer any IP-based service can be implemented. The main advantage of this approach is that terminals continue using IP based addressing and naming, and they can use any IP-based services locally or on top of any other IP network (Internet, 3G [20]).

C. Node Discovery

Nodes should have an automatic procedure to let the rest of nodes in the Ad Hoc network know about their presence in the group. The proposed routing module provides a node discovery mechanism. IPv6 [21] provides a node discovery mechanism that detects new nodes and resolves IP address collision problems. Nevertheless, the node discovery proposed for the routing module offers additional information about the network elements. This mechanism is part of the link state discovery performed among the nodes that are able to keep the link state information in the network. With this approach the nodes are able to exchange routing information and network capabilities to improve the routing process.

D. Network Service Discovery

When IP is used for communication the user is reached through the assigned IP address. The conflict appears when the Ad Hoc network due to its nomadic nature creates its own address space, and it has to be available to other networks. As a result, when the Ad Hoc network requires access to external networks, a Network Address Translator (NAT) is required. One of the nodes within the Ad Hoc network takes the role of NAT and replicates the information among all the nodes using the proactive protocol to let the other nodes be aware of the access point towards external networks.

Furthermore, if nodes in the Ad Hoc network request VoIP services, a Session Initiation Protocol (SIP) proxy or registrar will be required. Thus, one of the nodes within the Ad Hoc network can implement the functionality of the required server. In that sense, a completely self-sufficient infrastructure that provides IP accessibility and VoIP services can be established.

Similarly, in case of locating any other service within the Ad Hoc infrastructure, the nodes can share the location of other relevant network entities such as DHCP or SLP server. Figure 2 shows the required discovery mechanism when a new node enters in the Ad Hoc infrastructure.

The new (smart) node sends a network discovery broadcast to learn about the network capabilities and topology. Only the dummy nodes respond to the query but the answer contains the address of the existing smart nodes in the network.

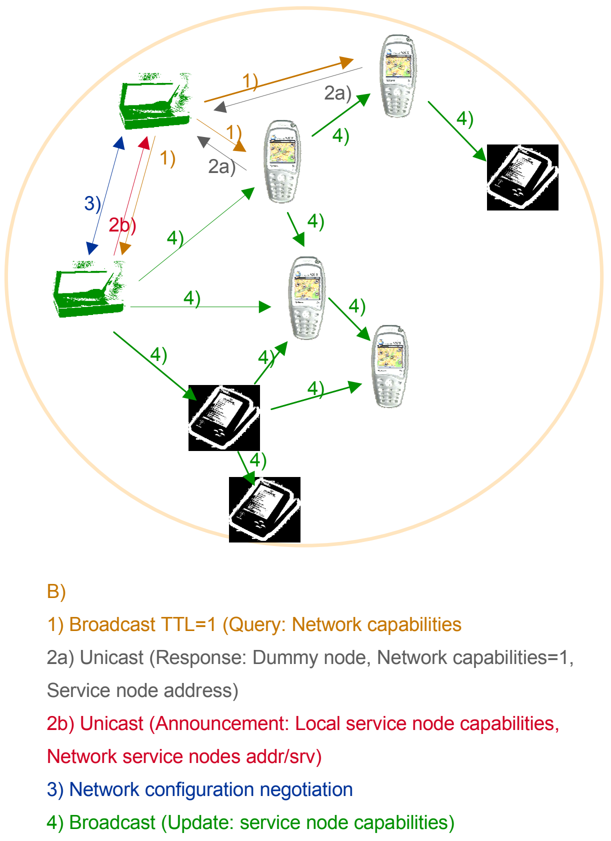

1)
2a)
1)
2a)
1)
3)
2b)
4)
B)
1) Broadcast TTL=1 (Query: Network capabilities
2a) Unicast (Response: Dummy node, Network capabilities=1, Service node address)
2b) Unicast (Announcement: Local service node capabilities, Network service nodes addr/srv)
3) Network configuration negotiation
4) Broadcast (Update: service node capabilities)

Figure 2. Network services and node discovery procedure.

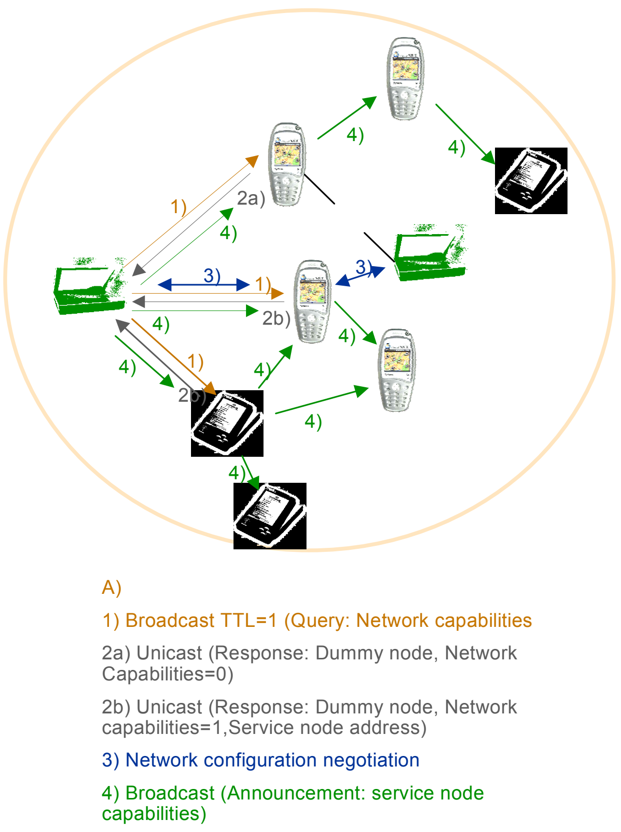

1)
2a)
3)
1)
2b)
3)
1)
2b)
4)
A)
1) Broadcast TTL=1 (Query: Network capabilities
2a) Unicast (Response: Dummy node, Network Capabilities=0)
2b) Unicast (Response: Dummy node, Network capabilities=1,Service node address)
3) Network configuration negotiation
4) Broadcast (Announcement: service node capabilities)

Figure 3. Network services and node discovery procedure.

When the querying node obtains the address of a smart node, the service provisioning is negotiated between the smart nodes. Afterwards, the new network capabilities are broadcast to all the dummy nodes to distribute the server address that is provisioning the new service.

In Figure 3, the new smart node obtains the address of a smart node, which is not directly reachable. The communication between the smart nodes is relayed by the intermediate dummy node. The service negotiation is done between the smart nodes and the network capabilities are broadcast over the network.

Figure 4, shows how the user application triggers the roaming of the terminal into Ad Hoc networks.

IV. Replication

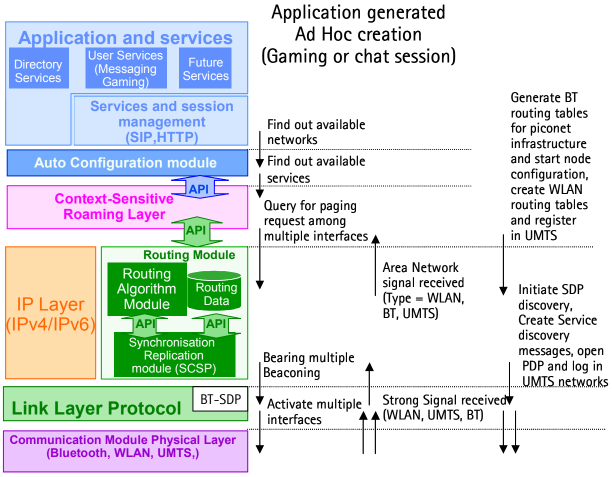
This section analyses the replication/synchronization requirements necessary for supporting proactive protocols in Ad Hoc networks. Some measurements from a SCSP prototype are presented. In our proposal, the routing and network capabilities information are considered as cache data kept in the mobile nodes.

In Ad Hoc networks not all the nodes are able to keep a big amount of data up to date. For that reason, the nodes are divided into two classes: dummy (lower level) nodes and smart (higher level) nodes. The smart nodes have enough power and resources for replicating caches. Therefore, the information is quickly accessible from different locations in the network and the load of the route discovery is reduced depending on the amount of smart nodes in the Ad Hoc network. The smart nodes implement the replication mechanism that allows developing scalable schemes to efficiently use the network topology.

One of the main disadvantages of proactive protocols is the inconvenience of storing the link state information in dynamic networks where the routes become obsolete very rapidly. Therefore, a suitable synchronization/replication algorithm should be tailored for Ad Hoc networks. The main requirement for the synchronization module is that the selected protocol should be independent of the underlying technology. Thus, the physical layer can be changed while keeping the same API to the upper layers. There are plenty of possible solutions but the target of the proposed approach is to utilize a synchronization protocol that also can accommodate for different routing protocols above it. The user can see an IP-based transport layer. The API has to be common to all the Ad Hoc nodes, or at least it has to be able to provide the ports or interfaces that can be used to communicate with the rest of nodes.

A. Replication API

The API provides the necessary primitives to model the replication process according to the routing requirements. Thus, the API accommodates the rate of updates and replications. Furthermore, the API provides the mechanism to initiate a node discovery procedure and Attach-Detach



Application generated
Ad Hoc creation
(Gaming or chat session)
Application and services
Directory Services
User Services (Messaging Gaming)
Future Services
Services and session management (SIP,HTTP)
Auto Configuration module
API
Context-Sensitive Roaming Layer
API
Routing Module
Routing Algorithm Module
Routing Data
API
API
Synchronisation Replication module (SCSP)
IP Layer (IPv4/IPv6)
Link Layer Protocol
BT-SDP
Communication Module Physical Layer (Bluetooth, WLAN, UMTS,)
Find out available networks
Find out available services
Query for paging request among multiple interfaces
Area Network signal received (Type = WLAN, BT, UMTS)
Bearing multiple Beaconing
Activate multiple interfaces
Strong Signal received (WLAN, UMTS, BT)
Generate BT routing tables for piconet infrastructure and start node configuration, create WLAN routing tables and register in UMTS
Initiate SDP discovery, Create Service discovery messages, open PDP and log in UMTS networks

Figure 4. Application based triggering for roaming in Ad Hoc networks.

mechanism among the smart Ad Hoc nodes. This section highlights the different alternatives for selecting a suitable synchronization protocol.

The most popular synchronization and replication protocols are the Internet Cache Protocol (ICP) [22], Cache Digest, CARP (Cisco) [23], WCCP (Microsoft), Flood-d [24], LDUP [25] and SCSP. SCSP uses a request-reply paradigm implemented independently of the transport protocol. SCSP uses the group-based replication and the creation of overlapping caching meshes to forward objects among groups. Another important technique adopted for cache-to-cache communication is the notion of "cache digest" (as in SQUID [26] and Summary Cache). Digests are used to reduce intercommunication load by exchanging summarized information instead of the whole object. Our proposed replication module, within the routing module, considers SCSP [27] as the generic protocol for synchronization purposes that can be used over any transport mechanism.

The rest of this section describes the SCSP as one alternative for implementing the replication module. SCSP considers a fully meshed replication without any topology restrictions. This approach guarantees that the information is updated in the servers that compose the group of smart nodes.

B. *Replication API with SCSP*

The routing protocol can be implemented independently of the replication protocol and use the SCSP synchronization capabilities through a common API. The generic nature of SCSP and its scalability permits to automatically include new nodes as part of the group [28].

SCSP starts the replication using a Hello protocol, which finds out which are the nodes that are part of the group and belong to the same family (Family_ID). The Hello message can be extended with a security mechanism that includes the node credentials to authenticate the new node and add it automatically to the group. Afterwards, SCSP starts the synchronization replication of the cache content among the nodes within the same group. The Hello protocol is used for keeping alive the connection among the nodes of the group. The Hello message contains a parameter that indicates the timeout for sending keep-alive messages. This parameter can be changed in subsequent messages to decrease the bandwidth consumption in case the channel is reliable enough.

C. *SCSP Scalability*

The results of the scalability analysis show the protocol performance when it works with an increasing number of nodes. We installed a database environment using MySQL and used it to recreate data sets for the trials. Different cache models such as Client Caching, Local Network Caching, Push Caching, Adaptive Web Caching, and Hierarchical Caching were analyzed. We find hierarchical caching as the most suitable for the Ad Hoc routing implementations. The hierarchical scheme simplifies the network by breaking it into small sub-networks where each level is responsible of its own cache.

In this testbed [29], SCSP is used over UDP but it is built in such modular way that the communications module can be changed to run over any other transport layer. The group-based replication and the creation of overlapping caching meshes to forward objects among the nodes are the basis for applying SCSP to Ad Hoc networks. The *cache digest* is used in SCSP to reduce intercommunication load by exchanging summarized information instead of the whole object. The tests performed in our prototype show the behavior of SCSP for two generic configurations: star and inline. *Star* configurations show the distribution of load for caches serving as link points to more than two servers, i.e. they generate convergent replication flows that must be processed in the server and delivered to other Direct Connected Servers (DCS). *Inline* configurations show the processing of cumulative replication load. Updates received in the Local Server from any DCS must be flooded to every DCS in the Server Group (SG) but the one that originated the update. *Inline* topologies generate longitudinal replication flows that hop several times before terminating. Both configurations are shown in Figure 5.

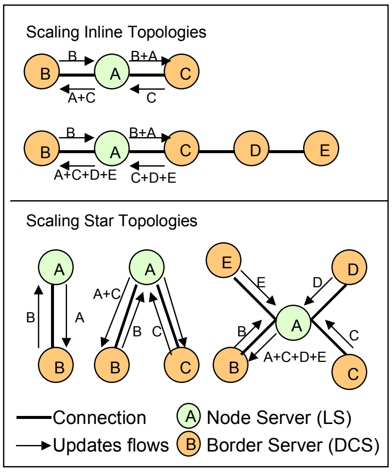


Figure 5. Inline and star topologies.

D. *SCSP Test Results*

This section includes the results of tests made with the replication module using SCSP. From the results (Figure 6) we deduce that the *database size* is the main factor for the alignment time. The evaluation shows the time and messages required for initializing different sizes of caches. Cache initialization will be needed when a new smart node joins an ad-hoc group. For moderate size groups the lower left-hand area of the curve in Figure 6 applies.

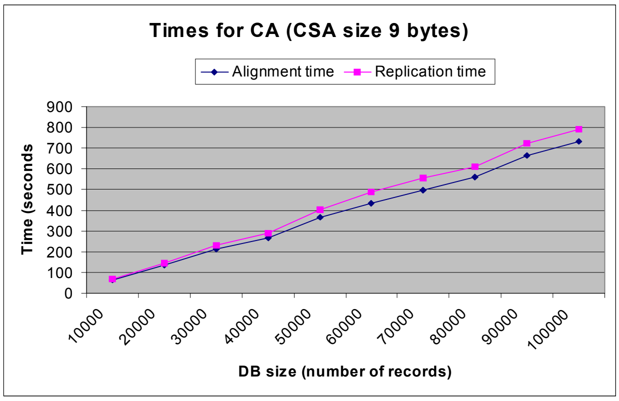


Figure 6. Alignment and replication for fixed entry size and variable database size.

The *Mean Update Time* (MUT) is the rate by which route updates are passed from the routing algorithm towards the replication module. Lower MUTs generate higher update rates, i.e. higher number of Cache State Alignments (CSAs) to replicate per unit of time. These tests show the response of the protocol to several update flows and finds the upper limit of the protocol in terms of update rate.

The relation "*CSA size / maximum message size*" describes the number of CSAs sent per message and the bandwidth consumption (Figure 7). This proportion can be tested for fixed MUTs and DB sizes to measure the influence that it has on the replication. Longer messages send more replication data in a packet, but are less resilient to errors. The influence of *maximum message size* in the replication depends on the communication bandwidth.

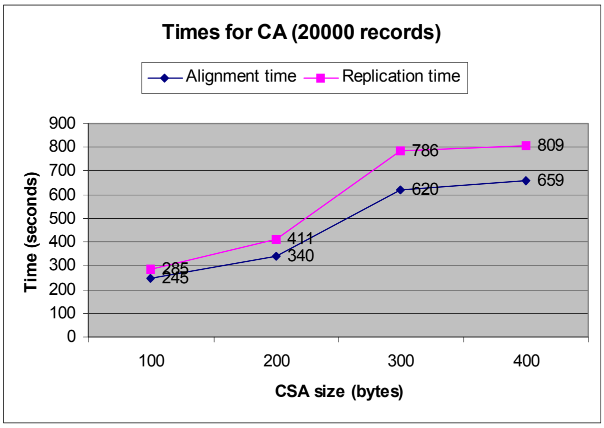


Figure 7. Alignment and replication for fixed database size and variable entry size.

The tests focus on the SCSP performance. Thus, they concentrate on the analysis for different sizes of CSA records, i.e. different sizes of routing and network capabilities information records.

In the test results an important area of low performance in the behavior of the protocol is found in tests for the star topology. The replication time grows significantly compared to inline topology. Therefore, we conclude that it is preferable to have a set of different levels of inline associations of server groups (SG). Another important outcome is that SCSP will perform more efficiently when used for small tables like in limited groups composed of reduced number of smart nodes as proposed for the suitable Ad Hoc architecture. From Figure 6 and 7 we deduce that a suitable size of a cache entry would be 200 bytes (Figure 8 shows an example of data structure for service information) and that the number of entries should preferably be kept under 10 or 20 thousand.

V. Ad Hoc Network Incentive

The Ad Hoc networks require a strong incentive to persuade the industry about their benefits. Ad Hoc networks are based on the collaborative concept for creating a network for communication among multiple nodes. This concept requires that all nodes participate in forwarding the packets of other terminals despite that they are not participating in the communications. We call this the node coordination problem. Therefore, it requires willingness to participate even without getting any benefit and with decreasing resources instead. This idea leads to two different use cases where the Ad Hoc network can be implemented: private and public networks.

A. Private Ad Hoc Networks

This case refers to the situation where the Ad Hoc network, including all the nodes that participate, is private and owned by a single entity or a set of entities with a common goal. An example of the common goal is emergency rescue by many organizations. In this type of Ad Hoc networks the node coordination problem is naturally resolved.

A clear application of private Ad Hoc networks can be found in companies that provide terminals capable of setting up Ad Hoc networks to their employees. In this case, the company builds its own network infrastructure using the terminals owned by the company and provided by the company for its employee's use. In this application the ad hoc network technology with emerging mobile terminals can potentially replace or complement current PBX –networks. When the terminal is out of the coverage area of the company, it behaves as a normal 2G or 3G terminal. Thus, the terminal performs the service negotiation as a part of the private Ad Hoc network every time when it enters the coverage area of the company. In this type of environment the appropriate Ad Hoc routing algorithm would be based on clusters or zones. Furthermore, when the company has different sites spread in a wide area, it will have fixed network entities to behave like cluster gateways.

Another case of this kind of Ad Hoc networks can be established within private companies that use this type of network for controlling industrial processes. In this situation many of the nodes are dummy with minimum memory and resources requirements and few are smart nodes that collect all the necessary information about process status and other issues.

B. Public Ad Hoc Networks

In the second case, the Ad Hoc networks are created using terminals that are owned by different entities without a common purpose among them. In this case the relationship between the ad hoc nodes should be created based on a basic negotiation or tacit agreement.

There are many other types of agreements that can be established to guarantee the collaborative behavior of the nodes. This type of negotiation should be established as part of the configuration mechanism of the Ad Hoc node. Thus, when the node enters the coverage area of an Ad Hoc network, it will be requested to participate. This negotiation will be based on certain benefits that the node will get from participating. Despite the nodes do not get any immediate benefit, they obtain bonuses that can be exchanged later when the nodes require certain services.

Therefore, for deploying the public network, security mechanisms are required to authenticate the nodes that are active in the Ad Hoc network. It is also necessary to legitimate the traffic that will be forwarded by the non-active nodes that still participate in the Ad Hoc network. After authenticating the originating nodes and the forwarded traffic, the non-active nodes collect information for creating bonuses that would be used later on for their own traffic.

Thus, a new service not considered yet in Ad Hoc networks is required: the charging or billing center that keeps track of the bonuses and authenticate the creation of the bonuses. This center has to be tightly linked to the security mechanism and it can be implemented either in a fixed node that acts as a Certification Authority (CA) or it can be distributed among various nodes that take that responsibility in the Ad Hoc network.

This mechanism can be implemented using extensions in the routing messages. Figure 8 shows the charging information that can be inserted in the service extension. The intermediate node or collaborative Ad Hoc node collects charging info that would be reported to the billing center.

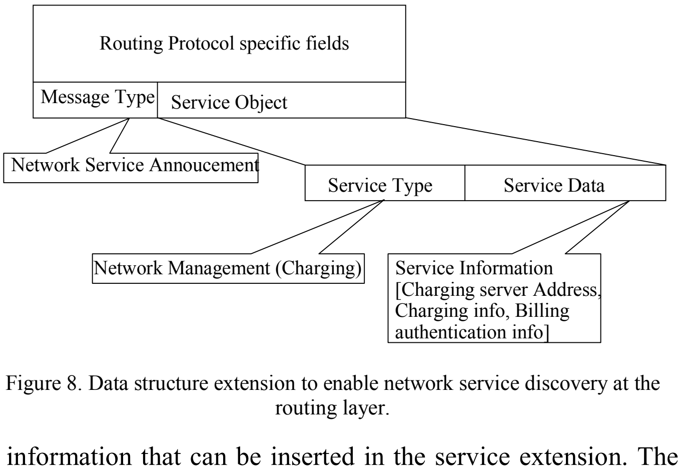

Routing Protocol specific fields
Message Type
Service Object
Network Service Annoucement
Service Type
Service Data
Network Management (Charging)
Service Information
[Charging server Address,
Charging info, Billing
authentication info]

Figure 8. Data structure extension to enable network service discovery at the routing layer.

The node will use the charging info for getting some other network services when the node requires them. Therefore, provided there are enough trustworthy nodes, a co-operative network can be established, where all the nodes that participate in the routing process collect charging information. That information is used to generate a set of bonuses that will be used when the node requires similar service.

VI. CONCLUSIONS

The Ad Hoc networking technology is an area under development that has become popular in the recent years. In deployment of Ad Hoc networks, there are many challenges related to mobility management, routing, service discovery, etc. Nevertheless, in this paper we have presented a proposal to divide the problem in smaller parts and treat them separately. By reducing the scope of the problem, the solution can be found as a conjunction of building blocks. The first issue we covered is the routing and network capabilities discovery. Within that wide area, the scope was reduced to the replication problem. Thus, an API between the routing and the replication layer was suggested. The routing functionality is kept as an application on top of the replication process. After analyzing the replication problem, we selected and tested one possible solution that uses SCSP as the synchronization protocol. A prototype was implemented and from the results we deduced that SCSP is only suitable for replicating small databases using an inline schema and an enhanced algorithm is preferable. Additional enhancements are recommended for using SCSP in Ad Hoc networking technology.

The proposed architecture is based on a new layered structure that includes a Context Sensitive Roaming Layer for handling the information received from the routing and application layer to facilitate the selection of the most suitable interface. This approach requires the classification of the terminals into dummy and smart nodes. This taxonomy is required as part of the hybrid reactive-proactive routing approach, selected as the optimal approach for Ad Hoc networks.

One issue is that commercial Ad Hoc networks do not conform to the collaborative philosophy where every node should forward the traffic from other nodes despite it is depleting its own battery and resources. Therefore, the proposed solution is based on the concept where certain nodes with enough resources are providing their resources in benefit of the rest of the Ad Hoc nodes. Those nodes collaborate because they are part of the same community where they are sharing resources towards a common benefit (chatting, messaging, group discussions or gaming, commercial announcements, etc).

We finally conclude that by breaking down the problem into smaller sub problems and by focusing on a modular solution for replication and routing, the Ad Hoc routing, service discovery and collaboration problems can hopefully be solved.

REFERENCES

[1] C. E. Perkins and E.M. Royer, "Ad-hoc On Demand Distance Vector Routing", Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, February 1999.

[2] D.b. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", in Mobile Computing, edited by T. Imielinski and H. Korth, chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[3] V. Park and M.S. Corson, IETF MANET Internet Draft "draft-ietf-MANET-tora-spe03.txt", November 2000.

[4] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Lanouiti, L. Viennot and T. Clausen, IETF MANET Internet Draft "draft-ietf-MANET-olsr-02.txt", July 2000.

[5] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks", IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks, vol. 17, No 8, pp. 1369-1379, August 1999.

[6] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol", IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks, vol 17, No 8, pp. 1395-1414, August 1999.

[7] R. Droms, "DHCP: Dynamic Host Configuration Protocol", RFC 2131, Network Working Group, IETF, 1997

[8] P. Mockapetris, "Domain Names: implementation and specification", RFC 1035, Network Working Group, IETF, 1987.

[9] P. Srisuresh, M. Holdrege," IP Network Address Translator (NAT) Terminology and Considerations ", RFC 2663, IETF, Aug 1999.

[10] M. Handley, H. Schulzrinne, E Schrooler, J. Rosenberg. "Session Initiation Protocol", RFC 2543, IETF.

[11] Sun Microsystems, Inc, "JiniTM Architecture Specification", Version 1.2, http://www.sun.com/jini/. December 2001.

[12] Universal Plug and Play Forum, "Universal Plug and Play Technology UPnP", http://www.upnp.org/.

[13] Salutation Consortium, "The Application Programmer's Interface of the Salutation Architecture ", http://www.salutation.org/.

[14] Open Services Gateway Initiative (OSGi), "OSGi - The Managed Services Specification", http://www.osgi.org/.

[15] E. Guttman, C. Perkins, J. Veizades, M. Day, "Service Location Protocol, Version 2", RFC 2608, IETF, Jun 1999.

[16] Bluetooth SIG, "Specification of the Bluetooth System Profiles, Version 1.0B volume 1, 1999. Part K:2", http://www.bluetooth.com/link/spec/bluetooth_k2.pdf.

[17] J. Costa-Requena, “Auto Configuration in Ad Hoc networks”, Residential and Virtual Home Environments - Seminar on Internetworking, Spring 2002, TML-C8 HUT TML Course T-110.551, ISBN 951-22-5959-1 ISSN 1455-9749, http://www.tml.hut.fi/Studies/T-110.551/2002/papers/May/index.html

[18] J. Luciani, G. Armitage, J. Halpern,N. Doraswamy, "Server Cache Synchronization Protocol (SCSP)”. RFC 2334. April 1998.

[19] T. Berners-Lee, R. Fielding and M. McCahill, "Uniform resource locators (URL)", RFC 1738, IETF, Dec 1994.

[20] 3rd Generation Partnership Project, 3GPP. www.3gpp.org.

[21] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2426, IETF, Dec. 1998.

[22] D. Wessels, K. Claffy "Internet Cache Protocol (ICP) version 2", RFC 2186, IETF, Sep 1997.

[23] V. Valloppillil, K. Ross, "Cache Array Routing Protocol v1.0", IETF draft, Aug 1998.

[24] Obraczka, K., Danzig, P., “Evaluating the Performance of Flood-d: A Tool for Efficiently Replicating Internet Informationn Services”. IEEE Journal on Selected Areas in Communications, VOL. 16, NO. 3, April 1998. 14 pages.

[25] Stokes, E., Good, G. “The LDUP Replication Update Protocol”. Internet draft, July 15, 2000.

[26] SQUID "Squid Web Proxy Cache", http://www.squid-cache.org/

[27] Jose Costa-Requena, "An Implementation of the Server Cache Synchronization Protocol”. Helsinki University of Technology, Laboratory of Telecommunications Technology, 1999.

[28] Igancio Gonzalez Olias, "Security and Autoconfiguration of Location Servers for IP Telephony”, Helsinki University of Technology, Laboratory of Telecommunications Technology, 2002.

[29] Julio Ramirez Yébenes, “A Scalability Analysis of the Server Cache Synchonization Protocol (SCSP)”, Helsinki University of Technology, Laboratory of Telecommunications Technology, 2001.