

# Use of Context-Awareness in Mobile Peer-to-Peer Networks

Richard Gold  
GMD FOKUS  
Kaiserin-Augusta-Allee 31  
D-10589 Berlin, Germany  
gold@fokus.gmd.de

Cecilia Mascolo  
Dept. of Computer Science  
University College London  
Gower St, London WC1E 6BT, UK  
c.mascolo@cs.ucl.ac.uk

## Abstract

*Mobile ad-hoc networks are an emerging research field due to the potential range of applications that they support and for the problems they present due to their dynamic nature. Peer-to-peer is an example of a class of applications that have recently been deployed on top of ad-hoc networks. In this paper we propose an approach based on context-awareness to allow peer-to-peer applications to exploit information on the underlying network context to achieve better performance and better group organization. Information such as availability of resources, battery power, services in reach and relative distances can be used to improve the routing structures of the peer-to-peer network, thus reducing the routing overhead.*

## 1 Introduction

Mobile ad-hoc networks [19, 8] have emerged as an important new research field, because of their high potential and the problems that they present due to their highly dynamic nature. In a similar way, peer-to-peer application layer networks [3, 18] (henceforth referred to as peer-to-peer networks) have also been very popular in recent years for the added flexibility that they bring to communication and interaction paradigms.

However, both these areas share a design restriction as they both present a flat routing space (i.e., no hierarchy) and lack of an infrastructure to assist routing. In these types of networks, every node is an endpoint and a router, which implies that the per-node resource consumption is much higher than in traditional networks. Moreover, the network load is typically much higher as broadcast techniques have to be used to perform node and route discovery in both ad-hoc [14] and peer-to-peer networks [15].

Given the popularity and the dynamicity of the peer-to-

peer paradigm there has been some recent research in pushing the use of these technologies on top of mobile ad-hoc networks [4, 13, 2] in order to support data sharing among roaming peers and to exploit peer resources when possible.

Mobile devices have usually very limited capabilities and resources. The indiscriminate use of peer-to-peer applications, which imply the usage of broadcast for peer discovery and content-location, may result in very low performance due to these constraints. Peer-to-peer systems are normally completely agnostic to the network that they are running over. The introduction of hierarchy and grouping would introduce performance benefits, as broadcasts would be significantly reduced, however this is a new area of research which presents many challenges.

In this paper we propose the use of context-awareness strategies (also referred to as cross-layering) in order to overcome these limitations. Considerable benefit can be drawn from combining the information provided by the underlying ad-hoc network concerning the availability of resources at specific moments and locations and supplying this to the peer-to-peer network. This information may be used to construct on-the-fly routing structures in the peer-to-peer network, which reflect the internal and external resources of the nodes in the network and can help to reduce the overhead required for routing. We believe research in the direction of route optimization and resource exploitation would lead to a larger diffusion of peer-to-peer paradigms on mobile ad-hoc systems and the increased likelihood of exploiting these paradigms for different applications.

The paper is organized as follows: Section 2 illustrates our approach to route aggregation in peer-to-peer networks. Section 4 shows how the use of reflection can be applied in mobile settings for gathering information from the network layer and expose it to the upper layers permitting context-awareness. Section 5 concludes the paper illustrating some future work.

## 2 Dynamic Route Aggregation

In this section we describe our approach of route aggregation in peer-to-peer networks (more details in [6]), then we illustrate the application of this approach to mobile peer-to-peer networks. As we show, the usage of peer-to-peer applications over ad-hoc networks implies very high load on the hosts as the condition of groups need to be kept monitored, given the high dynamicity of the structures. Therefore, optimization on the monitoring and introduction of more efficient route aggregation need to be applied. Section 4 will describe the use of context awareness techniques for allowing more efficient route aggregation.

### 2.1 Route Aggregation at the Application-layer

In [6] we describe an architecture for introducing dynamic grouping in a peer-to-peer network with a flat routing space. The approach described in that paper allows us to introduce hierarchy and route aggregation points into the network in order to avoid problems with decentralized networks like Gnutella [18] as reported in [15]. In fact, by effectively having to query all nodes in the network (although this never actually occurs due to limited network reachability), the resulting load that is imposed upon the network causes inordinately large quantities of bandwidth to be consumed [17]. Although other projects such as [10, 9] have investigated the usage of hierarchies to ease routing overhead, they have only used a two layer hierarchy. We propose the usage of a multi-level hierarchy in order to exploit the scalability offered by a more open-ended system.

Our solution introduces the notion of hierarchy and route aggregation for peer-to-peer networks, in a similar way that BGP did for the Internet [12]. BGP introduced the notion of an Autonomous System (AS) which allowed an entire range of addresses to be encapsulated by a single routing entry. E.g., the AS AS-FOKUS could have the range 193.175.135.x where x is a number between 1 and 255. Thus, in the inter-AS BGP routing tables, the AS can be represented by one entry instead of having to have every single machine in the AS individually listed in the routing tables.

We organize the network in groups led by *Aggregation Points* which are the equivalent of BGP Speakers. Groups consists of nodes aggregated to a specific aggregation point. A picture of this layered structure can be found in Figure 1.

Given the hierarchy of the virtual network, queries for a certain item of content are first sent to the AP of the group to ensure that if the data is locally available, then we are able to first use that copy of the data instead of flooding the entire peer-to-peer network with requests. When the requested item of content is not hosted by a neighboring node, there are two possible scenarios that could be pur-

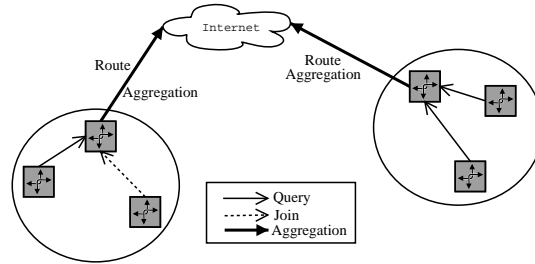
sued. The first scenario assumes that the network is aggregation end-system addresses in a similar fashion to BGP. This is the method detailed in [6]. In this case the Aggregation Point (equivalent to BGP-speaker/border router) of the group then looks up in its routing tables which AP is advertising the block of addresses which contain the address that we wish to access. This recipient AP then forwards the search queries (or download request etc.) to the destination node.

The second scenario involves using a content-aggregation scheme (also called Content-Addressable Networks (CANs) [16]), such as detailed in [7]. This system attempts to construct a distributed index for peer-to-peer applications so that the number of search queries are greatly reduced. In such a scheme the APs aggregate *content* instead of end-system addresses. Such a content-aggregation scheme allows routing decisions to be based on where a specific item of content is located, not on the address of the end-system which may happen to host the item of content. This separation between the content that the user is trying to locate and the physical end-system which is actually hosting the content allows us to deal with node mobility in an Ad-Hoc network. A node may move through the network, but as long as it can reach the specific aggregation points for the content that it wishes to advertise, its physical location is not important as far as search queries are concerned.

In an Ad-Hoc network it is fairly typical that a node does not always have the same address (IP addresses for mobile nodes are often obtained dynamically with protocols like DHCP or ARP, for example). With a content-based routing system, such as the one described in [7], we can easily deal with transient addressing by virtue of the fact that our peer-to-peer application only uses node addresses to actually retrieve the file once it has been located. All other network operations are based upon the content contained on the nodes of the network and is completely agnostic to the actual address used to locate these nodes.

### 2.2 Integrating Application-layer Route Aggregation and Link-layer Information

The approach to application-level network aggregation just described can also be layered over mobile ad-hoc networks. The advantages of having peer to peer paradigms enabled over the ad-hoc network are many, and in general they have to do with availability and exploitability of peers' resources. The main difficulty in integrating the two approaches is the limited radio range and sporadic connectivity of ad-hoc nodes. For example, the usage of link-layer technologies such as IEEE 802.11b, Bluetooth etc. means that we can expect nodes to pass in and out of radio range of each other quite often. So a node which is a member of a



**Figure 1. The hierarchy in an Aggregation Point based peer-to-peer network.**

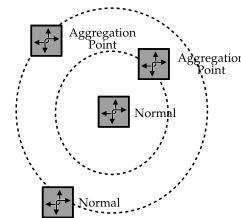
peer-to-peer group will suddenly not be able to contact any of its group members. As well as this, environmental conditions can easily mean that a node cannot reach other nodes because a physical obstacle (e.g. a wall or a building) is in the way.

In order for route aggregation to be integrated at both the application and link layers, we need to ensure that the application-layer groups reflect the current link-layer conditions and that there exist mechanisms for recovering from node or group loss. The high dynamicity of the underlying network structure means that the aggregation hierarchies can be very volatile. In particular, nodes can join and leave the aggregation groups rapidly and frequently. It may be convenient to ensure that aggregation points in an Ad-Hoc network withdraw from their role more often than in a fixed network, for example if their resources are too scarce (i.e., low battery power), or if the applications available in the area are of sub-standard quality. The overhead of managing this group dynamicity and the number of the join and leave operations is much higher than in standard peer-to-peer networks. In order to handle this complexity and to optimize the distribution of the groups and aggregation points, we have developed some techniques for the monitoring of context and for the delivery of this information to the peer to peer application. We will introduce these techniques in the next section.

### 3 Peer-to-Peer node group operations

We now illustrate the volatility of the peer-to-peer level topology and highlight the complexity and the issues that will be solved using context-awareness techniques. We assume a multi-hop connectivity where each host may reach other hosts through the use of the neighbouring hosts acting as routers for its packets. Multi-hop is an important attribute in an Ad-Hoc network where there are even less guarantees of routing transitivity than in a fixed network due to not only incomplete routing information but also due to the limited reachability of the wireless links.

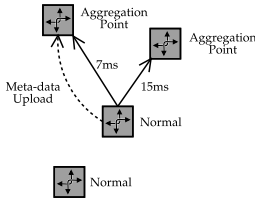
Whenever a node realizes it has more than  $N$  physical neighbours it triggers a *leader election* [5] in order to elect



**Figure 2. Aggregation Point discovery.**

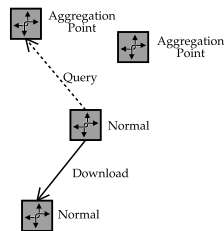
an *aggregation point* (AP) which will handle the requests of the nodes in the group and diminish the flooding effect due to message broadcasting. However, due to the volatile network topology, the aggregation point status need to be monitored and, if necessary changed and we will discuss why and how in the following.

- The "peer-to-peer" level operations allowed for a generic roaming host are:
  - *Aggregation Point Choice*: Whenever a node joins the network it has to choose an aggregation point and join the group. In order to do so the node has to sense the network and the availability of aggregation points (Figure 2). As we will see in Section 4 to choose the more convenient aggregation point some context conditions may be sensed and considered (Figure 3);
  - *Node Location*: Once a node is connected to an aggregation point it can begin to query the peer-to-peer network for available content (Figure 4). When the address of the node offering the requested content is retrieved, a direct connection to the node can be established and the communication happens without further involving the ag-



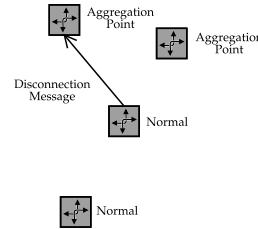
**Figure 3. Measuring Aggregation Point suitability and uploading Meta data.**

gregation point; this strategy allows a lower load on the network, as the look-up service is handled by the aggregation point. However the careful choice of the aggregation point is important. Again, context conditions such as distance, latency and load of the aggregation point are essential in the choice of the aggregation point, in order to make efficient use of the network and node resources.



**Figure 4. Querying and downloading of content.**

– *Graceful Disconnection*: When a normal node wants to disconnect it notifies the aggregation point and then leaves (Figure 5). In this case the aggregation point may update its records. However, it may be the case that the node fails or leaves without notification. In order to handle this situation the aggregation point regularly polls his group using heartbeat messages. When



**Figure 5. Graceful disconnection.**

a node does not reply the AP updates his tables.

- An aggregation point node may perform the following operations:
  - *Initialization*: Once the leader election is terminated the elected node advertises his presence and begin to record information from the subscribing nodes;
  - *Service Information Update*: Once in a while the AP updates his information about the nodes in the group and about the network service availability exchanging records with other APs;
  - *Graceful Disconnection*: An AP may withdraw from service for different reasons:
    - \* *Group Splitting*: When the group of the AP gets too big the AP withdraws in order to allow a better distribution of the nodes and new election of other APs to be triggered;
    - \* *Group Merging*: When an AP realizes that too few nodes are connected to it, it withdraw and try to associate to another AP;
    - \* *Other Reasons*: An AP may withdraw also for other reasons which may have to do with personal conditions such as low battery power, scarce resources, heavy computational load.

Given this behaviour it may have become clear why and how context-awareness helps in improving efficiency of the algorithms. Monitoring of context conditions and application awareness are the ways in which we enhance our system and we will discuss them in the next section.

## 4 Using Context-Awareness for Enhancing Route Aggregation

The use of context awareness techniques allows us to more efficiently deploy peer to peer structure on top of ad-hoc network. In fact, context-awareness allows us to make better and more precise decisions of how to use resources and how to assign responsibilities.

In this section we will give an overview of the techniques used for this purpose. In some previous work we used reflection techniques to enhance application awareness on the status of the surrounding resources [11]. In this paper we use these techniques to allow the peer-to-peer level network to acquire some insight on the status of the surrounding resources, in order to take more intelligent decisions, for instance, on how to construct and tear down the aggregation groups.

By definition, reflection allows a program to access, reason about and alter its own interpretation. The principle of reflection has been mainly adopted in programming languages, in order to allow a program to access its own implementation. In the context of our approach reflection permits the flow of environmental information such as the hosts/services currently in reach, the remaining battery life, the location and the bandwidth conditions to reach the application layer.

The ability to format this context information in a way which is understandable by the application layer is important. Our work in the areas of mobile computing middleware and meta-data [11] becomes useful at this stage.

In such peer-to-peer applications described in [6] we presented a framework for providing policy-based measurements to components of our route aggregation groups in order to decide which group a new node should join. In mobile ad-hoc setting, a basic mechanism which relies on the choice of the "first replying aggregation point" is obviously not good enough, as the distance and load of the AP would influence the efficiency of the communication, not only between the two nodes but of all the surrounding networks. On the contrary, for performance reasons, many different conditions should be considered for the choice of the aggregation point to connect to, such as the distance to it, the latency, the number of other hosts connected to it. It would be ideal if the node that has to make a choice on which AP to join, as well as measuring delay and distance, it could also receive information about the "internal resources" (i.e., the processing power, the connected clients, remaining battery life etc.) so to make a more intelligent choice. We then enhance the basic choice algorithm through context-awareness by making decisions based upon the current state of the candidate APs and, in general, the prevailing environmental network conditions.

Also the reasons that induce an AP to withdraw may be

made context dependent. This would allow the AP to withdraw whenever either the context conditions are not worthwhile for it to deliver its service (i.e., number of connecting nodes to low or too high), or when the device conditions are not good anymore (e.g., when the battery power is low, or other application running on the device are consuming large numbers of CPU cycles).

### 4.1 Use of Reflection and Meta Data for Context Awareness

We have described the use of context awareness in the context of peer to peer network on ad hoc network. We plan to use an approach similar to the one we described in [11] which exploits meta-data (and in particular XML [1]) for the formalization of policies and context information. Reflection is the mean by which context information is gathered from the network and lower layers and delivered to the application layer.

Information such as which hosts or services are in reach, battery power level, and in general device resources could be collected by a middleware and an application specific API may be defined to access this information and use it making intelligent decisions as described in the previous section. With XML the structure of the information can be defined quite clearly and the use of XML related technologies would allow the easy access to specific fields or attributes when needed.

An example of the encoding of context information is presented in Figure 6: the battery status and the memory load are encoded using meta data. The middleware has the role to gather information from the network and operating system and present it in this format to the application. The application level network's algorithms use specific calls defined in an API by the middleware in order to access the information and react to context change.

```
<resource name="battery">
  <status value="high"/>
</resource>
<resource name="memory">
  <status value=x/>
</resource>
```

**Figure 6. XML encoding of context information.**

Information related to location and delays may also be encoded in this way as long as a specific meta-data structure is defined together with a precise API.

## 5 Conclusions and Future Work

We have presented an outline of a system which integrates mobile ad-hoc networks and peer-to-peer networks to facilitate flexible and adaptive applications. We have described the problems inherent with the lack of hierarchy of those networks to assist with node and route discovery. It is our belief that these problems can be dealt with by the creation of a hierarchy. Hierarchy facilitates route aggregation to decrease the network load imposed by traditional broadcast-style techniques.

Due to the high dynamicity of ad hoc and peer-to-peer networks, we are faced with many research challenges in order to integrate these two approaches together. We developed a model for peer-to-peer networks which can also be applied in mobile setting. However, in order to improve the performance of the application we introduced some context awareness mechanisms to allow the application algorithm to make more intelligent decisions. We use meta data and reflection techniques to allow this flow of information between the network/OS and the application.

Security issues still have to be tackled, in particular the security of the aggregation point is quite important as information on which content is located on which nodes is stored there. We also plan to implement a prototype of the model presented and make some efficiency measurements in different mobility contexts (such as slow mobility or high mobility settings).

**Acknowledgements:** We would like to thank Christian Tschudin for the term "cross-layering" and also Jon Crowcroft, Licia Capra, Wolfgang Emmerich, Torsten Ackemann and Anke Speer for their useful comments on a previous draft of this paper.

## References

- [1] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language. Recommendation <http://www.w3.org/TR/1998/REC-xml-19980210>, World Wide Web Consortium, Mar. 1998.
- [2] P. Charas. Peer-to-Peer Mobile Network Architecture. In *Proc. of the International Conference on Peer-to-Peer Computing (P2P2001)*, Linköping, Sweden, Aug. 2001.
- [3] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. FreeNet: A distributed anonymous information storage and retrieval system. *Workshop on Design Issues in Anonymity and Unobservability*, 2000. <http://www.freenetproject.org/icsi-revised.ps.gz>.
- [4] G. Cugola and G. Picco. PeerWare: Core Middleware Support for Peer-To-Peer and Mobile Systems. Technical report, Politecnico di Milano, May 2001. Submitted for Publication.
- [5] H. Garcia-Molina. Elections in a distributed computing system, 1982. H. Garcia-Molina. Elections in a Distributed Computing System. *IEEE Trans. on Computers*, 31:48–59, January 1982.
- [6] R. Gold. Self-Organizing Route Aggregation for Active Ad-Hoc Networks. In *Ghosts of the Net short paper session, OpenArch*, 2001. <ftp://ftp.fokus.gmd.de/pub/glone/usr/rgo/pub/ghosts.ps.gz>.
- [7] R. Gold and D. Tidhar. Towards a Content Aggregation Network. In *Proc. of the International Conference on Peer-to-Peer Computing (P2P2001)*, Linköping, Sweden, Aug. 2001.
- [8] I. M. W. Group. MANET Charter, 2000. <http://www.ietf.org/html.charters/manet-charter.html>.
- [9] Z. Haas and M. Pearlman. Zone Routing Protocol. <http://community.roxen.com/developers/idocs/drafts/draft-ietf-manet-zone-zrp-02.html>, 1999.
- [10] S. Jain, R. Mahajan, and B. Niswonger. Self-organizing overlays. Technical report, Washington University, 2000. <http://www.cs.washington.edu/homes/sushjain/overlay/reports/spring2000/%report.ps>.
- [11] L. Capra and W. Emmerich and C. Mascolo. Reflective Middleware Solutions for Context-Aware Application. In *3th International Conference on Metalevel Architectures and Separation of Crosscutting Concerns (Reflection 01)*, LNCS, September 2001. To Appear.
- [12] K. Lougheed and Y. Rekhter. Request For Comments 1163: A Border Gateway Protocol, 1990. <http://gatekeeper.dec.com/pub/net/info/RFC/rfc1163.txt>.
- [13] C. Mascolo, L. Capra, and W. Emmerich. An XML-based Middleware for Peer-to-Peer Computing. In *Proc. of the International Conference on Peer-to-Peer Computing (P2P2001)*, Linköping, Sweden, Aug. 2001.
- [14] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad-hoc network. In *Mobicom'99*, 1999. <http://www.acm.org/pubs/articles/proceedings/comm/313451/p151-ni/p151-n%i.pdf>.
- [15] A. Pava. Gnutella is dead. *ZDNet Music*, 2000. [http://music.zdnet.com/features/highnote/092100\\_gnutella\\_dead.html](http://music.zdnet.com/features/highnote/092100_gnutella_dead.html).
- [16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of SIGCOMM 2001*, 2001. <http://www.acm.org/sigs/sigcomm/sigcomm2001/p13-ratnasamy.pdf>.
- [17] J. Ritter. Why Gnutella can't scale, no really..., 2001. <http://www.tch.org/gnutella.html>.
- [18] C. D. S. Solutions. Gnutella protocol specification v0.4. <http://dss.clip2.com/GnutellaProtocol04.pdf>, 2001.
- [19] C. Tschudin, H. Lundgren, and H. Gulbrandsen. Active Routing for Ad-Hoc Networks. *IEEE Communications Magazine*, pages 122–127, 2000.