

# Towards an Integrated Architecture for Peer-to-Peer and Ad Hoc Overlay Network Applications

Lu Yan, Kaisa Sere, Xinrong Zhou  
Turku Centre for Computer Science (TUCS) and  
Åbo Akademi University  
FIN-20520 Turku, FINLAND  
{Lu.Yan, Kaisa.Sere, Xinrong.Zhou}@abo.fi

Jun Pang  
Centrum voor Wiskunde en Informatica (CWI)  
P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands  
Jun.Pang@cwi.nl

## Abstract

*Peer-to-peer (P2P) networks and mobile ad hoc networks (MANET) share some key characteristics: self-organization and decentralization, and both need to solve the same fundamental problem: connectivity. We motivate a study for the convergence of the two overlay network technologies and sketch an evolving architecture towards integrating the two technologies in building overlay network applications.*

## 1. Introduction

Peer-to-peer (P2P) systems are self-organizing, decentralized overlay networks, in which participating nodes contribute resources and cooperate to provide a service. Mobile ad hoc network (MANET) is an autonomous system of mobile hosts (also serving as routers) connected by wireless links, the union of which forms a communication network with arbitrary communication topologies. A P2P network consists of a dynamically changing set of nodes connected via an infrastructure-based network, while a MANET consists of mobile nodes communicating with each other using multi-hop wireless links.

P2P and MANET share some key characteristics: self-organization and decentralization, which lead to a lot of similarities between the two overlay networks [2]:

- **Dynamic topology.** A node in P2P and MANET may join or leave the network at any time and the position of a node in MANET is changing arbitrarily, which leads to no constant routes for any nodes.

Both networks have a dynamically changing network topology.

- **Hop connection.** Connections in P2P and MANET are established via exchanging beacon messages only between neighbor nodes. A single hop connection in P2P is typically via TCP links without physical limits, while a single hop in MANET is via wireless links which are usually limited by the radio transmission range.
- **Routing protocol.** Both P2P and MANET routing protocols have to deal with dynamic network topologies due to membership changes or mobility. Typically, a host in P2P and MANET also serves as a router, and employs some flooding-based routing protocols.

The common characteristics shared by P2P and MANET also lead to the same fundamental challenge, that is, how to provide connectivity in a completely decentralized environment. Thus, we are motivated to study the convergence of the two overlay network technologies in terms of the design goals and principles of their routing protocols.

Previously, the P2P and MANET research communities have been working largely in isolation, while facing many common issues like self-organization and decentralization. We argue that it is a promising research direction to bring the two communities together to merge the techniques used in the two areas and perhaps discover unified tricks for the convergence of the two overlay network technologies. As a supporting example, in this paper, we sketch an evolving architecture developed as part of the *Ad Hoc Networking* project at TUCS towards an integrated architecture for peer-to-peer and ad hoc overlay network applications.

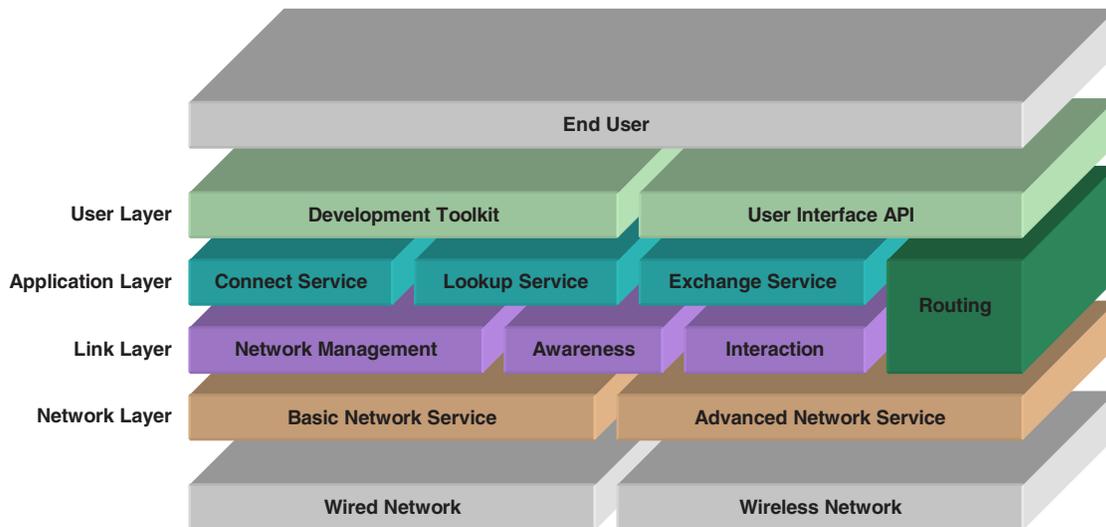


Figure 1: MIN Architecture

The remainder of the paper is organized as follows. We first discuss the research problems of P2P and MANET architectures in the next section. In Section 3, we describe the integrated architecture of P2P and MANET. Section 4 concludes the paper with future research efforts.

## 2. Research Problems

Today's networks are dependent on wired or wireless infrastructure. This dependence renders the networks vulnerable to disasters and attacks against the fixed infrastructure that supports them. Disasters like floods and earthquakes, as well as wars and terror strikes, can damage or shutdown the whole network. Thus a state-of-the-art research direction of nowadays network is on connectivity.

A network architecture that satisfies the above scenario will be radically different from the current existing network architectures since it cannot rely on a fix infrastructure and dedicated servers. Recent work on P2P overlay networks [5], [6], [7] offers a self-organizing substrate for decentralized network applications. Our general approach is to build a structured P2P overlay with existing technologies upon the basic connectivity provided by MANET in the absence of a dedicated server infrastructure. However, an important challenge is that existing P2P overlay protocols were designed for the Internet, which is a very different environment than MANET. The unique characteristics of this emerging class of networks calls for novel architectures. We present the key challenges as a set of research problems.

- **Self-organizing infrastructure.** Wired networks rely on a fixed infrastructure consisting of routers and DHCP and DNS servers. Any damage or interfere of the server will probably make the whole network out of service. Emerging P2P technologies promise to support self-organizing infrastructure, but these technologies are not directly applicable to the ad hoc wireless environments [3], because they are originally designed for the Internet with constantly stationary nodes, where as nodes are arbitrarily moving in MANET.
- **Decentralized service.** Existing networks depends on dedicated servers providing centralized basic network services like naming, authentication and timing etc. For instance, conventionally there are DHCP and DNS services in a typical network, while supporting this kind of critical network services is beyond the capability of existing P2P networks. Our approach is to build foundations from P2P system, but take advantages of the hierarchical overlay structure contributed by MANET to provide decentralized network services.
- **Integrated routing.** Integrating a P2P routing protocol into a MANET protocol is difficult. P2P overlays in the Internet rely on the IP routing mechanism which is actually application-level routing, while such kind of routing is usually carried out in link-level in MANET [4], [8], [16]. Although typical flooding and multi-hop routing protocols in MANET are peer-to-peer in natural, P2P routing protocols are not directly applicable in MANET due to the namespace problem.

### 3. The Integrated Architecture

We propose an evolving architecture which is able to provide network connectivity in a decentralized fashion and use self-organizing infrastructures to improve availability of today's network. In this architecture, ad hoc wireless networks can be combined with infrastructure-based networks through ad hoc communications between them. Once basic connectivity is established, hosts could self-organize and cooperatively provide network services that are normally provided by infrastructure servers.

Figure 1 shows the preliminary architecture of a subsystem we are building called the *MIN* that is aimed at addressing a subset of the problems listed in Section 2. The *MIN* is being built on top of an application-level P2P overlay over a link-level MANET, but the architecture is not specific to the implementation environment. We have chosen to focus on two issues, self-organizing infrastructure and integrated routing, which we believe to be fundamental. We feel that decentralized service could be elevated to a higher level of programming abstraction than typical one.

#### 3.1 Application Layer

The *MIN* architecture provides an abstract layer called application layer. This layer is mostly a structured P2P overlay, which is illustrated in Fig. 2.

Previously [1], we have specified a P2P overlay structure in OO-action systems [14]. As an example, you can see the specification of a host.

```

HOST = | [
  attr
    connected := false;
    keyword := NULL;
    target := NULL
  obj
    c: ConnectService;
    l: LookupService;
    e: ExchangeService
  meth
    SetKeyword(k) = Keyword := k;
    SetTarget(t) = target := t
  init
    c := new(ConnectService);
    l := new(LookupService);
    e := new(ExchangeService)
  do
    NOT(connected) →
      connected := c.Connect()
  [] connected AND keyword ≠ NULL →
    target := l.Search(keyword);

```

```

keyword := NULL
[] connected AND target ≠ NULL →
  e.Exchange(target);
  target := NULL
od
] |

```

As shown in this specification, three key services are identified in this layer: connect service, lookup service and exchange service [12].

- **Connect service.** A host connects itself to the P2P overlay by establishing a connection with another host currently on the network, and this kind of connection is passed around recursively.
- **Lookup service.** Once a host is connected to the P2P overlay, i.e. it has announced its existence to other members of the P2P overlay, it can then lookup the contents of the P2P overlay. Lookup requests are transmitted in a decentralized manner. One host sends a lookup request to its neighbors, which in turn pass the request along to their neighbors, and so on. Once a host in the P2P overlay has a match, it transmits the hit information back through all the intermediate hosts in the pathway towards the requesting host.
- **Exchange service.** The exchange service can be evoked in an either aggressive or passive manner. Due to the nature of P2P overlay, data are exchanged out-of-network, i.e. a direct exchange between the source and target hosts. Data are never transferred over the P2P overlay.

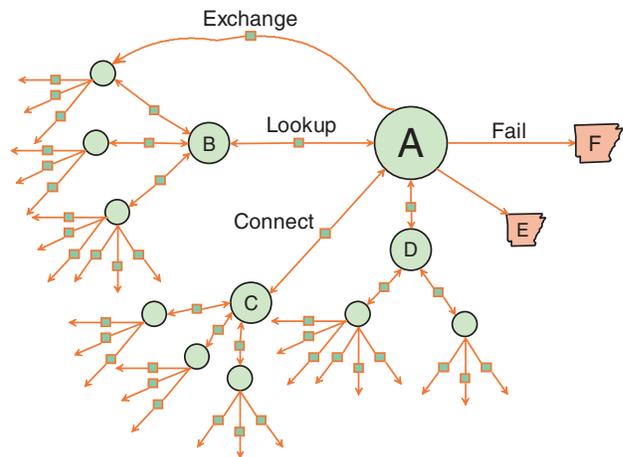


Figure 2: P2P Overlay

#### 3.2 Link Layer

To support ad hoc networking, the *MIN* architecture provides a link layer that allows application-level connections to result in connections to the appropriate logical links, which are either real wired and wireless links or virtual links among hosts. There are three key components in this layer: network management, awareness and interaction.

Previously [9], we have specified this layer in the B method [10]. As an example, you can see the specification of the general context of ad hoc networking environments.

**MACHINE** *AdHocNet*

**SETS**

*NODES*;  
*CommMSG* = { *commMsg*; *routeError* };  
*RouteMSG* = { *routeReq*, *routeRep* }

**CONSTANTS**

*myID*

**PROPERTIES**

*myID* ∈ *NODES*

**END**

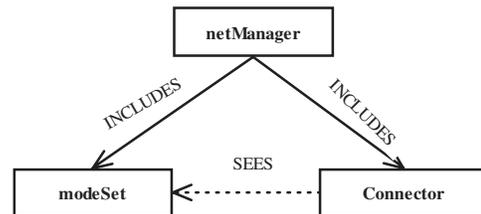
In the *AdHocNet* specification, there are three sets: *NODES*, *CommMSG* and *RouteMSG*. They are defined as set of nodes, set of communication messages, and set of routing messages in ad hoc networks. The *CommMSG* has two different elements: *commMsg* and *routeError*. The *commMsg* is used for the communication between nodes. The *routeError* is used when a route is broken. The *RouteMSG* consists of two kinds of routing messages: *routeReq*(route request) and *routeRep*(route reply), which are used in the Awareness component to detect remote nodes. The node's ID *myID* is a constant in the *AdHocNet*, which is an important property of *NODES* to identify a node in networks.

### 3.2.1 Network Management

Network management is the manager of node connections, which is an important aspect for the MANET design. In general, we consider not only mobility, but also restorability of networking. With this component, a host should be able to set the mode of the node, form, join or leave a network, and manage its connections. As a host is moving arbitrarily, disconnections may happen at any time due to the limited radio transmission range. In order to keep the network working, it is necessary to update the network topology

periodically. Moreover, in order to form a self-organizing network, and support multi-hop routing in forwarding packages, it is necessary to have the network manager in every host.

In the specification of network management, there are three components: *netManager*, *modeSet* and *Connector*. The relationship of components is shown in Fig. 3.



**Figure 3: Network Management**

The *netManager* includes *modeSet* and *Connector*, and uses their operations. In ad hoc networking, it manages activities of the system and updates connections to neighbor nodes. There are two modes that can be set in *modeSet*, either discoverable or non-discoverable. When a node is in discoverable mode, it can be discovered by other nodes in the network. Otherwise, it cannot be discovered if it is in non-discoverable mode. The *Connector* is used when a node wants to connect or disconnect its neighbors. For instance, in the network setup stage, a node can join into networks by connecting its neighbor nodes which are in discoverable mode. Once the connections are ready, the network is established. When shutting down the system, the *netManager* disconnects all the connections to neighbor nodes and sets the node into non-discoverable mode.

### 3.2.2 Awareness

Awareness of mobile computing is used to sense a certain environment in order to present and update context of mobile systems [11]. In our system, we focus on detecting local and remote nodes and processing incoming messages. The awareness of our system is divided into two parts.

- **Node awareness.** There are two kinds of awareness of node detections: local awareness – system can detect local nodes within the radio transmission range; remote awareness – system should also be able to detect a friendly remote node whose ID is already known. In local awareness, a node detects its neighbor nodes and the detected nodes will be connected and used to update topologies in Network Management. In remote awareness, a node tries to find out friendly

remote nodes with known IDs and possible routes for communication.

- Message awareness.** In message awareness, a node processes incoming messages according to the format of data packets. There are two kinds of messages in networking: communication message and routing message. In our specification, the communication message is used for communication between nodes, and there are two types of routing message: route request message and route reply message. Due to the different types of messages, the processing will be different. As shown in Fig. 4, if the received message is a communication message, the system will check the packet head, and receive or forward this message depending on the next hop ID on the route. In case this ID is unrecognizable, the system will report a broken route. If the incoming message is a routing message, the system will process this message according to the routing protocols in our system.

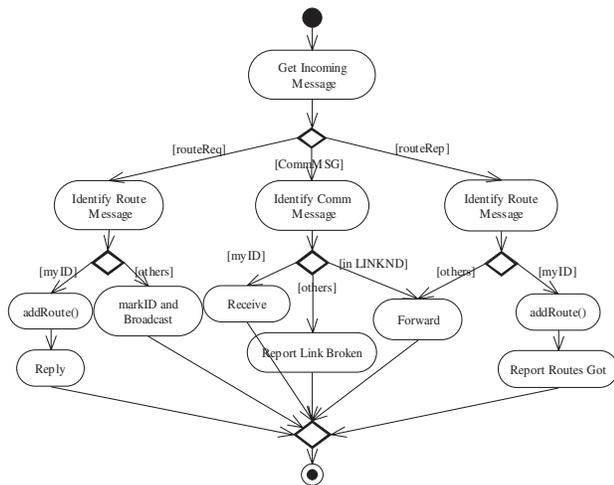


Figure 4: Incoming Message Processing

### 3.2.3 Interaction

Interaction mainly concerns communication links between nodes. We consider an opening session for interactive communication between nodes. In such a session, the source and destination nodes can send and receive messages and update routing information for communication.

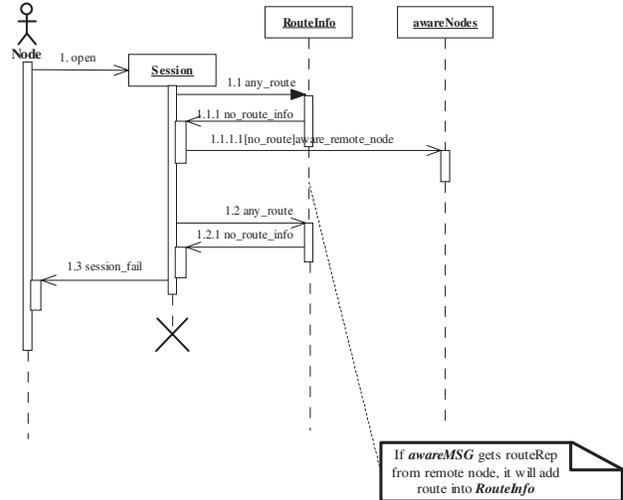
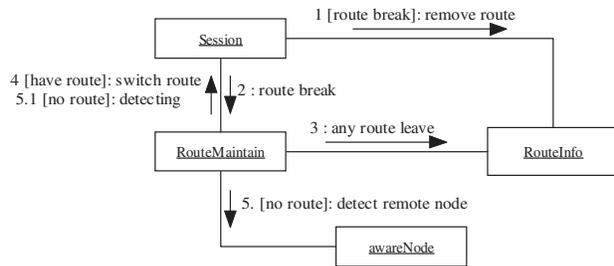


Figure 5: Opening Session

As shown in Fig. 5, when the system opens such a session and starts interactive communication, the source node will select a route from the routing table or detect a new route to reach the destination node. If there is no available route or the destination node is not detected in the network, the opening session fails and a failure message is sent back to the source node. In successful case, once a route is available, a communication session between the source node and destination node is created and the interactive communication starts.

In the interactive communication, topologies might be changed and it will lead to route breaks or changes. Thus the route maintenance and recovery are needed for interactive communication. Figure 6 shows how the route is recovered when the system knows that the route is broken. In our design, it is assumed that multiple routes discovery protocols are used. For example, when source node S is communicating with destination node D, S sends data packets to D along with the selected route. During their communication, if S gets to know that the communication route is broken, S doesn't need to rediscover a new route immediately because S might have detected several routes in the previous discovery. It can then choose another available route and replace the broken one. If none of the routes reaches to the destination, the system will start route discovery again.



**Figure 6: Route Recovery**

### 3.3 Integrated Routing

The primary challenge with using a P2P routing protocol in MANET is the fact that P2P overlays in the wired Internet rely on the IP routing infrastructure to perform hop-by-hop routing between neighbor nodes in the overlay. Thus the key problem in the integration is that P2P overlay routing protocols run in a logical namespace but MANET routing protocols run in a physical namespace. A possible solution to the integration is to build a one-to-one mapping between the IP address of the mobile nodes and their node IDs in the namespace, and replace the routing table entries which used to store IP addresses with source routes.

For instance, to integrate a Gnutella-like [12] P2P protocol into a DSR-like [15] MANET protocol, unique node IDs are first assigned to nodes in a MANET as is done in P2P overlay on top of the Internet. Node IDs can be generated by hashing the IP addresses of the hosts using collision-resistant hashing functions like SHA-1 [13], thus obtaining a unique node ID for each node in the network. The mobile nodes in the ad hoc network can then form a P2P overlay in the same fashion as in the Internet. Nodes can handle join, leave and fail actions in a similar way as before. The structure of the routing states is also similar as before, with one exception: the routing table stores the source route to reach the destination node ID, not just a simple IP address. To route a data packet, a message key is first generated by hashing the destination IP address, and then the message is routed in the overlay similarly to in the overlay on top of the Internet. The only difference is that each overlay hop in ad hoc networks is a multi-hop source route, while each overlay hop in the Internet is a multi-hop IP route.

### 4. Conclusions and Future Work

The main contribution of this work is that it proposes a novel architecture, integrating P2P and MANET technologies together, to reduce the dependence of networking on wired and wireless infrastructure, thus

extending the reachability of nowadays networks and increasing their resilience to disasters and attacks. Another contribution of this work is that it is the first architecture-centric approach for the construction of overlay network applications that allows us to define a unified networking environment, taking advantages from both P2P and MANET technologies.

The work presented in this paper is in its early stages. At present we are evaluating the initial version of the *MIN* framework through analysis, simulation, and a prototype implementation. In the formal respect, a complete formal specification of the architecture is being underway. As a future work, in particular, performance modeling and evaluation of integrating P2P and MANET routing protocols will be undertaken. A further future work of our research is to implement a middleware in this integrated architecture for use by developers of overlay network applications.

### Acknowledgements

The authors are grateful for Jincheng Ni and Nayyar Iqbal's contribution to this project.

### References

- [1] L. Yan and K. Sere, *Stepwise Development of Peer-to-Peer Systems*, Proceedings of the 6<sup>th</sup> International Workshop in Formal Methods (IWFM'03), Dublin, Ireland, July 2003. Electronic Workshops in Computing (eWiC), British Computer Society (BCS).
- [2] Y. C. Hu, S. M. Das, and H. Pucha, *Exploiting the synergy between peer-to-peer and mobile ad hoc networks*. In Proceedings of HotOS-IX: 9<sup>th</sup> Workshop on Hot Topics in Operating Systems, May 2003.
- [3] C. Blake and R. Rodrigues, *High availability, scalable storage, dynamic peer network: Pick two*. In Proc. HotOS IX, Kauai, Hawaii, May 2003.
- [4] M. Castro, P. Druschel, A. -M. Kermarrec, and A. Rowstron, *SCRIBE: A large-scale and decentralized application-level multicast infrastructure*. IEEE JSAC, 20(8), Oct. 2002.
- [5] A. Muthitacharoen, R. Morris, T. Gil, and B. Chen, *Ivy: A read/write peer-to-peer file system*. In Proc. of the 5<sup>th</sup> Symposium on Operating System Design and Implementation (OSDI 2002), Boston, MA, December 2002.

[6] S. Zhuang, B. Zhao, A. Joseph, R. Katz, and J. Kubiatiowicz, *Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination*. In NOSSDAV, June 2001.

[7] P. F. Tsuchiya, *Landmark hierarchy: a new hierarchy for routing in very large networks*. In Proc. of MobiHOC, Stanford, CA, Aug. 2000.

[8] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, *Application-level multicast using content-addressable networks*, In NGC, Nov. 2001.

[9] L. Yan, J. Ni and K. Sere, *Towards a Systematic Design for Ad hoc Network Applications*, Proceedings of the 15<sup>th</sup> Nordic Workshop on Programming Theory (NWPT'03), Oct. 2003.

[10] E. Sekerinski and K. Sere (Eds), *Program Development by Refinement: Case Studies Using the B Method*, Springer-Verlag, 1999.

[11] T. Selker and W. Burleson, *Context-aware Design and Interaction in Computer System*, IBM Systems Journal, Vol. 39, No. 3 & 4, 2000.

[12] I. Ivkovic, *Improving Gnutella Protocol: Protocol Analysis and Research Proposals*, Technical reports, LimeWire LLC, 2001.

[13] FIPS 180-1, *Secure Hash Standard*, Technical Report Publication 180-1, Federal Information Processing Standard (FIPS), NIST, US Department of Commerce, Washington D.C., April 1995.

[14] M. Bonsangue, J. N. Kok and K. Sere, *An approach to object-orientation in action systems*, Proceedings of Mathematics of Program Construction (MPC'98), Martstrand, Sweden, June 1998. LNCS 1422, Springer Verlag.

[15] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. In T. Imielinski and H. Korth (Eds), *Mobile Computing*, chap. 5, p. 153-181, Kluwer Academic Publishers, 1996.

[16] W. Adjie-Winoto, E. Schwartz and H. Balakrishnan, *An Architecture for Intentional Name Resolution and Application-level Routing*. MIT Technical Reports, TR-775, 1999.