

Mobile Chedar - A Peer-to-Peer Middleware for Mobile Devices

Kotilainen N.^{*}, Weber M., Vapa M.⁺ and Vuori J.
University of Jyväskylä, P.O.Box 35 (Agora), 40014 University of Jyväskylä
[npkotila, mweber, mikvapa, mimic]@jyu.fi

Abstract

This paper presents the Mobile Chedar Peer-to-Peer middleware for mobile peer-to-peer applications. The middleware is an extension to the Chedar peer-to-peer network allowing mobile devices to access the Chedar network and also to communicate with other Mobile Chedar peers. Currently, Mobile Chedar uses Bluetooth to connect to Chedar gateway peers, because Bluetooth is now the most widespread short-range radio technology in mobile phones. We also introduce in this paper one example of a Mobile Peer-to-Peer application for cooperative lecture notes taking, which is based on Mobile Chedar.

1. Introduction

Coulouris et al. [3] define distributed system as a system in which components located at networked computers communicate and coordinate their actions only by message passing. Peer-to-Peer networks (P2P) are an instance of distributed systems.

P2P networks allow the sharing of resources over the Internet. The resources can be for example, computing power, storage space, network bandwidth, printers etc. Another main feature of P2P networks is that all the tasks and responsibilities for managing the network are shared between the peers. This means that there is no single control entity responsible for providing the services.

Middleware provides an application programming interface (API) for accessing message passing functionalities and other common services needed in distributed systems. The main benefit is that by using middleware programmers can speed up the development of applications because the features needed for distribution are already provided by the middleware.

This paper describes one implementation of a mobile peer-to-peer (MP2P) middleware enabling information

sharing in a mobile environment. The proposed middleware is an extension to a non-mobile Chedar P2P network. In the following sections we discuss the related work on MP2P, introduce the Chedar P2P middleware, the Mobile Chedar MP2P middleware with an API for MP2P applications and finally present a co-operative learning application using the Mobile Chedar.

2. Related Work

Proem [5] is a mobile middleware providing solution for developing and deploying applications for mobile ad hoc networks. In Proem, middleware is responsible for presence and discovery services as well as being an identity, data space and community manager. Proem has been designed for mobile peers in ad hoc networks whereas in Mobile Chedar also peers with fixed P2P network connections are supported. The current prototype of Proem uses Wireless Local Area Network (WLAN) for communication and has been implemented using Java.

7DS [12] is a Java based data prefetching tool for mobile devices. It allows mobile users to advertise data items of their mobile devices and to query other user's data items through WLAN connections. 7DS works only on IP networks and is designed for disseminating rather static content. IP multicast is used for querying the peers.

XMIDDLE [10] is a reflective middleware enabling transparent sharing of XML documents between mobile peers. XMIDDLE does not use a fixed P2P infrastructure and therefore differs from our approach. Because data structure consists of XML trees, modifications to the branches of XML tree are fine-grain for example compared to modification of files. In Mobile Chedar, resource queries are matched using XPath expressions as in XMIDDLE, but modifications to the data stream are not guaranteed to have the same order in all peers. XMIDDLE solves the problem by allowing user to resolve the update conflicts. The current XMIDDLE prototype is based on WLAN and has been implemented using Java.

^{*} The work of N. Kotilainen is supported by Innovations in Business, Communication and Technology (InBCT) project.

⁺ The work of M. Vapa is supported by Graduate School in Electronics, Telecommunications and Automation (GETA).

MOBY [4] is a service network enabling access to services on wide area networks. The framework is built using Jini and Jini Technology Surrogate Architecture Specification. MOBY's approach seems to be the closest to ours. However, there are some differences in the design choices. In MOBY resources are registered to Jini Lookup Service, which is located in the local area network. In Mobile Chedar the mobile peers store their resources and no registration of resources to external server is needed. MOBY's P2P network is based on super-peer architecture i.e. the network is divided into domains by Mnode super-peers whereas in Chedar all peers are equal. Communication between Mnodes is handled using UDP, but in Chedar network connections are established using TCP. MOBY uses IP addresses to identify peers and Mobile Chedar, relying on Bluetooth service discovery, does not require IP addressing of mobile devices. Resource discovery is handled in MOBY using expanding ring [9] between Mnodes while Chedar currently uses breadth-first search. In overall, MOBY is designed more like a fixed overlay, because the links between Mnodes are preconfigured compared to the autonomous overlay approach used in Chedar P2P network.

Mobile Chedar is an extension to existing peer-to-peer network and therefore differs from the MP2P software presented in the literature. The middleware provides mechanisms for data streaming which is unique feature among the considered related work. Also Mobile Chedar uses Bluetooth as a transmission technology in contrast to WLAN used in other studies.

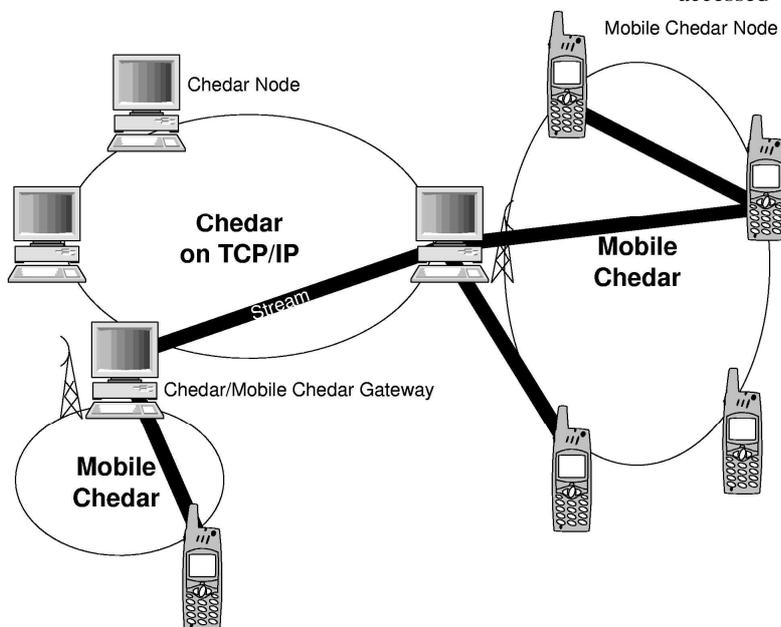


Figure 1. Stream delivery between Chedar and Mobile Chedar nodes

3. Mobile Chedar – A Mobile Peer-to-Peer Middleware

3.1. Chedar

Chedar (CHEap Distributed ARchitecture) is a peer-to-peer middleware designed for peer-to-peer applications. The goal of the Chedar software is to provide a convenient API for peer-to-peer application developers. For example Chedar can be used to locate unused resources in a computer network that could be used for a given purpose; one could thus locate idle computers with a given characteristics in order to run computationally intensive calculations. Chedar nodes maintain resources of different types: data (files), software (e.g. specific applications or operating systems) and hardware (e.g. CPU, printers and displays). It also provides platform independence and quick adaptation to new hardware. Chedar has been programmed with Java 2 Standard Edition and is currently being used for speeding up the computations of NeuroSearch resource discovery algorithm [13] with P2P Distributed Computing application (P2PDisCo) [6] and for studying distributed data fusion in peer-to-peer environment [11].

Each Chedar node is identified with a unique identifier (Chedar ID). The nodes maintain a database of locally available resources shared by the owner of the device. These resources can include for example files and databases, software running on the device that can be accessed or used by remote users, and hardware characteristics of the device. Also, remote resources discovered on the network can be added to the database combined with information about their owner identified by Chedar ID and meta-information about themselves. Meta-information can contain e.g. type and path for the files, name and version for applications or any useful description for the hardware depending on the application, which uses the information. The resource database is stored as an XML document using a specific DTD. This organization of data allows making rich and complex queries to the database in the form of XPath expressions.

3.2. Mobile Chedar

With the advent of mobile computing and the inherent peer-to-peer properties of mobile ad hoc networks, Chedar has been extended to the mobile platform as Mobile Chedar. It thus provides functionalities for

registering resources on the mobile device and for querying resources from other peers. Mobile Chedar is implemented using Java 2 Micro Edition (J2ME), which is suitable for mobile wireless devices and has spread widely among new mobile phone models. Mobile Chedar uses Bluetooth [1] as a transmission technology for connecting to other peers, because Bluetooth is the most common available short-range radio frequency wireless protocol stack on today's mobile devices.

Current Bluetooth implementations have a restriction that nodes can be connected to only one piconet at a time [8]. Therefore the only topology that is available for constructing Bluetooth network is star-shaped. One device functions as a master and others as slaves. In Mobile Chedar one node connected to piconet can be e.g. a workstation with a Bluetooth adapter and an Internet connection working as a Mobile Chedar/Chedar gateway node. Through the Internet connection it keeps contact with other Chedar nodes and through Bluetooth it can communicate with other Mobile Chedar peers.

A common use case for Mobile Chedar is the querying of a resource located on Chedar nodes or on other Mobile Chedar nodes through the gateway peer and then using the found resource. Chedar nodes can provide streamable resources to Mobile Chedar peers and depending on the device capabilities of the Mobile Chedar node they can subscribe to these streams. Currently text and picture are supported and in the future also audio and video streams will be supported also. Multiple peers can simultaneously subscribe to the same stream and after subscribing they also start to publish the stream as a resource. Therefore it is enough for a peer to locate one peer that provides the requested stream. This kind of a streaming is called end system multicasting [2]. Also, because streams are duplex, the data written to the streams by peers will be delivered to all other peers currently subscribed. Duplex streams can be realized by flooding all data inserted to the stream along the multicast tree to all other participants in the tree. However, the order of the data is not preserved and it is handled in a First-In-First-Out manner. Totally ordered delivery of data would require more complex implementation in this kind of environment [3]. Figure 1 illustrates a stream delivery between Chedar and Mobile Chedar peers.

Neighbor discovery is a prerequisite for resource queries. Since the nodes are able to communicate with each other using a wireless channel, it is easy to discover all nodes within range of the radio frequency transceiver using Bluetooth's Service Discovery Protocol (SDP). The discovery of resources is performed as one hop query, tagged with a unique Message-ID, to all the nodes within Bluetooth range. If the query arrives to a Chedar/Mobile Chedar gateway node, it checks whether the query has already been received: if not, it is forwarded to all of its Chedar neighbors with default time-to-live; otherwise, the

query is discarded. If the query message matches one of the resources owned by the node, the node replies to the neighbor from which it received the query with the same Message-ID as in the query message. The reply message then travels back to the originator of the query on the same path as the query traveled on. Once the location of the resource (or locations, if there exists multiple instances of the same resource in the network) is known, Mobile Chedar informs the application, which decides how to acquire or use the resource.

4. Mobile Chedar Application Programming Interface

Mobile Chedar provides the following API for MP2P applications:

| | |
|--|---|
| <i>register(String resourceidentifier)</i> | Adds a resource to the resource database. |
| <i>unregister(String resourceidentifier)</i> | Removes a resource from the resource database. |
| <i>connected()</i> | Checks if Mobile Chedar is connected to other Chedar nodes. |
| <i>query(String resourceidentifier)</i> | Executes a query. |
| <i>subscribe(Resource resource)</i> | Subscribes to the found resource. |
| <i>unsubscribe(Resource resource)</i> | Unsubscribes from a subscribed resource. |
| <i>send(Resource resource, Message data)</i> | Sends data to the subscribed resource. |

The MP2P applications must implement the following methods:

| | |
|---|---|
| <i>resourceFound(Resource resource)</i> | Informs the application when the query has located a matching resource. |
| <i>receive(Resource resource, Message data)</i> | Informs the application when new data has arrived to a subscribed resource. |

5. Mobile Peer-to-Peer Learning Environment

Mobile Peer-to-Peer Learning Environment (MP2PLE) [7] is designed for collaborative note taking during lectures as a test application for Chedar peer-to-peer network and Mobile Chedar middleware. The MP2PLE user interface contains a text area displaying the current state of notes and provides means for users to edit them. With MP2PLE, the

mobile device user may create a new stream for other participants to join or subscribe to an already existing one by executing a query. After subscription the user is allowed to modify any part of the notes by selecting a paragraph and submitting the changes. Whenever the data is being changed it is streamed to other participants subscribed to the same stream. At the moment, each user can be only subscribed to one stream at a time. The user interface of MP2PLE is shown in figure 2.

There are two common use cases for such kind of an application. Firstly, it serves as a personal note-taking tool to store lecture notes. Secondly, people who do not take notes can benefit from other user's notes, either during the lectures, or later, e.g. from home by accessing Mobile Chedar nodes through a gateway node.

MP2PLE has certain limitations in the current design. The tiny user interface is problematic and provides only primitive means to take notes e.g., pictures cannot be drawn and course presentation material cannot be integrated with MP2PLE. Also, taking lecture notes is difficult because of the small keypads in mobile phones. These limitations can only be overcome if larger screen sizes and more convenient input devices are being used.

Bluetooth does not allow multi-hop with current mobile phones because the device can only belong to one piconet at a time. To support multiple devices in a classroom one solution would be to use the approach presented in [8] and to equip Bluetooth base stations with two radio chips. They could, for example, be plugged into power supplies inside the classroom and equipped with Mobile Chedar middleware to function as relaying devices for queries. Another solution would be to use different transmission technology e.g., WLAN, which however is not yet available in low-cost mobile terminals.

6. Conclusion

Mobile peer-to-peer enables new kind of applications taking advantage of emerging short-range radio technologies and allowing collaborative resource sharing between peers. This paper describes one way to construct peer-to-peer networks with support for mobile devices and demonstrates the feasibility with a prototype implementation. The future work of Mobile Chedar and MP2PLE includes the support of audio and video streams and determining the feasibility of the approach with practical experiments.

7. References

- [1] Bluetooth Special Interest Group, "Bluetooth Core Specification v1.2", March 2004
- [2] Y. Chu, S. G. Rao, and H. Zhang, "A Case for End System Multicast", International conference on Measurement and modeling of computer systems, SIGMETRICS, Proceedings, 2000, pp. 1-12.
- [3] G. Coulouris, J. Dollimore, and T. Kindberg, "Distributed Systems – Concepts and Design", 3rd Edition, Addison-Wesley, 2001.
- [4] T. Horozov, A. Grama, V. Vasudevan, and S. Landis, "MOBY - A Mobile Peer-to-Peer Service and Data Network", International Conference on Parallel Processing, Proceedings, 18-21 August 2002, pp. 437-444.
- [5] G. Kortuem, "Proem: a middleware platform for mobile peer-to-peer computing", Mobile Computing and Communications Review, ACM, Volume: 6, Issue: 4, October 2002, pp. 62-64.

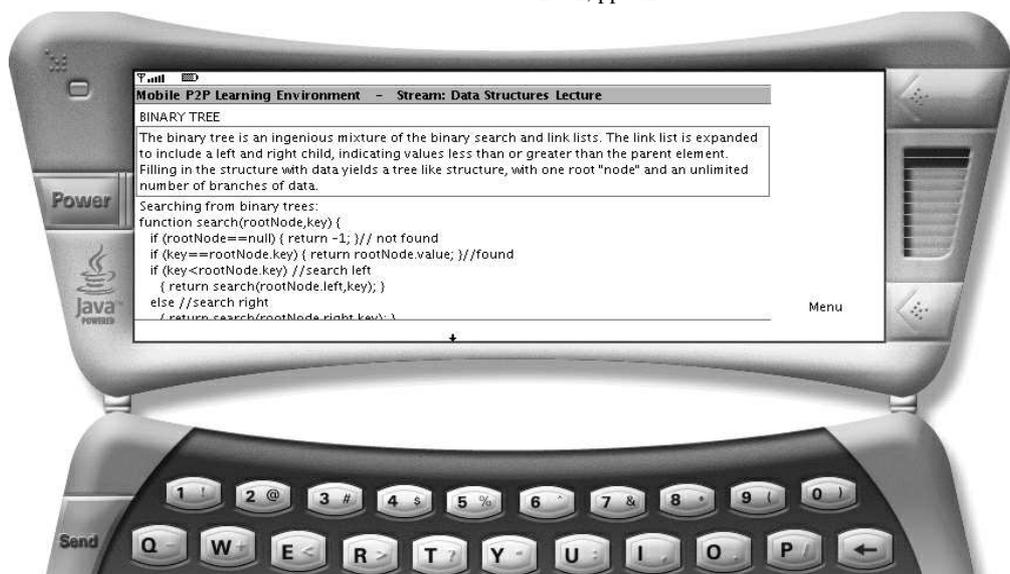


Figure 2. User Interface for Mobile Peer-to-Peer Learning Environment

[6] N. Kotilainen, M. Vapa, M. Weber, J. Töyrylä, and J. Vuori, "P2PDisCo – Java Distributed Computing For Workstations Using Cheddar Peer-to-Peer Middleware", University of Jyväskylä, 2004.

[7] J. Kurhinen, M. Vapa, M. Weber, N. Kotilainen, and J. Vuori, "Short Range Wireless P2P for Co-Operative Learning", The 3rd International Conference on Emerging Technologies and Applications, Proceedings, 16-18 October 2004, pp. 141-145.

[8] M. Leopold, M. B. Dydensborg, and P. Bonnet, "Bluetooth and Sensor Networks: A Reality Check", The First International Conference on Embedded Networked Sensor Systems, Proceedings, November 2003, pp. 103-113.

[9] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks", Proceedings of the 16th International Conference on Supercomputing, ACM Press, 2002, pp. 84-95.

[10] C. Mascolo, L. Capra, S. Zachariadis, and W. Emmerich, "XMIDDLE: A Data-Sharing Middleware for Mobile Computing", International Journal on Wireless Personal Communications, Kluwer Academic Publisher, Volume: 21, Issue: 1, April 2002, pp. 77-103.

[11] S. Nazarko, "Evaluation of the data fusion methods using Kalman filtering and Transferable Belief model", Master's Thesis, University of Jyväskylä, 2002.

[12] M. Papadopouli, and H. Schulzrinne, "Design and Implementation of a Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users", First New York Metro Area Networking Workshop, IBM T. J. Watson Research Center, Hawthorne, New York, 12 March 2002.

[13] M. Vapa, N. Kotilainen, A. Auvinen, H. Kainulainen, and J. Vuori, "Resource Discovery in P2P Networks Using Evolutionary Neural Networks", International Conference on Advances in Intelligent Systems – Theory and Applications, Proceedings, IEEE Press, Luxembourg, 15-18 November 2004.