

# LightPeers - A Framework Supporting Nomadic Learning in Mixed Environments with Mobile Ad Hoc Networking

Bent Guldbjerg Christensen  
Center for InteractiveSpaces, ISIS Katrinebjerg  
Aabogade 34, 8200 Aarhus N, Denmark  
bentor@daimi.au.dk

## ABSTRACT

This paper describes the LightPeer framework, which is a generic P2P framework inspired from work with school children who could benefit from a P2P based infrastructure in their project oriented field work. The focus for developing the LightPeers framework has been on the following set of objectives: to define a lightweight framework, that can run on devices with limited resources, introduce the use of a community tagging mechanism as a means for loosely describing applications and resources, and finally to extract the notion of shared sensors in a dedicated logical framework layer.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

## Keywords

LightPeers, peer-to-peer, P2P, tagging, sensor

## 1. INTRODUCTION

The current interest and popularity of peer-to-peer (P2P) systems for common use as well as in research started out with file sharing applications in the years 1999 and 2000 such as Napster and Gnutella [10, 11]. Although file sharing probably still today is the best-known application of P2P systems, applications of P2P systems go beyond that, and can be divided into at least the following four categories: Information sharing, Communication, Collaboration, and Distributed computation.

- **Information sharing** refers to applications where peers share information. Examples of such systems are: Napster, Freenet, and BitTorrent [10, 7, 8].
- **Communication** is crucial when collaborating, and the communication needs only to be directly between peers. Among such systems exist: ICQ (ICQ server maps from ICQ-specific addresses to IP addresses, the communication is P2P) and Skype [3, 4].

- **Collaboration** between humans is in essence P2P, and this is reflected in system. Systems of this kind include: Groove and Distributed Knight [2, 9].
- **Distributed Computation** is where otherwise unused computing power and network bandwidth at the "edge" of the Internet is organised in a P2P network and each peer is used for computing a small fractions of a large problem space. E.g.: The Grid and GreenTea [5, 1].

The LightPeers framework presented in this paper can be used for applications in all four of these categories, but are developed especially to support applications for communication and collaboration. The primary domain for LightPeers applications is support of nomadic learning for pupils in public schools. From our studies conducted at the Center for Interactive Spaces<sup>1</sup> of public school activities [14, 13, 6] we see that learning becomes more project oriented and nomadic in the sense, that pupils spend their daily life as "nomads" in transit between many physical places ("oases") such as classrooms, labs, workshops, libraries, museums, the city, nature, clubs and at home. The aim of the LightPeers framework is to support nomadic and project oriented learning characterised by ongoing production of material anywhere with the devices at hand, and support ad hoc communication and collaboration among group members in mixed environments.

To appropriately support applications of the domain area, the following set of objectives has been selected as focus for the LightPeers framework development: to define a lightweight framework, that can run on devices with limited resources, introduce the use of a community tagging mechanism as a means for loosely describing applications and resources, and to extract the notion of shared sensors in a dedicated logical layer.

The rest of the papers is organised as follows. In section 2 the LightPeers framework layers and components are presented. In section 3 the datamodel are described, followed by the plan for future work in section 4, and the paper is concluded in section 5.

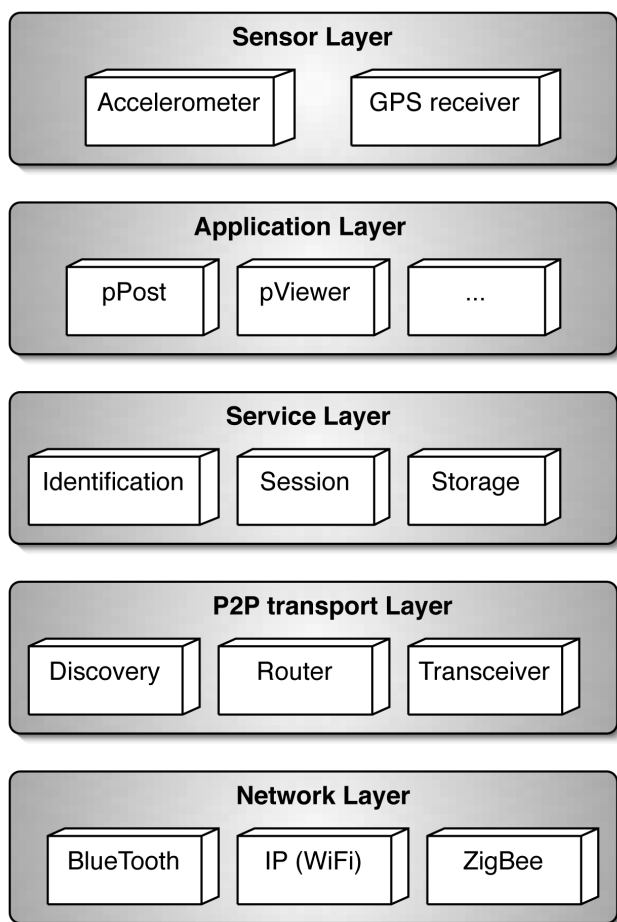
## 2. THE LIGHTPEERS FRAMEWORK

The following description of the LightPeers framework takes a starting point in describing what a single device in a LightPeers network should implement to participate.

The LightPeers framework is divided into the five layers:

<sup>1</sup><http://www.interactivespaces.net>

Network Layer, P2P transport Layer, Service Layer, Application Layer, and Sensor Layer as seen in Figure 1.



**Figure 1: The five LightPeers Framework Layers: Network, P2P transport, Service, Application, and Sensor layers.**

The Network Layer contains components for handling the fundamental communication technology of the device. This could be all sorts of transceivers from traditional WiFi cards to BlueTooth<sup>2</sup> or ZigBee<sup>3</sup> hardware to more specialised and product specific hardware. The functionality that should be supported by each component in the Network Layer is to be able to transmit and receive data packets (messages). Thus a mechanism to split data packets into appropriate sized units according to hardware limitations should be supported at this layer. E.g. this could be the case for communication devices only offering the medium access control layer, as is the case for some ZigBee ready devices.

The P2P transport Layer encapsulates the basic components to support an ad hoc networking structure and data exchange. The key component in the framework is the *Discovery* component that facilitates devices running the LightPeers framework to discover each other. The component is responsible of maintaining an updated representation of the LightPeers network the host device participates in. This is

<sup>2</sup><http://www.bluetooth.org>

<sup>3</sup><http://www.zigbee.org>

an especially important task as the framework is intended to be used with highly mobile devices, thus the network topology is likely to be of a very dynamic nature. The mobile devices are potentially less reliable as participants in a network than traditional stationary desktop computers or servers due to several reasons.

- The mobile devices are likely to be physically moved out of network reach.
- Power supply for mobile devices is indeed a limited resource, and could be used up untimely.
- As the devices typically both are handheld and operated by the user at the same time, they are in danger of being dropped or otherwise roughly handled.

The *Discovery* component should aggregate the discovery mechanisms for all of the Network Layer components. This may result in a situation where a certain device can be contacted through several means of communication media, e.g., through WiFi and ZigBee at the same time. In such a case the *Router* component decides, based on a simple priority metric, which communication media to send data packets through. The routing metric for this setup could be extended to include parameters such as stability of communication media and power usage. The *Router* component is tightly coupled to the *Transceiver* component. When the network topology is established by the *Discovery* component, the *Transceiver* component can be used as a single point of transmitting and receiving data packets for a LightPeers device. The *Transceiver* component includes message queues for incoming and outgoing data packets, and a newly arrived data packet in one of these queues is examined by the *Router* component for determining its next hop (if not already at destination) and priority.

The third layer is the Service Layer in which common service components are logically grouped. One of these, is the *Identification* component that is used to retrieve information about peers on a discovered device. The notion of a peer is here used as a digital identity representing a person or a software agent. In practice a mobile device is often also a personal device, thus a one-to-one correspondance between peer and device is the result. However, there is no conceptual restriction in the framework for a peer to be present at several devices, or a device having several peers present at the same time. A session is a group with one or more peers participating in a shared activity. The *Session* component offers state information about sessions that peers on a device are participating in. Notice, that the *Identification* and the *Session* components only handles information about local peers and sessions with local peers participating. The data actually sent from and to the *Identification* and *Session* components are detailed in the datamodel section further down. The last component of the Service Layer is the *Storage* component, which is dedicated to handle shared storage among peers participating in a session, thus a session can have a session store where each peer can contribute appropriately.

The Application Layer is the fourth layer of the framework and is where the actual applications are logically placed. Thus, a LightPeers programmer would have access to the components of the Service Layer below and the Sensor Layer above. This is realised in the reference implementation by a

skeleton application (pFoo) having references for these components from the start.

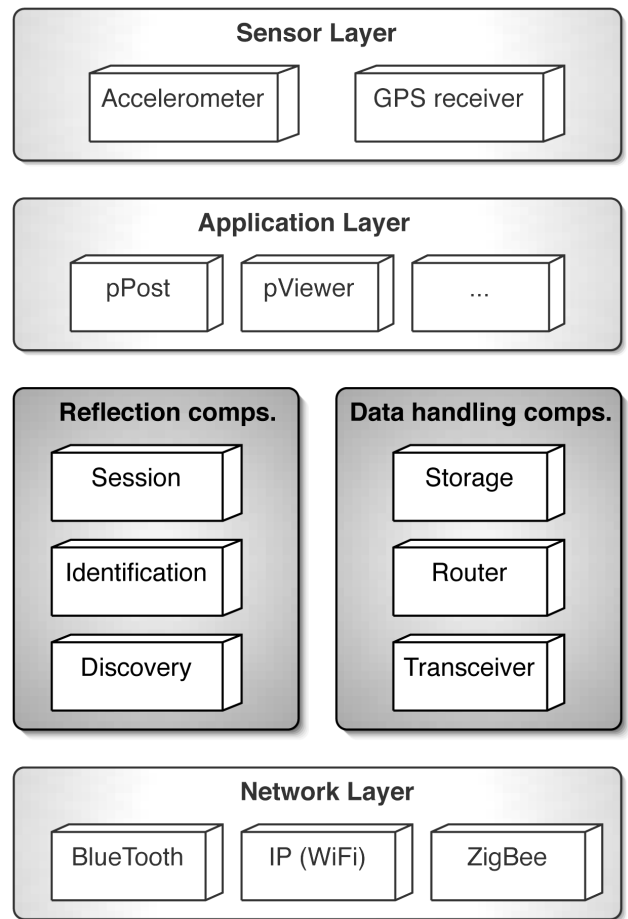
The final and fifth layer, the Sensor Layer, is where components for accessing sensor equipment are grouped. The reason to encapsulate sensors with components is to allow sharing of the equipment among applications on the same device, and to facilitate sharing of the sensor as a resource among peers participating in the same session. Each sensor component implements an interface that requires the component to implement: a simple data pull mechanism, and a notification mechanism when sensed data is available. Sensors of special interest for use on mobile devices is the movement and motion sensors, because this information can be used actively to, e.g., tune the discovery protocol.

An alternative perspective on the basic components in the P2P transport Layer and the Service Layer can be useful to distinguish their functionality in the framework. One grouping of components handles how to: discover LightPeer devices, identify peers on devices, and exchange information about session state. These components handle information primarily used to maintain the ad hoc network infrastructure by examining their LightPeer context and offer information about their own state. In Object Oriented languages like Object C, C++, and Java this property of examining objects at runtime is called Object Reflection. Inspired by this, the group of components is called Reflection components, as can be seen in Figure 2. The other grouping contains components for ordinary data handling, i.e., transmitting and receiving data packets, routing data packets, and storing data. This group of components is therefore called Data handling components.

## 2.1 The LightPeers Datamodel

In the reference implementation the datamodel used to represent, e.g., a peer or a session, can be expressed as XML structures. In fact each object in the datamodel implements an interface with the two methods *toXML* and *fromXML* that allow serialization and deserialization of an object state which are useful when transmitting an object between devices. The choice of using XML structured data for transmissions may seem in conflict with the idea of having a lightweight framework because of the overhead of ASCII characters being sent and the need of XML parsing at the destination. The reason for this choice is foremost because XML offers a simple way of expressing possibly complex data structures, and secondly because debugging of data flow is more accessible in the development phase. However, when need be, a compact version of the transmission protocol in a binary format, can be expressed as a mapping to and from the XML based datamodel. The LightPeers datamodel is quite simple and describes the following three object types: peers, sessions, and messages. Notice, that the two first objects belong to Reflection components and the last one the Data handling components. The state of a peer which is serialized to XML and ready for transmission would look like the following example shown in Figure 3.

Notice, that a peer in the LightPeers framework is a digital identity representing a person or a software agent. A peer is modelled with a **guid** element for unique identification, a **name** element for easy recognition, and a collection of device endpoints (**endpoint-guid**) where the peer can be contacted. The endpoints are referred to by their ids, which are again globally unique ids. This is the minimum required



**Figure 2: An alternative perspective on the P2P transport Layer and Service Layer where the components are grouped into Reflection components and Data handling components.**

information about a peer, but further optional descriptions may be provided, e.g., an icon, person age or phone number.

A LightPeers session conceptually consists of a group of participating peers, a set of applications used by the peers to communicate and share data, and a list of shared resources (often hardware based) which are offered by endpoints at the devices in use. An XML example of such a session is presented in Figure 4, where the session also has a **guid** element for unique identification, a **name** element for easy human recognition, and a **creator** element that refers to the owner peer of the session. An important feature of a LightPeers session is the possibility to use tags to describe applications and shared resources of peers participating in the session. The tags are used like freeform meta-data as known and used by the systems: Del.icio.us<sup>4</sup>, Flickr<sup>5</sup>, and CiteULike<sup>6</sup>. Furthermore, the session description is allowed to contain more detailed or application specific elements, e.g., a logo or a high score list. The session is a key concept in the LightPeers framework, because sessions define the

<sup>4</sup><http://del.icio.us/>

<sup>5</sup><http://www.flickr.org/>

<sup>6</sup><http://www.citeulike.org>

---

```

<?xml version="1.0"?>

<peer xmlns="http://lightpeers.daimi.au.dk">
  <name>foo</name>
  <guid>...</guid>

  <endpoints>
    <endpoint-guid>...</endpoint-guid>
    <endpoint-guid>...</endpoint-guid>
  </endpoints>
  ...
</peer>

```

---

**Figure 3: The state of a peer from the LightPeers datamodel serialized as an XML structure.**

context or environment of all peer collaboration.

Application specific data is packaged into LightPeers messages. The message format is structured as simple SOAP 1.2 [12] messages as seen in Figure 5 with SOAP Header and Body elements. As with the data exchanged by the Reflection components, the SOAP based message format are not restricted to be represented as XML structures [12][sec. 1.3]. In the Header element the participating endpoints, the recipients, the sender, the message id, and the expire-time of the message is listed. The message protocol is based on a simple graph coloring algorithm. The Body element contains the content of the message.

### 3. FUTURE WORK

The LightPeers framework presented in this paper is still work in progress, therefore the obvious next step, to evaluate the framework, is ahead of us. However, the plan for evaluation is twofold, as it is going to be evaluated: from a developer's perspective on how the framework facilitates application development, and from a user's or pupil's perspective on how the applications build upon the framework are supporting nomadic learning. A series of more technical performance tests like: average time of discovery and throughput of data over several hops, for a given hardware configuration is also planned to fine-tune framework configuration parameters.

For the pupil's perspective evaluation, a LightPeers kit including mobile phones, PDAs, tablets, and laptops all on which the LightPeers framework have been implemented, is compiled, and in collaboration with local public school teachers, the LightPeers kit is being used for ordinary project work for a period of two to three weeks in one or two classes. To measure the effect of using the LightPeers kit compared with using traditional tools on a set of parameters, a control group of pupils are used. Both the product of and the process during the pupils' project work is compared and evaluated.

Based on findings and experiences from performing tests with these two evaluations perspectives, the focus for further work on the LightPeers framework is planned.

### 4. CONCLUSION

In this paper, the LightPeers framework has been introduced as a framework for supporting applications for the do-

---

```

<?xml version="1.0"?>

<session xmlns="http://lightpeers.daimi.au.dk">

  <name>Fieldtrip session</name>
  <guid>...</guid>

  <creator>
    <peer>...</peer>
  </creator>

  <participants>
    <peer>...</peer>
    <peer>...</peer>
    ...
  </participants>

  <applications>
    <application>
      <guid>...</guid>
      <name>pPost</name>
      <tags>
        <tag>pPost</tag>
        <tag>IM</tag>
      </tags>
    </application>
    <application>
      <guid>...</guid>
      <name>HyConExplorer</name>
      <tags>
        <tag>HyConExplorer</tag>
        <tag>hypermedia</tag>
        <tag>browser</tag>
      </tags>
    </application>
  </applications>

  <shared resources>
    <resource>
      <tags>
        <tag>storage</tag>
      </tags>
      <endpoint>...</endpoint>
    </resource>
    <resource>
      <tags>
        <tag>accelerometer</tag>
        <tag>bluetooth</tag>
      </tags>
      <endpoint>...</endpoint>
    </resource>
    ...
  </shared resources>
  ...
</session>

```

---

**Figure 4: The state of a session from the LightPeers datamodel serialized as an XML structure.**

---

```

<?xml version="1.0"?>

<soap:Envelope xmlns:soap=\
"http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://lightpeers.daimi.au.dk">
  <soap:Header xmlns:lp=\
    "http://lightpeers.daimi.au.dk/message">

    <lp:session-endpoints>
      <lp:endpoint lp:visited="false">...</lp:endpoint>
      <lp:endpoint lp:visited="true">...</lp:endpoint>
    </lp:session-endpoints>

    <lp:recipients>
      <lp:peer lp:visited="false">...</lp:peer>
      <lp:peer lp:visited="false">...</lp:peer>
    </lp:recipients>

    <lp:sender>
      <lp:peer>...</lp:peer>
    </lp:sender>

    <lp:message-info>
      <lp:message-id>...</lp:message-id>
      <lp:expire-time>...</lp:expire-time>
    </lp:message-info>

  </soap:Header>
  <soap:Body xmlns:m=\
    "http://lightpeers.daimi.au.dk/pPost">

    <m:content>
      Hi Morten,
      I have added some nice pictures from the field
      trip to the session storage. See ya, Bent
    </m:content>

  </soap:Body>
</soap:Envelope>

```

---

**Figure 5: A LightPeers message produced from the pPost application.**

main of nomadic learning in mixed environments, i.e., learning where the kids may work in many different places in and outside the school buildings. Particularly, the framework development has been focused on three objectives:

- define a lightweight framework able to run on devices with limited resources,
- propose a freeform community tagging mechanism of applications and resources, and
- distill the handling of shared sensors in a dedicated logical layer of the framework.

Finally, the LightPeers datamodel for peers, sessions, and messages have been described, where the LightPeers messages is based on the SOAP standard.

## 5. REFERENCES

- [1] GreenTea. <http://www.greenteatech.com/>.
- [2] Groove - Virtual office. <http://www.groove.net/>.
- [3] ICQ. <http://www.icq.com/>.
- [4] Skype: Free internet telephony that just works. <http://www.skype.com>.
- [5] The Grid. <http://www.grid.org>.
- [6] C. Brodersen, B. G. Christensen, K. Grønbæk, C. Dindler, and B. Sundararajah. ebag: a ubiquitous web infrastructure for nomadic learning. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 298–306, New York, NY, USA, 2005. ACM Press.
- [7] I. Clarke. Freenet. 1999. <http://freenetproject.org/>.
- [8] B. Cohen. BitTorrent. 2002. <http://www.bittorrent.com/>.
- [9] C. Damm and K. Hansen. Distributing Knight — Using type-based publish/subscribe for building distributed collaboration tools. In *Proceedings of NWPER*, 2002.
- [10] S. Fanning. Napster. 2000. <http://www.napster.com>.
- [11] J. Frankel and T. Pepper. Gnutella. <http://www.gnutella.com>.
- [12] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. F. Nielsen. SOAP 1.2. W3C Recommendation, W3C, June 2003. <http://www.w3.org/TR/soap12-part1/>.
- [13] O. S. Iversen and C. Nielsen. Using digital cultural probes in design with children. Poster paper in the Proceedings of the Interaction Design and Children 2003 conference (IDC'03), 2003.
- [14] P. Nørregaard, J. Andersen, C. Dindler, J. Frich, O. S. Iversen, and C. Nielsen. Networking news—a method for engaging children actively in design. In *Proceedings of the 26<sup>th</sup> Information Systems Research Seminar in Scandinavia*, 2003.