2009 3rd IEEE International Conference on Digital Ecosystems and Technologies

Efficient Controlling of End-Devices in a JXTA-Based P2P Platform and its Application to Online Learning

Leonard Barolli[1], Member, IEEE, and Fatos Xhafa[2]

[1]Fukuoka Institute of Technology, 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan,
e-mail: barolli@fit.ac.jp
[2]Technical University of Catalonia, Jordi Girona 1-3, 08034 Barcelona, Spain, email: fatos@lsi.upc.edu

***Abstract*—The current Internet infrastructure is becoming each time more heterogeneous regarding computational resources. With the fast improvements in computational capabilities of devices, end-devices are becoming important components for the development of nowadays Internet applications. However, the integration of end-devices into Internet applications is a challenging task due to the intrinsic difficulties of current Internet architecture and protocols as well as computational limitations of end-devices. In this work, we address the issue of the efficient control of end-devices in a JXTA based P2P platform. We take advantage of P2P protocols in order to overcome the limitations of client/server architecture to enable the communication of peer nodes with devices despite of presence of NATs and firewalls. We consider as an end-device the SmartBox that is able to stimulate learners during their learning activity and thus increasing their learning efficiency and outcomes. We evaluate the proposed system by experimental study and show the usefulness of using SmartBox end-device in the development of e-learning applications.**

***Index Terms*—P2P Systems, JXTA library, JXTA-Overlay, Java Applications, End-device Control, SmartBox.**

I. Introduction

The Internet is growing every day and the performance of computers is significantly increased. However, the Internet architecture is based on Client/Server topology and can not use efficiently the client features. With appearance of new network technologies, new network devices and applications are appearing. Thus, it is very important to monitor and control the network devices via communication channels and exploit their capabilities for the everyday real life activities. However, in large scale networks such as Internet, it is very difficult to control the network devices, because of the security problems. Each network has its own security policy and the information should overcome firewalls, which are used for checking the information between private and public networks. The information is transmitted according to some decided rules and it is very difficult to change the network security policy. Therefore, recently many researchers are working on Peer-to-Peer (P2P) networks, which are able to overcome firewalls and other security devices without changing the network policy. The P2P architecture is thus very important for controlling end-devices in Wide Area Networks (WANs).

In P2P systems, the computational burden of the system can be distributed to peer nodes. Therefore, the users become themselves actors by sharing, contributing and controlling the resources of the systems. This characteristic makes P2P systems very interesting for the development of decentralized applications [1], [2]. However much research efforts are oriented towards overlays and file sharing. Certainly, today's P2P systems are far from being full featured, that is, systems that can integrate any connected device on the network, ranging from cell phones and wireless PDAs to PCs and servers. Recently, there has been an increasing interest in designing, implementing and deploying P2P networks that integrate mobile devices such as PDAs and end-devices. It should be noted that the integration of both mobile devices and end-devices into P2P platforms is still a difficult task given that there are not mature libraries that fully support their programming for P2P systems.

This work is motivated by the need to develop decentralized P2P approaches for the control of end-devices without changing the network policy. In this work, we present the implementation of main features of a JXTA-based P2P system (called JXTA-Overlay). We also show the implementation and the design of a smart box (SmartBox) environment which we consider as an end-device. The SmartBox is integrated with the JXTA-Overlay by using efficient message sending between peers in the system.

The structure of this paper is as follows. In Section II, we introduce the issue of integration of end-devices into P2P systems. In Section III we discuss the importance of using end-devices in advanced online learning systems. In Section IV, we give a short description of the main protocols of the JXTA library and describe the main features of the JXTA-Overlay platform. In Section V, we present the integration of the SmartBox end-device and its control in JXTA-Overlay. Experimental results with the SmartBox end-device are discussed in Section VI. We end the paper with some conclusions in Section VII.

II. Integration of End-Devices into P2P Systems

The high degree of heterogeneity of computational resources is a real challenge for the today's Internet applications. The great variety of computational resources ranging from servers, PC, laptops to hand-held and end-devices makes their integration very complex. Among other issues we could distinguish the difficulties of integration with current Internet architecture, the lack of a standard middleware that would facilitate and make transparent the programming task of the diverse computational devices and, not less importantly, the security issues.

Current research work is addressing the integration of hand-held and end devices in web applications, in Grid applications and P2P applications. The objective is to develop pervasive and ubiquitous applications using web, Grid, P2P, smart environments and sensor technologies.

In this section, we briefly discuss some research work in the literature on the use end-devices in P2P systems. Kumar et al. [3] presented a middleware for digital rights management in P2P networks. The authors considered P2P networks comprising different types of user devices. Charas [4] introduced the concept of local policy enforcement to define terminal centric control with the aim to develop a mobile architecture including ubiquitous end user devices. Hu [5] presented techniques for NAT traversal techniques and P2P applications. Another research work in this direction is reported in [6] where a NAT traversal for Pure P2P e-Learning system is proposed.

Considerable research efforts are currently being devoted to the extension of P2P networks with mobile devices such as PDAs. Iwata et al. [7] described a P2P platform implementation on mobile devices which enables mobile devices to behave autonomously as peers within ad hoc wireless network. A survey of the current technologies and standards on MIDP (Mobile Information Device Profile) devices and Java (J2ME) interfaces is presented by Tuisku [8] exploring the Mobile Internet and Grid technologies for new applications. Kubo et al. [9] developed a network monitoring system for a variety of devices using a hybrid P2P technology. Purvis et al. [10] presented a multi-agent approach for P2P applications on small portable devices. In particular, several studies are done for measuring the efficiency in P2P systems [11].

III. Use of End-Devices in Online Learning

Virtual campuses are nowadays a common approach widely used for online distance teaching and learning. In fact, this approach is used not only for open universities but also in semi-open teaching and learning courses taught by different institutions and organizations world-wide.

Most of the online applications that support distance teaching and learning are web-based. Due to the very fast development in web technologies as well as the emergence of new paradigms such as Grid, P2P and mobile computing, the online learning systems are currently undergoing important changes. The rationale behind these changes is to shift from the "*old*" paradigm of offering remotely teaching and learning supported through virtual classrooms, to the *new* paradigm of learning and teaching "anytime, anywhere". This new paradigm of distance teaching and learning is being possible due to the everyday increase of hand-held devices such as PDAs, mobile phones as well as many types of end-devices. By such variety of computational devices users are enabled to connect to Internet at anytime and anywhere.

The implementation of this new paradigm has certainly many benefits for online learning as compared to only web-based online applications. Among the most remarkable features of such new online learning applications using hand-held and end devices, we could distinguish:

- *Permanent connection with the virtual classroom*: unlike online web applications that require connecting to the virtual classroom from wired computers, online learning applications using computational devices enable the connection at anytime and from anywhere.
- *Downloading material courses*: students can download material courses from every where and moreover multimedia materials can be used in online learning applications with computational devices.
- *Awareness*: using computational devices in online learning applications enables prompt feedback and provides awareness to students about new events happening in the virtual classroom.
- *Monitoring activity in classrooms*: end-devices, such as smart-box and different types of sensors can be used to monitor the activity of students regarding their performance and accomplishments of their learning activities.
- *Alerting about important calendar dates and events*: through hand-held and end-devices can be implemented functionalities to alert students about important dates of the learning activity calendar.

It should be noted however that currently the implementation of such advanced online learning systems is a challenging task due to the intrinsic complexities of the hand-held and end devices. Indeed, the programming of computational devices and their integration into distributed applications is very difficult. In such applications the target is to use different technologies: web, Grid, P2P and computational devices. The heterogeneity of computational resources in such applications is a major research issue still to be addressed and solved for practical purposes. The first steps in this directions are done by combining web, Grid and P2P computing and nowadays we can find some applications using such technologies. Apart from the heterogeneity and the variety of computational devices, other difficulties arise from the limitations of the computational devices and security issues in overcoming firewalls and NATs.

Fortunately, libraries for supporting the programming of computational devices and their integration into distributed applications are proposed in the literature. Among these libraries, there is JXTA, which offers a set of protocols that enable the connection and communication with any type of computational devices able to be connected acting as JXTA edge peers.

IV. JXTA-Overlay Platform

JXTA technology [12], [13] is a generalized group of XML-based protocols that allow different types of peers to communicate and collaborate among them. Peers can be organized into peer groups in a decentralized way. Peers communicate using pipes, which abstract the way in which two peers communicate, where other peers are allowed as

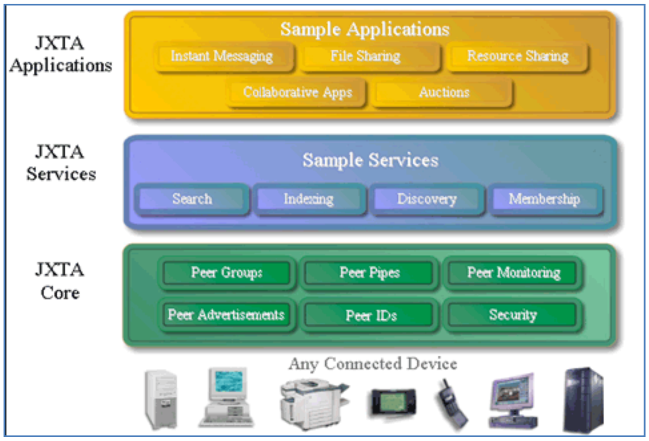

Fig. 1. JXTA layers and services.

intermediaries if communication would not be able due to network partitioning and restrictions. By using these protocols, peers connected to the JXTA network can exchange messages among them in decentralized way. Moreover, by using JXTA protocols it is possible that a peer in a private network can be connected to a peer in the Internet by overcoming existing firewalls or NATs or when different communication protocols are used.

Peers are uniquely identified allowing that peers can change their address still conserving their unique peer Id. JXTA layers and services are shown in Fig. 1.

A. JXTA protocols

JXTA comprises a set of open protocols that *"enable any connected device on the network, ranging from cell phones and wireless PDAs to PCs and servers, to communicate and collaborate in a P2P manner."* We briefly describe below these protocols (core and standard protocols).

Peer Resolver Protocol (PRP): PRP offers a generic interface that allows peers to send one or more peers generic requests and to receive one or multiple answers; applications and services can use the protocol for the creation of resolution services.

Endpoint Routing Protocol (ERP): ERP defines a set of messages which are processed by a routing service to enable a peer's message routing to the destination. Thus, ERP is used to find the available routes to send messages to the destination peers; this is achieved through the exchange of messages among the "router peers".

Peer Discovery Protocol (PDP): PDP is used to discover the published resources by the peers. The resources are represented through advertisements. A resource can be a peer, a peergroup, a pipe, or any other resource that has an advertisement. Each resource has to be represented by an advertisement.

Rendezvous Protocol (RVP): RVP is used to propagate messages in a group of peers. The RVP provides mechanisms for controlled propagation of the messages. This protocol comprises the PeerView Protocol, Rendezvous Lease Protocol and Rendezvous Propagation Protocol.

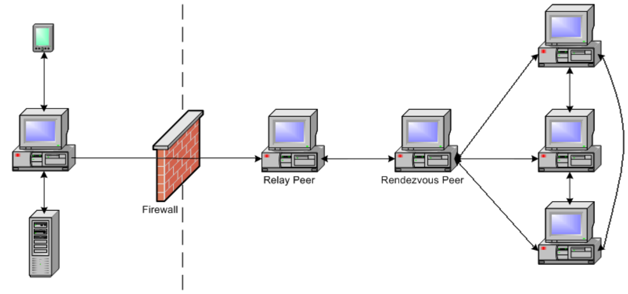

Fig. 2. Communication of peers in a JXTA network.

Peer Information Protocol (PIP): PIP provides a set of messages to obtain information about the state of a peer. The PIP protocol is optional for JXTA peers; in fact, a peer need not to reply to the queries made with PIP protocol.

Pipe Binding Protocol (PBP): PBP is used by the applications and the services in order to communicate with the other peers. A pipe is a virtual channel between two peers and is described as "pipe advertisement". Every time that a pipe is established, an input pipe and an output pipe are established. In fact, PBP can be viewed as a layer over the ERP, and it can use a variety of transport protocols such as "Transport JXTA HTTP", the "Transport JXTA TCP/IP" and the "Secure JXTA TLS Transport" in sending messages.

B. JXTA Entities

The main entities of JXTA platform are as follows.

Peer: Any interconnected node is called peer. Peers work independently and asynchronously with other peers. Peers publish one or more interfaces that are used by other peers to establish P2P connections. Peers can be classified in different types (Limited Edge Peer, Complete Edge Peer, Rendezvous Peer, Relay Peer) according to their role in the P2P network.

- *Rendezvous*: The Rendezvous peers are in charge of coordinating the other peers in the JXTA network. Additionally, they provide the necessary services for the propagation of messages. Each sub-network of JXTA must have at least a Rendezvous peer.
- *Relay*: The Relay peers allow that peers behind firewalls, NATs or special peers having limited computational power such as mobile devices, PDAs, etc., to be part of a JXTA network. The relay peers achieve this by using protocols which allow to cross the limitations imposed by these systems, like for example, the HTTP protocol (see Fig. 2).
- *Edge*: Edge peers are peers at the edge of the network and usually have limited bandwidth as compared to other types of more powerful peers.

PeerGroup: A PeerGroup is a collection of peers that provide a secure shared environment for participating peers. A PeerGroup can decide its own policy of peer membership. Peers can belong to more than one PeerGroup.

Pipes: A pipe is a virtual communication channel established between two processes. A computer connected to the network can open, at transport level, as many pipes as its

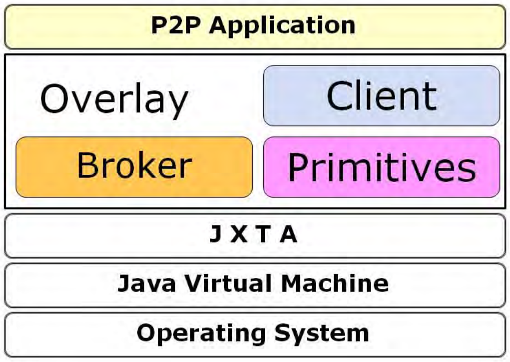

Fig. 3. Structure of JXTA-Overlay.

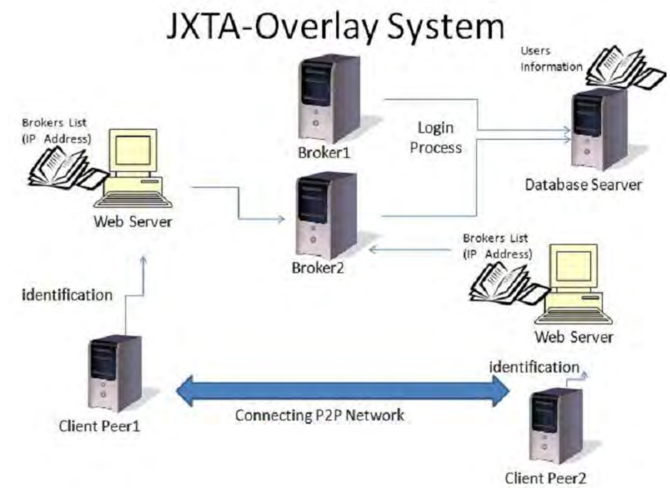

Fig. 4. System image of JXTA-Overlay.

operating system permits. JXTA offers both unidirectional, not secure pipes and bidirectional secure pipes.

Messages: Messages are objects used for communicating and interchanging data. A message is essentially an ordered sequence of tags/values, which could also include binary code (appropriately encoded).

Advertisements: JXTA resources and services are represented using advertisements. An advertisement is a meta-data structured information (XML document), which is published with a certain lifetime specifying its availability.

C. JXTA-Overlay

JXTA-Overlay[1] [14] is a midleware that abstracts a new layer on top of JXTA through a set of primitive operations (services) commonly used in JXTA-based applications.

JXTA-Overlay comprises primitives for: (a) peer discovery, (b) peers resources discovery, (c) resource allocation, (d) task submission and execution, (e) file/data sharing, discovery and transmission, (f) instant communication, (g) peer group functionalities (groups, rooms etc.), and monitoring of peers, groups, and tasks. This set of basic functionalities is intended to be as complete as possible to satisfy the needs of JXTA-based applications. The overlay is built on top of JXTA layer and provides a set of primitives that can be used by other applications, which on their hand, will be built on top of the overlay, with complete independence. The JXTA-Overlay offers several improvements of the original JXTA protocols/services to increase the reliability of JXTA-based distributed applications and to support group management and file sharing [14]. The architecture of the P2P distributed platform we have developed using JXTA technology has these building blocks: Broker Module, Primitives Module and Client Module. Altogether these three modules form a new overlay on top of JXTA. The JXTA-Overlay structure is shown in Fig. 3 and the system image in Fig. 4.

For the definition and implementation of the JXTA-Overlay primitives, the *Broker peer* and two types of *Client peers* have been defined and implemented.

Broker peer: Broker peers are extension of rendezvous peers and are in charge to control the peers in JXTA network. As such, broker peers act as *bots* actually they don't interact with users but are permanently waiting for events/advertisements from peers in the networks. Therefore, broker peers are able to keep the state of the network and propagate it to the peers in the network. Thus, broker peer implements both rendezvous and relay peer functions. Among other proper functions of the broker peers we can mention the login control of peers, management of peergroups and rooms, task assignment and allocation, search of the best peer for file transfer, file search by content, and so on. Although not necessary, broker peers should run on fast machines in order to be able to process the amount of information generated in the network in short times and maintain the updated state of the network.

GUI Client peer: This type of peer implements an edge peer. It is endowed with a graphical user interface, hence called GUI Client peer, to facilitate the operability of a user with the JXTA network. The use can thus collaborate with other peers, send requests for task executions, share, send and receive files. The accomplishment of these functionalities is done through the generation of events propagated to other peers in the network.

Simple Client peer: This type of peer implements an edge peer. However, simple client peers act as bots by not offering interaction with the user. This kind of peer is used to increase the performance and amount of resources in the JXTA network, especially, for distributed computing applications and data storage. The functioning of such peers is completely transparent to the user. They are used by broker peers for task execution and the users need not be aware of such peers, although that they can know which simple client peers are in the network and those participating in a task resolution.

D. Transmission Control and Management in JXTA-Overlay

In JXTA-based applications it is important to know the functioning of message JXTA transmission protocol. JXTA protocol uses Universally Unique Identifier (UUID) in order to identify peers in the private network from the Internet. We implemented a control system that is able to control a peer in a private network from a peer in the Internet. The control targets are considered the network devices such as RS232C

[1] https://jxta-overlay.dev.java.net/

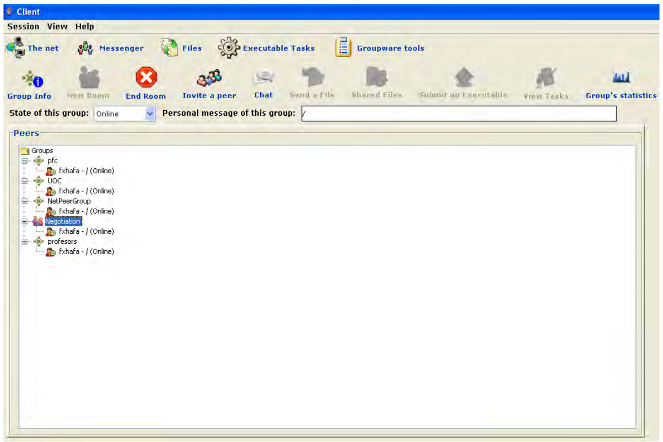

Fig. 5. Snapshot of the JXTA-Overlay and groupware tools.

port, LPT port and USB port. By implementing this kind of control system, we are able to collect data and control the peers in a WAN. Thus, we will be able to control all devices that are connected to the peers. The application of this control function will be described in detail in the following section. We are concentrated in the control of USB devices and RS232C equipment. This is because USB devices are very popular and are used almost in every computer. Also, by using USB is possible to control the motors and LEDs. The RS232C is legal interface and many devices have implemented it.

E. Groupware Tools in JXTA-Overlay Platform

As part of JXTA-Overlay Platform, we have developed also groupware tools to support online learning. The groupware tools in the current version comprise: instant messaging, management of rooms, management of learning scenarios and task coordination among peers of a group (i.e., students of a study group) within a learning scenario (see Fig. 5) for a snapshot of the JXTA-Overlay and groupware tools.

It should be noted that currently, the JXTA-Overlay can be deployed for P2P networks of *standard* peers such as PCs and laptops. In this work we extend the capabilities of the JXTA-Overlay to support also end-devices and use them for enhancing learners' motivations.

V. SmartBox End-Device

Our target implementation is to build and design some end-devices for control in a smart environment. As an end-device, in this work we consider the SmartBox.

The SmartBox uses RFID (Radio Frequency Identification) and Vital Sensors. The size of SmartBox is $37 \times 7 \times 15$ cm. The SmartBox has the following sensors and functions (see Fig. 6).

- RFID Sensor: for identifying user's IC tag card.
- Chair Vibrator Control: for vibrating the user's chair.
- Light Control: for adjusting the room light.
- Aromatic Control: for controlling the room smell.
- Buzzer Control: to emit relaxing sounds.

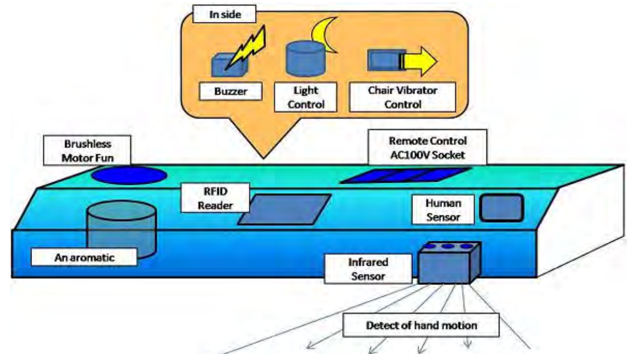

Fig. 6. SmartBox functions.

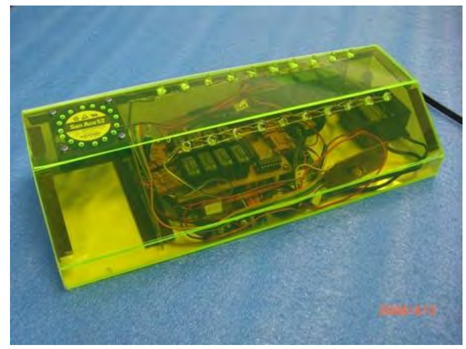

Fig. 7. A snapshot of implemented SmartBox.

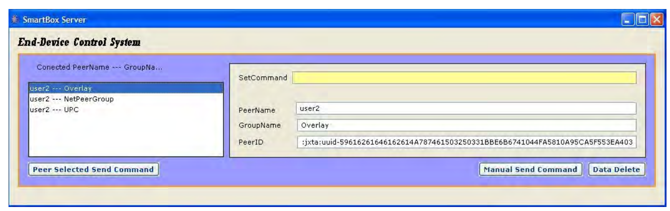

Fig. 8. SmartBox control system interface.

- Remote Control Socket: for controlling AC 100V socket (on-off control).

A snapshot of the implemented SmartBox is shown in Fig. 7. The SmarBox can detect the computer users' movement by its body sensor and hand sensors. The body sensor is used for controlling the body movement of the user. On the other hand, the hand sensors control the hand motion of the user. The RFID sensor can read IC tag information and record the time a user uses the computer. We used the SmartBox as an end device in a P2P e-learning system and we control its functions by using JXTA-Overlay. We developed a control system for controlling the SmartBox. The control system interface is shown in Fig. 8.

VI. Experimental Results with SmartBox

In this section we present some experimental results about the monitoring and performance of the SmartBox in the JXTA-Overlay P2P system. We carried out experiments in real environment and confirmed the effectiveness of JXTA-Overlay. We carried out many experiments by controlling different learners that were using the SmartBox. We verified

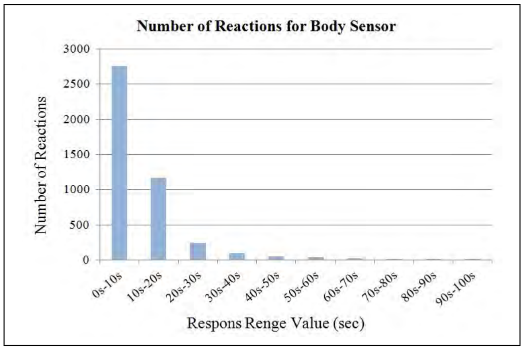

Fig. 9. Reaction number for body sensor.

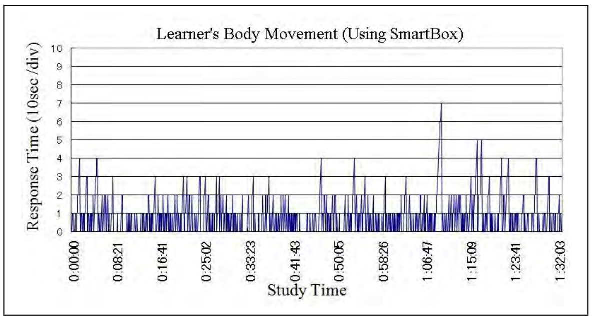

Fig. 10. Student concentration using SmartBox.

the effectiveness of SmartBox by stimulating different users with its functions.

The proposed system can detect the learner's movement by using body sensors. The measurement data for learners' body movements are shown in Fig. 9. We obtained these data after observing two learners studying for a total of 40 hours distributed along different periods. After a sensor reacts, the interval time will be reset to zero. We consider where is the point of effective stimulation for learners from the sensing rate values. We pay attention to the reaction frequency of sensor. The interval of sensors response is very active at the point around 20-30 seconds, but it gets less active after 30 seconds. This tells us that the learners make certain actions every 30 seconds while they are studying. After 50 seconds, there is almost no reaction from the sensor.

In order to check the effects of the SmartBox on the learners, we investigated a high school student while studying English and Mathematics. In the first experiment, we used the SmartBox and in the second experiment we did not use the SmartBox. The learner's body movement for these two cases are shown in Fig. 10 and Fig. 11, respectively. The comparison between these two figures shows that the use of SmartBox is an effective way to improve the learner motivation, because the learner's concentration is higher using the SmartBox.

VII. Conclusions

In this work, we presented the use of JXTA-Overlay P2P system for developing applications for efficient control and management of end-devices. The design, implementation and deployment of full featured P2P networks that integrate not only PCs, laptops and servers but also mobile and end-devices

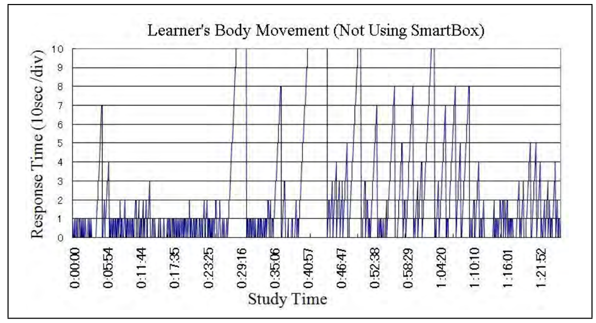

Fig. 11. Student concentration without using SmartBox.

is very important in today's research of P2P systems.

In this work we have shown the control of a SmartBox as end-device in the JXTA-Overlay as well as its use for online learning applications. The proposed systems is validated in practice through many experiments in a real setting. The experimental evaluation showed that the proposed system has a good performance and can be used successfully for the control of end-devices.

In our future work we plan to integrate other end-devices to the JXTA-Overlay platform.

References

[1] F. Xhafa, R. Fernandez, T. Daradoumis, L. Barolli, S. Caballé S, "Improvement of JXTA Protocols for Supporting Reliable Distributed Applications in P2P Systems", in Proc. of NBiS-2007 (Regensburg, Germany), LNCS, vol. 4658, pp. 345-354, 2007.

[2] L. Barolli, F. Xhafa, A. Durresi, G. De Marco, "M3PS: A JXTA-based Multi-platform P2P System and Its Web Application Tools", International Journal of Web Information Systems, vol. 2, no. 3/4, pp. 187-196, 2006.

[3] P. Kumar, G. Sridhar, V. Sridhar, and R. Gadh, "DMW - A Middleware for Digital Rights Management in Peer-to-Peer Networks", in Proc. of DEXA-2005 Workshops (Copenhagen, Denmark), pp. 246-250, 2005.

[4] P. Charas, "Peer-to-Peer Mobile Network Architecture", in Proc. of the 1st International Conference on Peer-to-Peer Computing, pp. 55-61, 2001.

[5] Z. Hu, "NAT Traversal Techniques and Peer-to-Peer Applications", in Proc. of the New Zealand Computer Science Research Student Conference, pp. 242-245, 2008.

[6] K. Kuramochi, T. Kawamura, K. Sugahara, "NAT Traversal for Pure P2P e-Learning System", in Proc. of 3-rd International Conference on Internet and Web Applications and Services, pp. 358-363, 2008.

[7] T. Iwata, S. Miyazaki, M. Takemoto, K. Ueda and H. Sunaga, "P2P Platform Implementation on PDAs Organizing Ad Hoc Wireless Network", in Proc. of Workshops of the Symposium on Applications and the Internet (SAINT-2004 Workshops), pp. 568-573, 2004.

[8] M. Tuisku, "Wireless Java-enabled MIDP Devices as Peers in a Grid Infrastructure", LNCS, vol. 2970, pp. 273-281, 2004.

[9] K. Kubo, J. Murakami, and T. Hoshi, "Hybrid Peer-to-Peer Systems for Network Monitoring of Field Devices", in Proc. of SICE Annual Conference, Fukui University, Japan, 2003.

[10] M. K. Purvis, N. Garside, C. Stephen, M. Nowostawski, M. Oliveira, "Multi-agent System Technology for P2P Applications on Small Portable Devices", Moro et al. (eds.): 3-rd International Workshop on Agents and Peer-to-Peer Computing, LNCS vol. 3601, pp. 153-160, 2005.

[11] T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy, "Transport Layer Identification of P2P Traffic", in Proc. of the 4-th ACM SIGCOMM Conference on Internet Measurement, pp. 121-134, 2004.

[12] D. Brookshier, D. Govoni, N. Krishnan, J. C. Soto, "JXTA: Java P2P Programming, Sams Publishing, 2002.

[13] A. Oram, "Peer-to-Peer: Harnessing the Power of Disruptive Technologies", O'Reilly Publishing, 1st edition, pages 448, 2001.

[14] F. Xhafa, L. Barolli, T. Daradoumis, R. Fernandez and S. Caballé, "JXTA-Overlay: An Interface for Efficient Peer Selection in P2P JXTA-based Systems", International Journal on Computer Standards & Interfaces, Elsevier, Available online 17 March 2008.