A P2P-to-UPnP Proxy Gateway Architecture for Home Multimedia Content Distribution

Chih-Lin Hu, Hsin-Cheng Lin, Yu-Feng Hsu and Bing-Jung Hsieh
Department of Communication Engineering, National Central University
Taoyuan, Taiwan
[e-mail: {clhu, 985203036, 995203010, 995203009}@cc.ncu.edu.tw]
*Corresponding author: Chih-Lin Hu

Received September 19, 2011; revised November 24, 2011; December 15, 2011; Published January 31, 2012

Abstract

Deploying advanced home networking technologies and modern home-networked devices in residential environments provides a playground for new home applications and services. Because home multimedia entertainment is among the most essential home applications, this paper presents an appealing home media content sharing scenario: home-networked devices can discover neighboring devices and share local media content, as well as enormous amounts of Internet media content in a convenient and networked manner. This ideal scenario differs from traditional usages that merely offer local media content and require tedious manual operations of connection setup and file transfer among various devices. To achieve this goal, this study proposes a proxy gateway architecture for home multimedia content distribution. The proposed architecture integrates several functional mechanisms, including UPnP-based device discovery, home gateway, Internet media provision, and in-home media content delivery. This design addresses several inherent limitations of device heterogeneity and network interoperability on home and public networks, and allows diverse home-networked devices to play media content in an identical and networked manner. Prototypical implementation of the proposed proxy gateway architecture develops a proof-of-concept software, integrating a BitTorrent peer-to-peer client, a UPnP protocol stack, and a UPnP AV media server, as well as media distribution and management components on the OSGi home gateway platform. Practical demonstration shows the proposed design and scenario realization, offering users an unlimited volume of media content for home multimedia entertainment.

Keywords: Proxy gateway, universal plug and play (UPnP), peer-to-peer (P2P) file service, media content sharing, home multimedia, home networks

This article is aimed at supplanting the previous conference version, presented in Proceedings of the 29th IEEE International Conference on Consumer Electronics (IEEE ICCE'11), Las Vegas, USA, January 9-12. This article contains fully materials of extended architecture design, prototypical development, and demonstration. This work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC98-2221-E-008-041 and NSC99-2221-E-008-011.

DOI: 10.3837/tiis.2012.01.023

1. Introduction

Recent advances in broadband access systems [1] and home networking systems [2] have provided users with a new playground for home multimedia applications and services [1]. Many modern home-networked devices (HNDs[3]) have upgraded computation, networking, multimedia processing, and storage capabilities. Users with such HNDs now access Internet and Web services much in the manner they used to on desktop computers. Hence, it is natural and inevitable that people in home communities exchange, publish, and share media content with HNDs [2][3][4][5].

Fig. 1 illustrates three ideal scenarios of home multimedia entertainment in a residential environment, consisting of a kitchen, a study, a living room, and bedrooms. First, a user operates a networked TV in the living room to browse and play media files from a desktop in the study. Second, a user in the kitchen operates a tablet PC to watch recorded video files from a personal video recorder in the living room. Third, a user with a mobile phone moves in her bedroom. A mobile phone, which is battery-powered and has lower wireless transmission throughput than does a desktop computer with high-speed Internet access, can communicate with the desktop computer in the study to download large media files from the Internet on behalf of itself, and later access these downloaded files on a home network through short-distance data transmissions. Thus, leveraging various HND resource capabilities is beneficial in supporting home multimedia content distribution from the aspects of user experience and resource utilization.

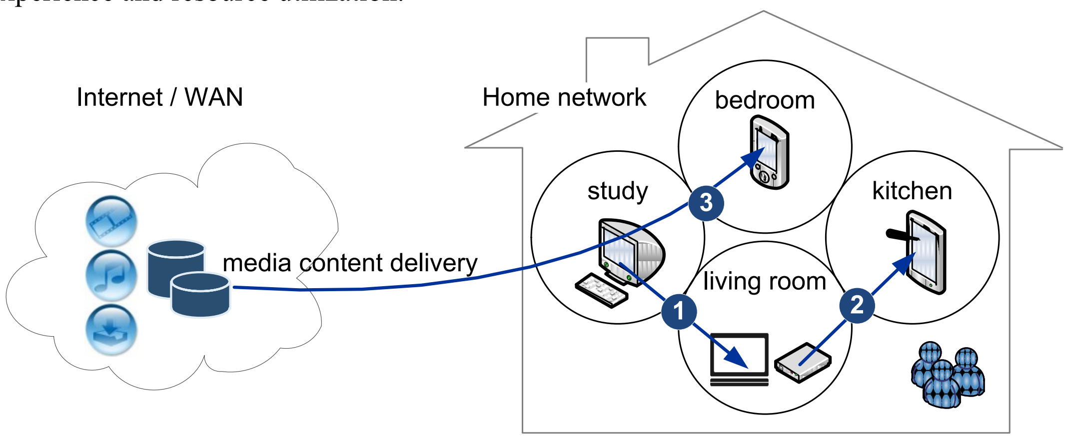


Fig. 1. Home multimedia content distribution in a residential environment.

The purpose of this paper is to describe importing Internet media content onto home networks to increase the volume of available media files for home multimedia entertainment. Regarding how to access external media content, the most popular approach is to use

[1] A broadcast access system to home areas offers users with high-speed Internet access over broadband cable, xDSL, fiber-to-home, or mobile connections.
[2] Several home networking systems, such as DHS, HomePlug, Ethernet, Wi-Fi and FireComms, are used as main technologies for low cost, easy configuration, and reliable high-speed data transmission over phone lines, power lines, coaxes, data cables, and wireless and optical media.
[3] An HND, such as a TV, mobile phone, game console, personal video recorder, IPTV set-top-box, network attached storage, desktop, laptop or tablet PC, is the major product concept of networked multimedia devices including PCs, consumer electronics, and mobile product categories in residential areas. Without ambiguity, this paper uses this abbreviation *HND* to implicitly represent all types of wired, wireless, and mobile-networked devices.

peer-to-peer (P2P) file sharing systems as the source channels for accessing an exceptionally large volume of media files on the Internet [6][7][8]. Observe that some existing HNDs, such as desktops and network attached storage, can already run P2P client applications and exclusively use downloaded files in local storages. Although other HNDs exist on the same network, they cannot access those downloaded files due to distinct principles of networking and media transport protocols between home and Internet application domains. To mediate this disagreement, this study designed a P2P-to-UPnP proxy gateway architecture for enabling P2P file sharing on UPnP-based home networks.

The design of this proxy gateway architecture contends with several considerations and requirements of cost-effectiveness, network interoperability, distribution transparency, scalability, and quality of user experience, as Section 2 explains. Hence, the proposed architecture consists of four mechanisms to perform inter-networking and device discovery, as well as media content convergence, distribution, and management:

- Device and service discovery
- Provision of Internet media files
- Home gateway/server
- In-home media content distribution

The device and service discovery mechanism involves a prerequisite phase to locate resources of interest in neighboring devices to configure and bind resources automatically and dynamically, as compared with the traditional usage that must configure the connection setup and media transfer manually and awkwardly. Previous studies [9][10] have presented several discovery protocols based on LANs and single-hop wireless communications, such as Bluetooth, Bonjour, Jini, UPnP, and service location protocols. Comparatively, UPnP technology [11] complies with existing Internet standards; that is, TCP/IP, HTTP, and Web technologies. Resolving device platform dependency, network interoperability, and service transparency in a single administrative network context is warranted, especially in domestic environments. More importantly, the Digital Living Home Alliance (DLNA) [2] recognizes UPnP technology as an open “micro-middleware” for developing interoperable home network platforms and UPnP-specific devices and services on home networks. Thus, the design of this device and service discovery mechanism exploits UPnP technology to support various device categories and add-on services on small IP networks. In addition, the UPnP AV profile [12] is an extension for media playback services in UPnP-based home network environments.

In providing Internet media files, the proposed architecture integrates a P2P-specific mechanism to participate in the existing BiTorrent[4] P2P file sharing systems on the Internet. HNDs can access external media content using P2P client peers. Because media files in P2P file sharing systems are often encapsulated in basic and compatible data formats, such as mp3 and mpeg, HNDs can simply distribute downloaded media files to others without extra transformation overhead. By contrast, P2P media and live streaming systems require all peers to agree with dedicated media formats and pre-installed playback services, inducing the problems of platform heterogeneity and media format compatibility for home applications [7]. Thus, this architecture design resorts to the benefits of P2P file sharing systems to import Internet media files onto home networks.

An application-level home gateway mechanism plays the dual role of a UPnP media server and a P2P content proxy server. Deploying this home gateway across private and public networks achieves three major benefits. First, a powerful home server exhibiting fast and

[4] As the measure report in [8], BitTorrent is the most popular P2P file sharing application in the global. The BitTorrent and its derivatives contribute a dominant fraction of today’s Internet traffic.

inexpensive connectivity and large content storage is useful for energy savings, cost-effective communication, and content management. Aggregating media files in a home server can facilitate home content distribution in a manageable fashion. Second, a home server can provide higher data throughput to shorten the transmission duration. Resource-restricted or battery-powered HNDs can request the home server to retrieve indicated media content instead, thereby reducing transmission duration and energy consumption. Finally, a home gateway contends with inter-networking, media transport, and media transformation, as well as application and service agreements between UPnP and P2P domains [13][14][15]. HNDs containing distinct hardware and software platforms can obtain a common baseline from the home gateway to support home media applications.

The proposed architecture provides an efficient in-home media content distribution mechanism associated with the home gateway to enable local and P2P content sharing on home networks. This mechanism complies with UPnP conventions to support UPnP-compatible home applications and services. Considering UPnP AV, all HNDs can be classified into UPnP media server, media renderer, control point, media player, and non-UPnP devices. For local media sharing, UPnP media servers enumerate their local media files and then advertise UPnP media players of what media files avaiable on the home network. For P2P media files, the home gateway first transforms downloaded media files adhering to UPnP AV services, and the UPnP media players or control points can then browse downloaded files using UPnP operations. Furthermore, this mechanism designs a P2P-specific metadata directory service that allows HNDs to share P2P meta-files and initiated downloading sessions. Multiple HNDs can subscribe to the same downloading session without redundant media transmission and file duplication, thus resulting in cost-effective performance.

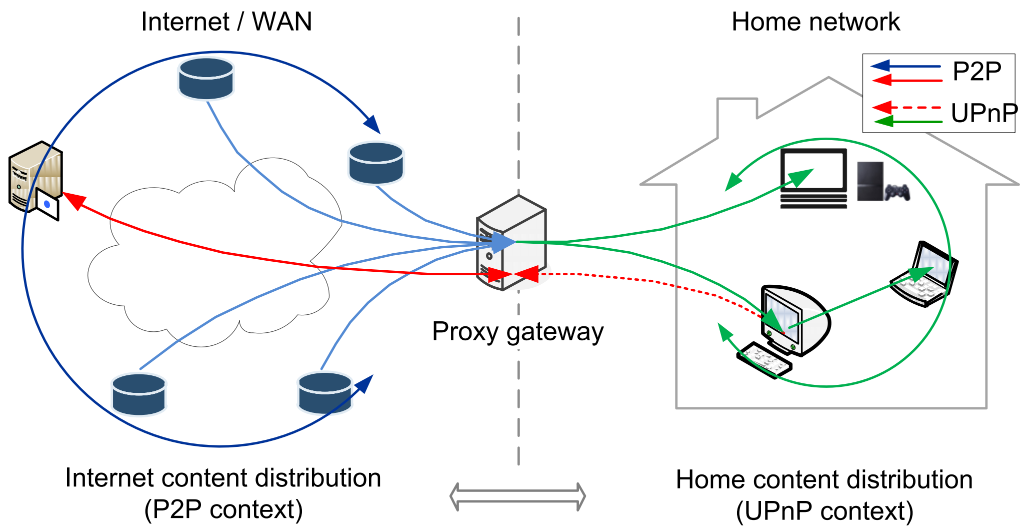


Fig. 2. P2P-to-UPnP proxy gateway for home multimedia content distribution

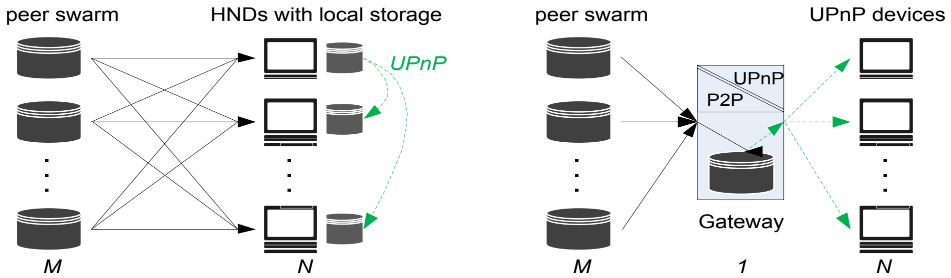
As **Fig. 2** illustrates, the P2P-to-UPnP proxy gateway architecture is able to distribute P2P media content on UPnP-based home networks. Corresponding to the aforementioned mechanisms, the proposed software design of the P2P-to-UPnP proxy gateway architecture consists of four functional components on the home gateway platform, including UPnP device architecture, a P2P client peer, a UPnP downloading manager, and a UPnP AV media server, as detailed in Section 3. Therefore, this architecture design adopts the UPnP middleware for device and service discovery on a distributed home network. All HNDs communicate according to UPnP-specific messaging and interoperability principles. The proxy gateway contends with the convergence of UPnP AV and P2P media content delivery services, providing local media content and Internet media files downloaded from P2P neworks online. Furthermore, the UPnP AV media server supports in-home media content distribution in an accustomed UPnP-friendly manner and attains cost-effective performance. For scenario realization, prototype development involves implementing a proof-of-concept software architecture on the open service gateway initiative (OSGi) service framework [16]. Demonstration results show the effects of the proposed proxy gateway architecture for home multimedia entertainment with new user experience.

The remainder of this article is organized as follows. Section 2 describes design considerations and requirements of the proposed proxy gateway architecture. Section 3 illustrates the software architecture and functional component designs. Section 4 specifies service development and interactive procedures. Section 5 presents the proof-of-concept prototype and real scenario demonstration. Finally, concluding remarks are provided in Section 6.

2. Design Considerations and Requirements

The proposed architecture addresses five design considerations and requirements: (1) cost-effectiveness; (2) network interoperability; (3) distribution transparency; (4) scalability; (5) quality of user experience.

Cost-effectiveness The desgin of the proxy gateway performs the dual role of a UPnP AV media server on a home network and a P2P proxy server to the Internet. The proxy gateway can be deployed on a dedicated HND with higher computing, networking, and storage abilities than other HNDs, thus improving resource utilization of energy, communication, and storage. First, P2P downloading over xDSL connections often requires long hourly durations. The proxy gateway can assist other HNDs in avoiding long downloading processes and fast energy drain. Second, due to peer swarming [17], P2P downloading leads to numerous connections and sessions that HNDs must manage simultaneously. This situation is impractical to most HNDs that are based on embedded systems with limited resource and service capabilities. Using a home gateway instead can reduce communication cost by order of magnitude. For example, as shown in **Fig. 3**, the home gateway results in a lower cost of M+N workload than the M·N workload in a usual situation. Finally, many HNDs have small or no built-in storages. Whereas AV files are counted in dozens or hundreds of MBs, a home gateway with a large storage can contain enough media files, as well as efficiently manage media content to facilitate user experience such as content browsing and searching.



(a) Ordinary scenario **(b)** Proposed scenario

Fig. 3. Conceptual comparison between the ordinary scenario and the ideal scenario

Network heterogeneity The proxy gateway interconnects the public access network and home network, and can mitigate asymmetric resource concerns (that is, bandwidth, computing, and display capabilities) between two sides. Thus, the proxy gateway can support various QoS services, such as trans-coding, trans-rating, and transport modes, to tailor P2P media content to meet HND playback abilities [2]. In addition, the gateway can contend with various bandwidth and end-to-end connection conditions over unreliable, best-effort data delivery environments [13].

Distribution transparency The proxy gateway accounts for access transparency and location transparency [18] to hide the fact that media files are physically distributed across multiple sources on either local or remote networks. Access transparency is intended to hide differences in data representation and how a media resource is accessed. Location transparency involves hiding where a media resource is located. This gateway design offers a virtual view of media files in its home domain, thereby resolving any playback failure resulting from information inconsistency. Thus, other UPnP devices are not required to monitor target media files.

Scalability Size scalability is not harsh to the proxy gateway, because the number of HNDs on a home network is not too large. Requesting workloads and allocating resources can be executed instantly in response to HND media playback and downloading operations. By contrast, *geographical scalability* is noteworthy to the proxy gateway. As **Fig. 3-(a)** shows, peer swarms in P2P systems result in wide-area communications on the Internet [17]. Downloading P2P media files undergoes long communication latency and expends network resources. To mitigate this problem, replication and caching at the proxy gateway is profitable for scaling performance. The gateway device can conduct asynchronous communications with UPnP devices to hide communication latency on P2P networks. In this case, replication inconsistency is negligible because downloaded media files are static and persistent on a home network; thus, no synchronization mechanism is required.

Quality of user experience Inspecting P2P and UPnP media services results in several practical problems, such as file duplication, name ambiguity, and access pattern, which must be resolved. Duplicate files are commonplace because users often back up or move files in different directories or devices. During media playback, users likely browse duplicates from diverse media sources. However, naming ambiguity worsens this situation due to users' renaming operations. Therefore, duplicate files likely have different, similar or ambiguous names, confusing users as generating the playlist. Empirically, storing and sharing media content through the gateway results in an approach to alleviating these problems. Furthermore, the gateway can reflectively analyze access patterns among media files that were played, thus serving as a "featured" media server that provides special information, such as favorites, bookmarks, ranking, and preference profiles, to enhance the quality of user experience.

3. Architecture Design

This section illustrates the proposed proxy gateway architecture designs for home multimedia content distribution. Subsection 3.1 abstracts the architecture design, and Subsections 3.2–3.5 specify four system components and their functionalities.

3.1 Design Abstraction

The design of the proposed P2P-to-UPnP proxy gateway architecture is an elaborate software architecture that consists of four functional components, including UPnP device architecture, a P2P client peer, a UPnP downloading manager, and a UPnP AV media server. **Fig. 4** shows numerous functional components, encapsulated into various bundles and services on the OSGi service framework. A proxy gateway is the hub for media content delivery on home networks. This device can interact with all other HNDs in all processes of inter-networking, networking, messaging, and device discovery, as well as media content convergence, distribution and management. Please notice that the home gateway and other types of UPnP devices have many symmetric functions, including UPnP protocol stacks, UPnP AV profiles and UPnP downloading manager, for media content sharing in home network environments. This section mainly focuses on the gateway specification to save page space but without loss of critical information. The following lists these components and their functionalities.

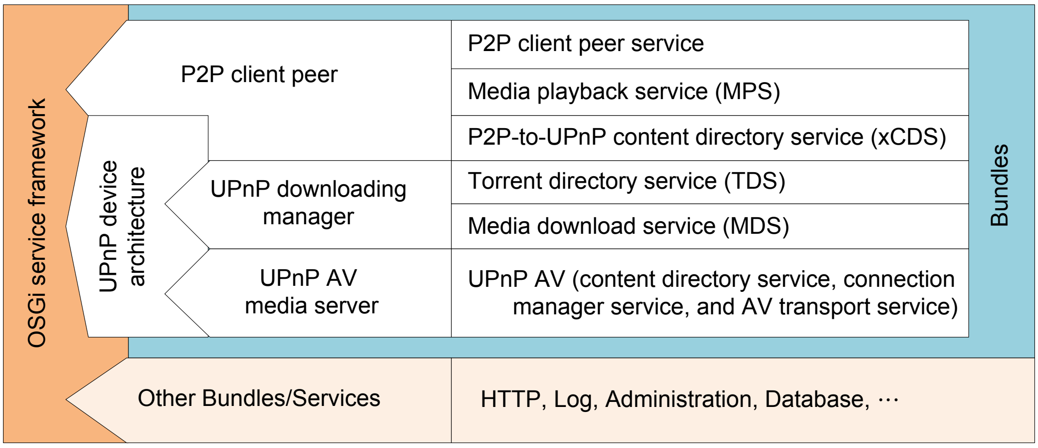


Fig. 4. Software architecture of the P2P-to-UPnP proxy gateway

3.2 UPnP Device Architecture

This component uses UPnP technology to support proximity networking, device discovery, event notification, control action, and data transfer functions on home networks. This involves two device categories: control points, and controlled devices (or simply "devices"). A controlled device functions as a server, offering services that a control point can monitor or control. As **Fig. 5** shows, each UPnP device performs six function layers in bottom-up order. *Addressing* is an underlying function enabling a device to acquire a unique IP address when it joins a network. *Description* is used for a device to summarize its services in a well-defined XML format, so that control points can parse description files and know what a device offers. *Discovery* is used for a device to advertise its appearance on the network, so that control points can locate the device and its services. *Control* is used for a device to manage requests from control points and to invoke specific actions. *Eventing* is operated by a device to notify subscribed control points of state changes. *Presentation* is an HTML-based interface provided by devices for users to control or monitor them directly.

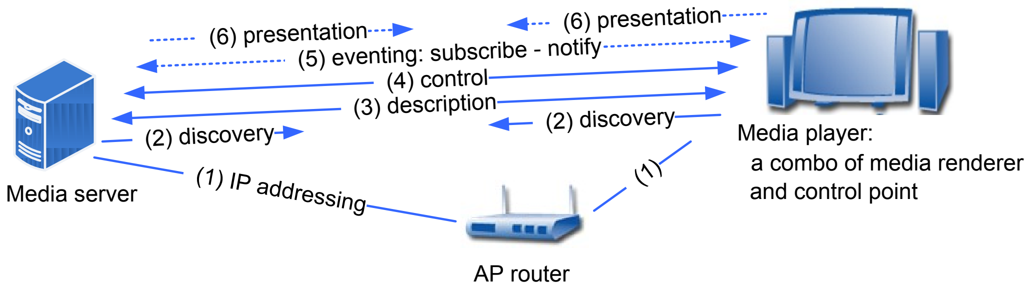


Fig. 5. UPnP messaging and interaction flows.

The proposed architecture leverages the IP-addressing, discovery, description, control, and eventing function layers, but excludes the presentation layer.

UPnP addressing uses two address assignment methods: dynamic host configuration protocol (DHCP) and automatic IP configuration [19]. Automatic IP configuration is used as the DHCP service is not available. When UPnP devices power on, they search for any DHCP server on the network. If no server is present, they use automatic IP assignment to configure dynamic assignment of IPv4 link-local addresses within the 169.254/16 range. When network hosts have IP addresses within the same network domain, they can discover each other.

UPnP discovery is based on the simple service discovery protocol [20], involving a simple HTTP-based discovery mechanism on a small local area network without centralized administration. A UPnP device periodically advertises its appearance on a well-known address/port, 239.255.255.250:1900; that is, an HTTP multicast over a UDP. Each device can directly query the network, and each resource host can directly respond to the request. The HTTP LOCATION headers in these advertisements and response messages specify the URLs in reference to the same description of the UPnP root device. A UPnP device uses an XML-style description to present its services and capabilities. Interested UPnP devices can fetch and parse this description, learning what services the device offers and its profile information.

UPnP control is based on the simple object access protocol [21], essentially an application-level communication protocol over an HTTP, providing a Web-based messaging and remote control mechanism. The proxy gateway architecture preserves this function layer because the UPnP Forum has standardized several device control profiles to achieve consensus on the activities of various device categories [22]. For example, a UPnP AV profile is applied to instantiate any HND that streams AV content to other UPnP devices. However, these profiles have not included Internet content delivery services. Instead, the proposed architecture adopts "customized control profiles" that are newly developed to support the requirements of P2P-specific content browsing, downloading, and meta-data retrieval.

UPnP eventing uses the publisher-subscriber model based on the general event notification architecture [23]. A UPnP device can register at a service for later notification of any changes of service states. The service responds with a subscription ID and valid duration to the subscriber. Subsequent operations, such as renewal and cancellation, can use this ID to refer to the subscription record. In the proposed architecture, for example, multiple UPnP devices can subscribe to the P2P downloading service for the same file. The proxy gateway then periodically notifies all subscribers of the downloading progress.

3.3 P2P Client Peer

The proxy gateway is the only entrance where P2P media content flows onto a UPnP-based home network. This gateway design integrates a P2P client peer software component that provides three services: basic P2P client peer service, media playback service (MPS), and P2P-to-UPnP content directory service (xCDS). First, the proxy gateway runs the P2P client peer service to join the existing BitTorrent P2P system[5] and perform the P2P-specific behavior such as resolving media file metadata (that is, so-called *torrents*[6] in BitTorrent P2P systems), communicating trackers, downloading files, and uploading files to external peers on the Internet. Second, with any downloaded media files, MPS provides media playback functions for non-UPnP HNDs and UPnP devices to browse and play those files in an out-of-band data transfer. Third, for UPnP AV devices and DLNA-specific digital media players, the gateway must account for media transformation and management of downloaded media in consideration of DLNA-specific formats. The xCDS then implements these files into a shared UPnP AV content directory and catalogues those files in compliance with the template of the UPnP AV content directory service [24]. Therefore, the UPnP AV server can share those downloaded files with other UPnP devices on a home network.

3.4 UPnP Downloading Manager

This component provides two UPnP-specific services, torrent directory service (TDS) and media download service (MDS), which UPnP devices can discover and then use to download P2P media files. First, TDS maintains a torrent directory that contains file metadata records, such as <names, lengths, piece sizes used, piece-based hashing codes, and tracker URLs>. TDS also provides control actions for UPnP devices to perform torrent addition and retrieval, torrent enumeration, and torrent searching operations. Using TDS, a UPnP device is able to share torrents with others on a home network. Second, MDS is used to monitor and manage P2P downloading processes on behalf of UPnP devices. Because a P2P client peer is not UPnP-customized, UPnP devices cannot directly interact with this client peer. UPnP devices must request MDS to launch downloading tasks using the indicated torrents. Functionally, MDS coordinates the client peer via internal APIs to perform P2P downloading operations, including concurrent connection establishments, peer selection and assignment, session control and management, file segment assembly, downloading progress report, and publishing and subscribing event notifications.

3.5 UPnP AV Media Server

The UPnP AV [12] further defines three dedicated devices: the media server, media renderer, and control point. A media server provides the content directory service, connection manager service, and AV transport service. The content directory service specifies a set of actions that control points invoke to enumerate media items. For example, the “browse” action allows control points to obtain file attributes, such as data formats and transfer protocols, of media items available on the media server. The connection manager service manages the connections associated with a device. A control point invokes a “preparing-for-connection” action to notify a media server to prepare for upcoming media transfer. This action returns an instance ID of AV transport service that the control points can use to control the transfer flow and connection.

---

[5] The BitTorrent system consists of three roles, trackers, torrent servers and peers, forming in an application-level overlay where the system provides users with basic operations of creating, publishing and downloading torrents, and sharing files for large-scale peer-to-peer content distribution applications on the Internet [6][7].

[6] In BitTorrent systems, a peer treats the file as a number of pieces of identical size, usually between 32 KB and 4 MB. To share files, a peer first creates a torrent that contains file metadata, such as names, lengths, piece sizes used, piece-based hashing codes for data integrity, and a tracker’s URL.

The AV transport service controls the player operations; for example, stop and pause. By contrast, a media renderer is instructed by a control point to control what and how media items are rendered, providing the rendering control service, connection manager service, and AV transport service. The rendering control service provides actions for a control point to adjust the rendering of media contents; for example, brightness and volume. The connection manager service and AV transport service are similar to those of a media server.

The proxy gateway runs with a UPnP AV media server to introduce copious downloaded media files onto the home network. This media server performs the roles of content directory, connection manager, and AV transport services; thus, UPnP devices can play downloaded media files using the UPnP AV conventions. In other words, when the downloading manager completely downloads any files, MDS invokes xCDS to implement those files into a shared content directory and correspondingly update the xCDS-specific DIDL-Lite XML document (that is, the media catalog) [24]. The UPnP devices can then locate newly imported media files using the content directory service, negotiate with the UPnP AV media server using the connection manager service, and transfer media streams for playback using the AV transport service. In addition, MDS can notify UPnP devices (that have subscribed to those media files) of downloading completion events. Therefore, UPnP media servers, media renderers, and control points are able to access P2P media content in a UPnP-friendly runtime context.

4. Service Development and Specification

The service development in the proposed software architecture includes the P2P client peer service, UPnP AV service, TDS, MDS, MPS, and xCDS, as depicted in **Fig. 4**. Subsections 3.3 and 3.5 appropriately mention the first two services. This section mainly specifies TDS, MDS, MPS and xCDS in reference to the state machinee diagram in **Fig. 6**. Accordingly, **Table 1** lists the primitive sets for specific services on the software framework. **Fig. 7** illustrates the procedural action flows and messaging information among these services.

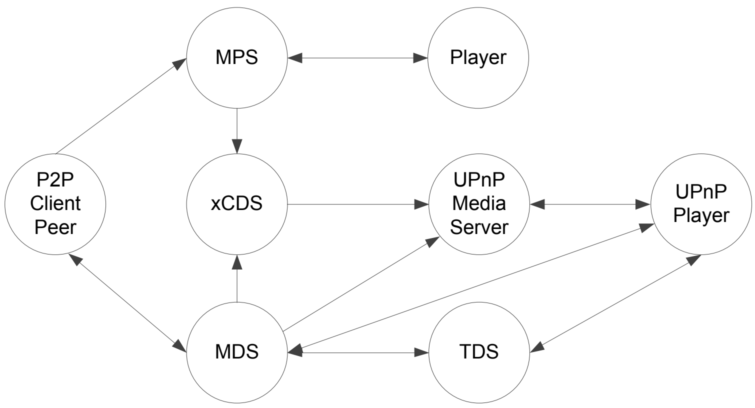


Fig. 6. State machine among specific services

Table 1. Primitives in specific services

TDS	MDS	MPS	xCDS
addTorrent(t_id)	startDownload(d_id)	playMediaObject(o_id)	updateXCDS()
addTorrentURL(t_url)	stopDownload(d_id)	stopMediaObject(o_id)	addItemXCDS()
removeTorrent(t_id)	suspendDownload(d_id)	pauseMediaObject(o_id)	removeItemXCDS()
searchTorrent(t_name)	resumeDownload(d_id)	seekMediaObject(o_id)	cleanXCDS()
browseTorrent()	subscribeDownloadEvent(s_id)		
	unsubscribeDownloadEvent(s_id)		
	getDownloadProgressReport(d_id)		

4.1 Torrent Directory Service (TDS)

The proxy gateway employs a P2P client peer to participate in the existing BitTorrent P2P system on the Internet. Initially, with a torrent of an interested file, the P2P client peer can resolve the torrent content. According to prescribed information in the file metadata, the P2P client peer connects to the *tracker*[7] and queries about the media file. In BitTorrent P2P systems, torrents are originally published on websites and registered with at least one tracker. To distribute P2P media content on a home network, the proposed architecture specifies a UPnP-compliable TDS for torrent exchange and management. TDS provides a torrent repository and control actions for UPnP devices to add, remove, browse, and search for torrents. In addition, TDS can validate torrents and then maintain torrent-specific metadata records in the torrent repository. Therefore, UPnP devices can browse the available torrent list, select interested torrents, and request to access indicated media files from the P2P system. Using TDS, UPnP devices can exchange torrents and indirectly share P2P media content on home networks.

4.2 Media Download Service (MDS)

The design of MDS is sophisticated in coordinating, monitoring, and managing P2P downloading processes on behalf of UPnP devices on a home network. In response to any specific torrents, MDS communicates with the P2P client peer to launch the file downloading tasks. To enhance service agility, MDS maintains a torrent management table where each tuple records < the task index, status, torrent name, torrent URL, temporary local URL, and subscriber list> with respect to every downloading task. This table summarizes the information of all ongoing downloading tasks. MDS repeatedly synchronizes all tasks' information of download statuses (active, complete or idle) and downloading progress (in percentage) with the P2P client peer that is responsible for downloading files and calculating the sizes of accumulative data bytes of those uncompleted files

Users desire to know up-to-the-second downloading progress of their requested files. MDS further maintains a small subscription table, where each tuple only records <the task index, downloading progress, and subscriber list>, to expedite the reporting process in a timely manner. The UPnP devices can actively query about the downloading progress in a request-response manner. To improve efficiency, MDS further provides a publishing and subscribing event notification service based on the UPnP eventing mechanism. When multiple UPnP devices ask for the same file, MDS keeps a subscriber list corresponding to the downloading task. Therefore, only one file copy is stored in the home gateway. MDS notifies

[7] The tracker coordinates the file distribution and tracks all of the peers that have either partial or complete files; therefore, it directs peers to connect with others for file sharing. The peer connects to the tracker indicated in the torrent, from which it receives a random list of peers currently downloading pieces of the indicated file.

every subscribed UPnP device of the URL location reference where UPnP device can play the indicated media file. Therefore, MDS is able to save communication resources, manage the media distribution, and alleviate the concern of file duplication on a home network.

Furthermore, MDS can control bandwidth allocation to improve download throughput among simultaneous downloading tasks. As the number of ongoing downloading tasks increases, the overall download throughput may be affected. Because of radical properties of P2P systems, empirically, it often requires numerous instances of server peer replacement during a downloading session, thus leading to drastically fluctuant throughput. Notably, MDS can reflectively learn the conditions of different downloading tasks according to the periodic reports of downloading progress and session information (practically measured by the client peer). MDS can suspend or resume some downloading tasks that have reduced to extremely low download ratios. MDS can then reclaim more bandwidth and transmission capacities for remaining downloading sessions.

4.3 Media Playback Service (MPS)

The proxy gateway provides two approaches for playing downloaded media files: MPS and UPnP AV, used by non-UPnP HNDs and UPnP devices, respectively. The former is mentioned below, and the latter is described later in xCDS and UPnP AV services.

Specifically, MPS is not yet compliant with UPnP AV conventions. This service aims to offer P2P media content to more media players on a home network. In this design, MPS provides basic media playback operations, such as play, pause, stop, and seek. When HND devices receive the URLs of downloaded media files from an MDS, they can directly communicate with the MPS to play indicated media files using proprietary HTTP/TCP/UDP communications in a request-response manner. This type of playback usage is similar to the fashion that users are accustomed to in the ordinary situation of playing Internet media files.

4.4 P2P-to-UPnP Content Directory Service (xCDS)

Because P2P and UPnP belong to separate application and network domains, the home gateway must contend with media transport, as well as media transformation and management between UPnP and P2P contexts. Unlike MPS operating media playback in a customized and proprietary manner, the proxy gateway must transform downloaded media files to be consistent with the mandatory formats, such as jpeg, mp3, and mpeg, according to DLNA interoperability guidelines [2]. An internal process is in charge of media operation, administration, and management among MDS, xCDS, and P2P client peer on the proxy gateway. Using xCDS involves moving transformed media files into a UPnP AV shared content directory that varies from that used by the P2P client peer. In addition, xCDS can introduce newly imported media content onto a home network. This service catalogues these media files in the shared content directory and promptly updates an xCDS-specific DIDL-Lite XML document to ensure the consistency of content directory service. This process is then followed by the UPnP AV server that advertises the latest information of content directory service to other control points and media players in the UPnP context.

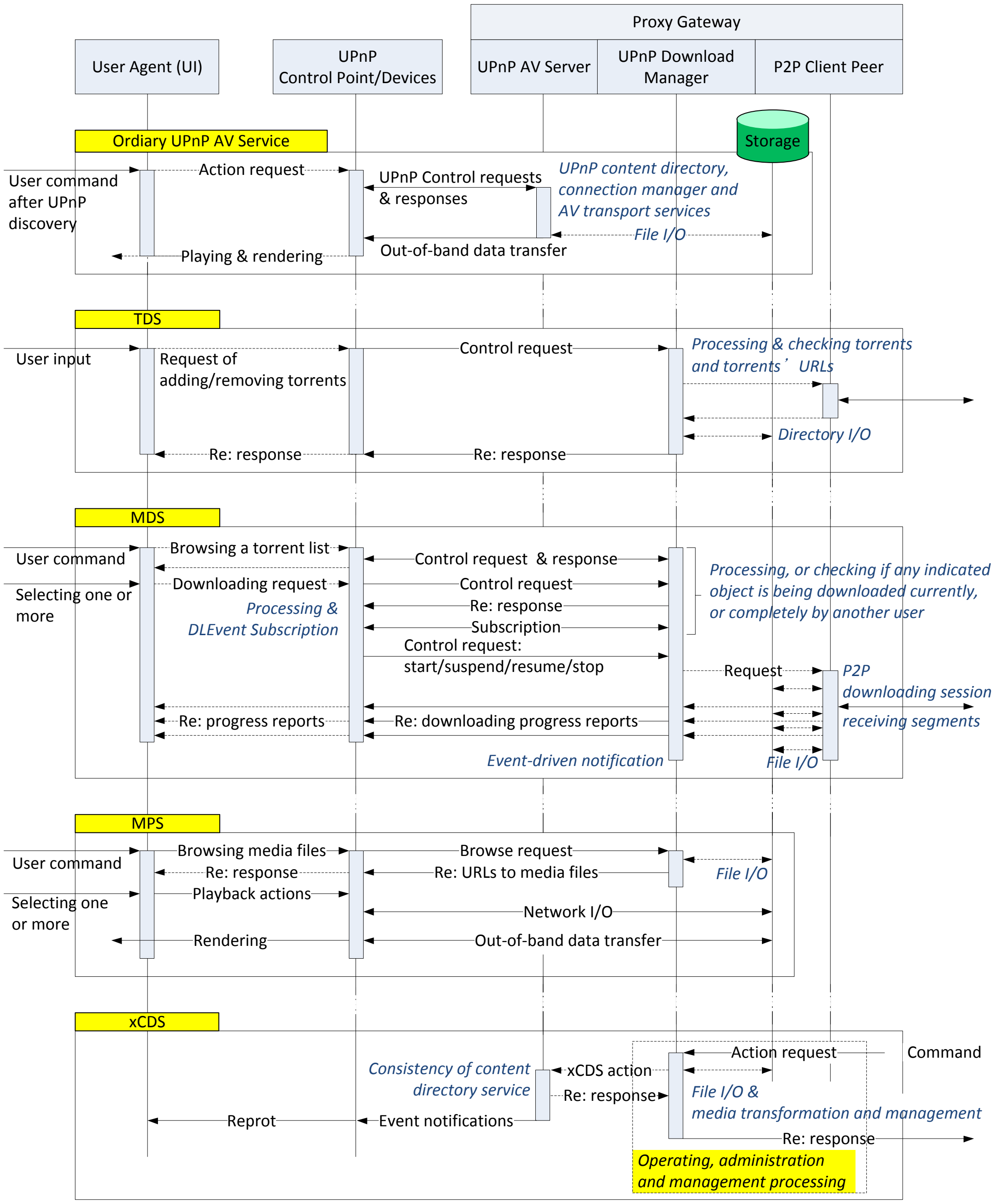


Fig. 7. UPnP messaging and interaction flows.

5. Prototype and Demonstration

This section describes the prototype development, software implementation, and real scenario demonstration to account for the effect of the proposed P2P-to-UPnP proxy gateway architecture for home multimedia content distribution.

5.1 Prototype Development

The prototype development achieves a P2P-to-UPnP proxy gateway software using a Java service development kit (SDK) [25] and Knopflerfish OSGi SDK [26]. The software implementation contains UPnP micro-stack, P2P client peer, UPnP downloading manager, and UPnP AV media server modules, listed as follows.

- Prototypical work develops the software packages of UPnP protocol micro-stack and UPnP AV media server on the OSGi service framework. This allows for developing new UPnP services and applications in the home gateway device. This work also implements the media player software using Java VLC SDK [27] and Java Swing GUI [25] for multimedia content processing and rendering.
- TDS, xCDS, and MDS are developed in consideration of UPnP specifications. UPnP control points and devices can operate these services according to conventional UPnP protocols over HTTP communications. All UPnP messages are represented in simple object access protocols and XML formats [21]. By contrast, non-UPnP devices can use MPS to play P2P media files.
- The prototype implements the BitTorrent P2P client peer application [28] for the purpose of real demonstration and presentation. Thus, the P2P client peer can truly join the existing P2P file sharing overlay on the Internet.
- The software implementation results in a set of OSGi bundles and services that can be registered and loaded into the OSGi runtime context. An HND with an OSGi execution environment can deploy this software as a proxy gateway to offer the developed services and functionalities.

Therefore, the software implementation supports various types of HNDs, such as home gateway, UPnP, and non-UPnP AV devices, to perform home multimedia content distribution collaboratively on a home network. First, OGSi-based devices can load and actiavte the developed proxy gateway software to access P2P media content and share downloaded media files with other HNDs on a home network. Second, UPnP devices can apply the UPnP-compliable TDS and MDS to interact with the UPnP AV meida server that runs on the proxy gateway side. Finally, non-UPnP media player devices can employ MPS to play media files from the proxy gateway, using proprietary HTTP/TCP/UDP communications in a request-response manner.

5.2 Scenario Demonstration

Referring to the aforementioned scenarios in Section 1, this subsection addresses the scenario realization of enabling P2P media content sharing on a UPnP-based home network. Demonstration presents several snapshots of real cases and usages to exhibit the proposed architecture and services.

Fig. 8 displays the demonstrative environment involving three types of HNDs, one home gateway (an OSGi-based proxy gateway running on a dedicated PC), one UPnP AV device (off-the-shelf DLNA digital media player) and three UPnP devices (running UPnP AV media players on laptops). In this case, the proxy gateway is requested to download indicated P2P media files from the Internet on behalf of UPnP devices. While other UPnP devices have the common access interest in a particular file, the proxy gateway serves as a UPnP AV media server that shares the downloaded file on a home network. As shown, the proxy gateway informs those subscribed UPnP devices of 100 % downloading progress when the indicated file is downloaded completely. The UPnP devices then run as UPnP AV media players to play the same media file from the proxy gateway in a UPnP-friendly manner. During the interactive procedure, TDS, MDS, xCDS, and MPS all perform effectively, as specified in Section 4. This demonstration result, therefore, presents the P2P content distribution on a home network.

Fig. 9 displays the GUIs of a UPnP device that currently operates the P2P content delivery, using TDS and MDS. As shown, UPnP devices can apply TDS to add and remove torrents, or obtain a list of torrents stored in the repository on the proxy gateway side. The UPnP devices can then select one or more torrents and request MDS to download indicated media files. Correspondingly, Fig. 10 shows the GUI of the downloading manager on the proxy gateway side. As shown, MDS manages all downloading tasks, each of which is set in an active, complete, or idle state, corresponding to various torrents in practical P2P systems.

As Fig. 10 shows, MDS maintains a torrent management table and a subscription list table that are used to record downloading sessions and their downloading progress. In this design, MDS extracts any secondary requests for the same files into a subscriber list and repeatedly informs subscribed UPnP devices of updated downloading progress. In addition, UPnP devices can actively query MDS about the downloading progress of indicated tasks. Fig. 11 illustrates an example of reporting the downloading progress via UPnP control actions over HTTP connections. Thus, UPnP devices can know the downloading progress of every requested file. When a downloading task is completed, all subscribed UPnP devices can be notified and then invoke MPS to play the media file from the proxy gateway. The media playing and rendering GUIs of a UPnP device are shown in Fig. 9.

Fig. 12 depicts two real traffic I/O graphs of P2P-initiated TCP connections on the Intranet and Internet, respectively. In both cases, the experiments makes three TCP connections to download concurrently distinct segments of the same file from different uploading peers (that are all on the Internet or the Intranet). By comparing Fig. 12-(a) with Fig. 12-(b), P2P-initiated data throughput in the case of the Internet is drastically fluctuant and discontinuous in practice. This is mainly because the server peer assignment does not account for topological proximity. A P2P data transfer likely traverses a long distance, thus resulting in long latency. In addition, the frequent occurrence of peer replacement and connection establishment can induce considerable communication overhead. Therefore, deploying P2P applications on the home gateway rather than in other HNDs is beneficial for the purpose of cost-effective performance, as addressed in Section 2.

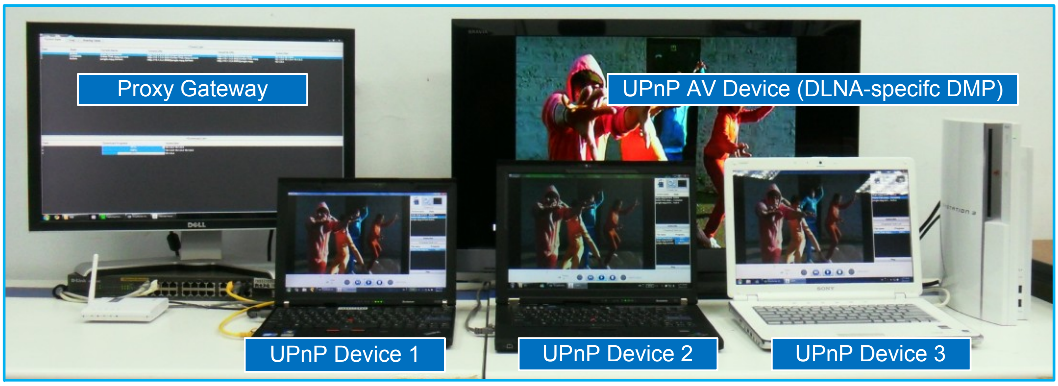


Fig. 8. Prototype demonstration

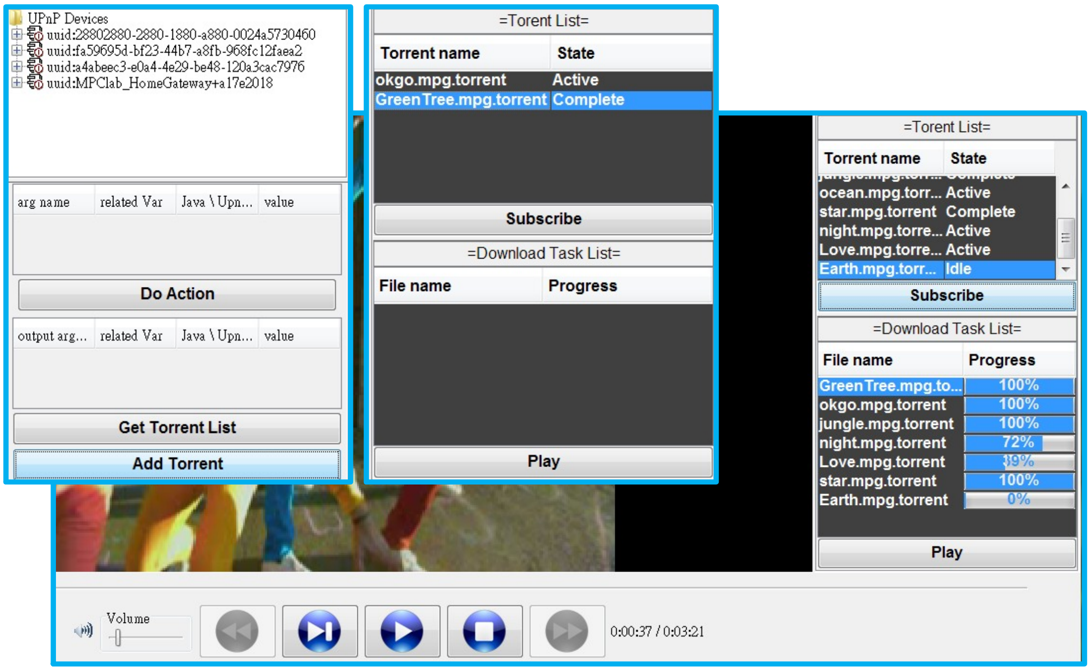


Fig. 9. A UPnP device's GUI

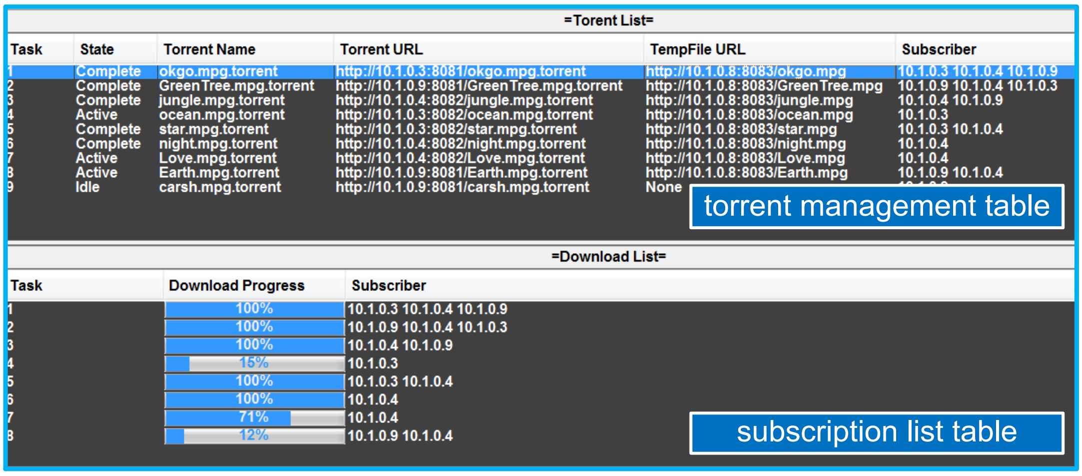


Fig. 10. A UPnP downloading manager's GUI.

```
POST /service/2/ctrl HTTP/1.0
Content-Type: text/xml; charset="utf-8"
HOST: 10.1.0.8
Content-Length: 462
SOAPACTION: "urn:schemas-upnp-org:service:TorrentDirectory:1#GetDownloadProgress"
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/">
   <s:Body>
      <u:GetDownloadProgress xmlns:u="urn:schemas-upnp-org:service:TorrentDirectory:1">
         <TaskNum>8</TaskNum>
         <TorrentName>Earth.mpg.torrent</TorrentName>
         <SubscriberIP>10.1.0.9</SubscriberIP>
      </u:GetDownloadProgress>
   </s:Body>
</s:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Server: Windows 7/6.1 UPnP/1.0 CyberDomo/1.8
Content-Length: 397
EXT:
Date: Wed, 05 Jan 2011 05:08:25 GMT

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/">
   <s:Body>
      <u:GetDownloadProgressResponse xmlns:u="urn:schemas-upnp-
org:service:TorrentDirectory:1">
         <Progress>Earth.mpg.torrent,100</Progress>
      </u:GetDownloadProgressResponse>
   </s:Body>
</s:Envelope>
```


Fig. 11. An example of reporting the downloading progress

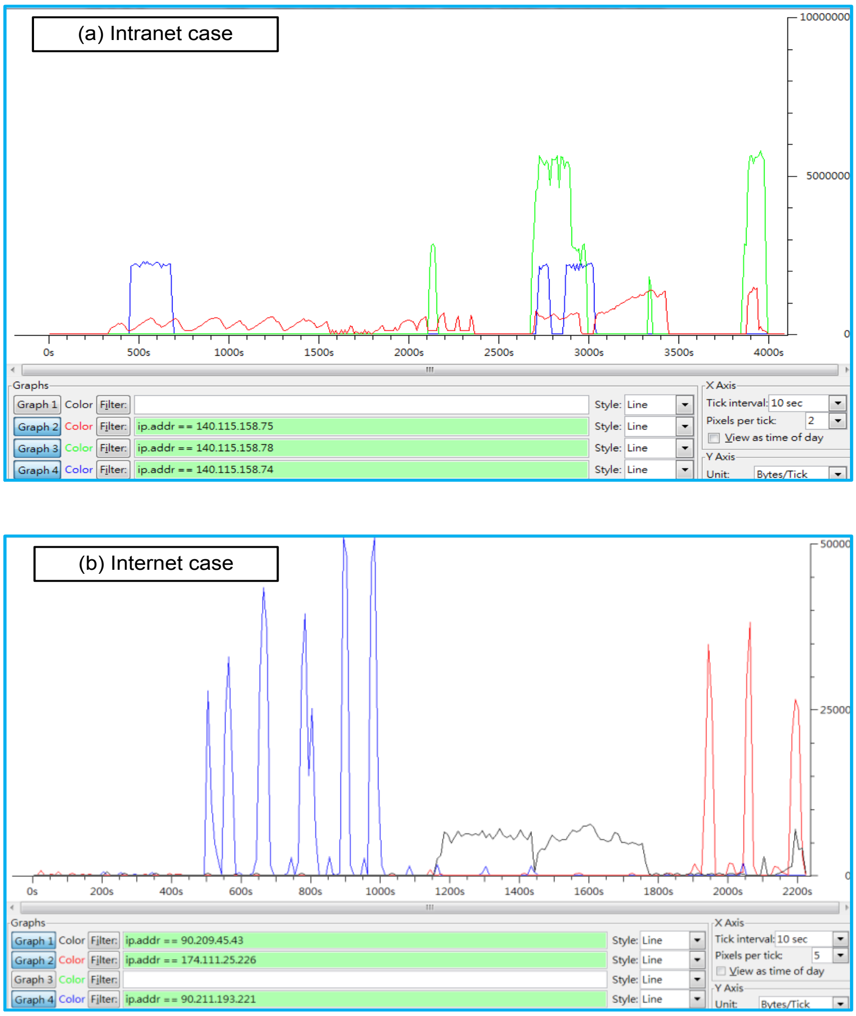


Fig. 12. **(a)** and **(b)** are examples of P2P data traffic I/O graphs representing Intra- and Internet cases.

5. Conclusion

This paper proposes a P2P-to-UPnP proxy gateway software architecture for enabling P2P multimedia services on UPnP-based home networks. The design of this software architecture integrates UPnP and P2P applications based on OSGi frameworks. This design comprises four software mechanisms: device and service discovery, Internet content provision with BitTorrent file sharing services, home gateway with UPnP and P2P application bridging, and in-home media content delivery. On behalf of UPnP devices on a home network, the proposed architecture contends with network heterogeneity, content distribution, media transformation, and quality of user experience in a cost-effective manner. This architecture also provides several UPnP-compliable services, including xCDS, MPS, MDS, TDS, and a UPnP AV media server, by which UPnP devices can provide local media content, as well as P2P media files from the Internet. Prototype development in this study achieved a proof-of-concept software implementation for home multimedia content distribution. Real demonstration confirmed that users are now able to experience new P2P media content in a UPnP-friendly manner in a home network environment.

References

[1] L. Harte, "Home Media Networks: Systems, Technologies, and Operation," *Journal of Althos Publisher*, , 2011.
[2] http://www.dlna.org/industry/certification/guidelines/
[3] J. T. Kim, S. Park, J. H. Lee, E. H. Paik and K. R. Park, "Provision of the multimedia service framework in the ubiquitous home network," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 501-506, May. 2008. Article (CrossRef Link)
[4] C. L. Hu, W. S. Liao and Y. J. Huang, "Mobile media content sharing in UPnP-based home network environment," *Journal of Information Science and Engineering*, vol. 24, no. 6, pp. 1753-1769, Nov. 2008.
[5] T. Hwang, H. Park, E. Paik and J. Chung, "EAFR-Based DLNA proxy for High-Quality video distribution in extended home space," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 120-125, Feb. 2011. Article (CrossRef Link)
[6] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335-371, 2004. Article (CrossRef Link)
[7] J. F. Buford, H. Yu and E. K. Lua, "*P2P Networking and Applications*," *Morgan Kaufmann Publishers*, 2009.
[8] http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf.
[9] K. Vanthournout, G. Deconinck and R. Belmans, "A Taxonomy for Resource Discovery," *Personal and Ubiquitous Computing*, vol. 9, no. 2, pp. 81-89, Mar. 2005. Article (CrossRef Link)
[10] W. K. Edwards, "Discovery Systems in Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 5, no. 2, pp. 70-77, Jun. 2006. Article (CrossRef Link)
[11] http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.
[12] http://www.upnp.org/specs/av/UPnP-av-AVArchitecture-v1.pdf.
[13] H. Park, I. Lee, T. Hwang and N. Kim, "Architecture of home gateway for device collaboration in extended home space," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1692-1697, Nov. 2008. Article (CrossRef Link)
[14] E. Kawamoto, K. Kadowaki, T. Koita and K. Sato, "Content sharing among UPnP gateways on unstructured P2P network using dynamic overlay topology optimization," *IEEE on Consumer Communications and Networking Conference*, pp. 1273-1277, 2009. Article (CrossRef Link)
[15] N. Venkitaraman, "Wide-area media sharing with UPnP/DLNA," *The 5th IEEE Consumer Communications and Networking Conference*, pp. 294-298, 2008. Article (CrossRef Link)

[16] http://www.osgi.org/.
[17] D. S. Menasché, A. A. Aragão Rocha, E. de Souza e Silva, R. M. Meri Leão, D. Towsley, and A. Venkataramani, "Modeling Chunk Availability in P2P Swarming Systems," *ACM Sigmetrics Performance Evaluation Review*, vol.37, no.2, Sep. 2009. Article (CrossRef Link)
[18] A. S. Tanenbaum, and M. Van Steen, "Distributed Systems: Principles and Paradigms," *2nd edition, Pearson Press*, 2007.
[19] S. Cheshire, B. Aboba and E. Guttman, "Dynamic configuration of IPv4 link-local addresses," *IETF Request for Comments (RFC) 3927*, May 2005. Article (CrossRef Link)
[20] http://tools.ietf.org/id/draft-cai-ssdp-v1-00.txt.
[21] http://www.w3.org/TR/soap/.
[22] http://upnp.org/sdcps-and-certification/standards/sdcps/.
[23] http://tools.ietf.org/id/draft-cohen-gena-client-00.txt
[24] http://www.upnp.org/specs/av/UPnP-av-ContentDirectory-v3-Service.pdf
[25] http://www.oracle.com/technetwork/java/javase/downloads/.
[26] http://www.knopflerish.org/download.html/.
[27] http://www.videolan.org/.
[28] BitTorrent P2P Client Application – Wikipedia Online Reference, 2011.

Chih-Lin Hu received the BS degree in computer science from the National Cheng-Chi University in 1997, the MS degree in computer science from the National Chung-Hsing University in 1999, and the PhD degree in electrical engineering from the National Taiwan University in 2003. He was a researcher at BenQ and Qisda Advanced Technology Centers, Taipei City, Taiwan, from 2003 to 2007. Since 2008, he has been an assistant professor in the Department of Communication Engineering, National Central University, Taoyuan, Taiwan, R.O.C. He had the honor to get the best paper award in IEEE ICPADS 2000, BenQ Innovation Awards in 2006 and 2007, and the new scholar outstanding research award in College of Electrical Engineering & Computer Science, National Central University. He had co-organized MDM'09 Workshop on Mobile Peer-to-Peer Information Services (MP2PIS), and IEEE PerCom'10, PerCom'11 and PerCom'12 Workshops on Mobile Peer-to-Peer Computing (MP2P). He has editorial services in International Journal of Mobile Network Communications & Telematics and International Journal of Wireless and Mobile Networks. His research interests include mobile and pervasive computing systems, broadcast information system, digital home network, and Internet technology. Dr. Hu is a member of the ACM and the IEEE

Hsin-Cheng Lin received both BS and MS degrees in communication engineering from the National Central University, Taiwan, R.O.C., in 2009 and 2011. He is currently pursuing the PhD degree at the same faculty. His research interests include P2P systems, home gateway design, and home networking & Internet technologies.

Yu-Feng Hsu received the BS degree in the communication engineering from the National Central University, Taiwan, in 2010. He is currently pursuing the MS degree at the same faculty. His research interests include home networking technology, delay tolerant network, and Internet technology.

Bing-Jung Hsieh received both BS and MS degrees in communication engineering from the National Central University, Taiwan, R.O.C., in 2010 and 2011. His research interests include network programming, home networking technology, and delay tolerant network