A Multimedia Service Migration Protocol for Single User Multiple Devices

Chi-Yu Li*, Ioannis Pefkianakis*, Bojie Li†, Chenghui Peng†, Wei Zhang† and Songwu Lu*
*Computer Science UCLA, Los Angeles, California, USA
{lichiyu, pefkian, slu}@cs.ucla.edu
†Huawei Technologies co., Shenzhen, China
{libojie, pengchenghui, wendy.zhangwei}@huawei.com

***Abstract*—This paper describes a new protocol SMP, which supports multimedia transfer for single-user, multiple-device scenarios. Through its novel naming and control/data plane designs, SMP is able to retain the current client and server protocol operations while placing new functions at the proxy. Our initial evaluation has confirmed its viability.**

I. Introduction

In recent years, it has become the norm rather than exception that a user owns multiple devices with networking capabilities. In an example scenario, a user has a laptop in the office, a desktop at home, while carrying an iPhone wherever (s)he goes. This emerging single-user, multi-device setting opens new venue for networking protocol design and operations.

There are two main challenges for network design. First, the protocol operations should support new data communication patterns for multiple devices of the same user. Data sessions can seamlessly migrate among the devices owned by the same user, or simultaneously multicast to these devices. For example, as friends of the given user want to share video clips with him, he can directly use his office desktop while still in office. However, if he walks out for lunch, he can proceed the ongoing video session via his iPhone or iPad. Second, users are able to continue to run legacy network protocols (particularly those at the transport layer or above) and applications with minimal changes while supporting the notion of single-user, multi-device in data communications. This will enable reuse of most existing Internet applications. Many existing protocols can achieve one of these two goals, but not both.

In this paper, we describe a novel solution, called Service Migration Protocol (SMP), that supports ”single-user, multi-device” multimedia communications. Data sessions are grouped by the user and can seamlessly migrate among the devices belonging to the same user. A key innovation in SMP is the proxy bridging the client and the server in the existing client-server communication model. The proxy offers two critical services of naming and session/data transfer. By carefully designing the functions of the proxy, SMP is able to reuse existing protocol operations at both the client and the server without modifications. Our initial evaluation has confirmed the effectiveness of SMP design.

The rest of the paper is organized as follows. Section II illustrates the usage scenario and identifies the design requirements. Section III describes the related work, and Section IV presents the architecture. Sections V and VI elaborate on the SMP design and the naming support. Section VII evaluates SMP and Section VIII concludes the paper.

II. Single User Multiple Devices

In this section, we present an example scenario of our goals and identify the requirements for our design.

A. An Example Scenario

Alice wants to share a video clip in real time with her friend Bob over the network, after taking a video clip by herself or finding an interesting one on Youtube. However, Bob has multiple devices including a smartphone, an office laptop, and a home desktop. Alice clicks the ”Share video” button and selects Bob’s name from her contact list. The video is then delivered to the most appropriate device of Bob’s at the time. Initially, the video is sent to the office laptop while Bob is in the office. Upon receiving the Alice’s request, Bob clicks the ”Accept” button to receive the video. Sooner on his way to home, it is delivered to his Smartphone. After arriving at home, Bob chooses to switch the video sharing to his home desktop. This scenario is shown in Figure 1. The video service is smoothly switched from one device to another so that Bob can experience uninterrupted video among his multiple devices.

B. System Requirements

To achieve the above scenario, we need to address the following three issues. First, how to deliver video service to the most appropriate device of a user? Second, how to migrate an ongoing session from one device to another? Third, how should the namespace be designed to support the single-user, multi-device scenario? We describe the requirements of our design in the following based on these issues.

1) Service Delivery and Migration: To find and locate one’s most appropriate device for a given situation and then deliver video content to it, the system needs to keep the ranking of preferred device(s) by each user, as well as the status and the locator of each device. It should also consider both control-plane and data-plane. The former consists of all the signaling functions including migration triggers, the discovery of one’s best device and the device to which service is migrated, and service transfer. The latter needs to keep track of each service

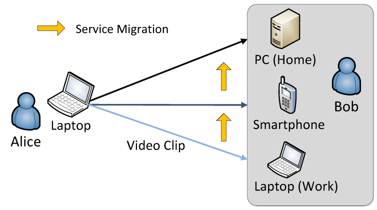

Fig. 1. Alice shares a video clip with Bob.

session, as well as deal with migration delay and possible transient loss during service migration.

2) Namespace: The namespace should consider the identities of both user and device, as well as provide mapping functions to map one user to multiple devices and associate each device to its locator. Although IP address acts as the roles of both identity and locator, these two roles should be decoupled in order to prevent long-term usage identity from changing with transient locator. Therefore, the namespace design should be based on the Identity/Locator split approach.

Moreover, for easy deployment, our design should bypass the modification of the existing protocols and applications.

C. Applications

Our design can be applied to various multimedia applications which provide either open source support or developer API, and are able to run on multiple platforms, such as VLC [1] and YouTube [2].

III. Related Work

There have been several solutions [3]–[5] designed for service migration, but all of them cannot satisfy our requirements. Msocks [3] is unable to support migration among different devices. In SockMi [4], either applications or the existing protocols need to be changed. TSMP [5] is not designed for multimedia service.

Our namespace design is based on the Identity/Locator split approach, which has been used to address mobility issues by a number of protocols [6]–[11]. Three major identities are introduced respectively: Service ID (SID), User ID (UID), and Device ID (DID). C2DM identifies only user with UID, whereas only device is identified by DID in APNS. DONA [8] provides service-oriented mobility by identifying only service with SID. SBone [9] presents the concept of identifying all but service. However, our requirements cannot be satisfied with only part of these identities. UIA [10] and HIP [11] replace the identifier role of IP address with DID so that an ongoing session of the transport layer would not be interrupted as the IP address of either end changes. However, they cannot work without the modification of transport protocols.

IV. Service Migration Protocol Architecture

We employ a proxy-based solution, in which a proxy mediates video service between two ends, to achieve best delivery of service and service migration without the modification of the existing protocols and applications. As shown in Figure 2, the architecture of service migration protocol (SMP) consists of three major components: SMP Server (SMPS), SMP Application (SMPA), and SMP Proxy (SMPP). SMPS performs namespace management and resolution service. SMPA, which is installed on each SMP-enabled device, is used to manage the owner's namespace group and trigger SMP service. SMPP, which is interposed between video server and client application, deals with service delivery and migration.

A. SMPS

SMPS maintains a database of the global namespace with the namespace management module (NMM), which manages user registration, user and device introduction, and namespace synchronization. The first two functions are used for a user to construct his/her namespace group, which includes his/her own devices, friends and friends' devices. The last function keeps the global namespace and each namespace group synchronized. Based on the namespace database, SMPS is able to resolve a device's locator and the most appropriate device(s) of a user with the resolution service module.

B. SMPA

SMPA provides users an interface with a contact list and the functions of SMP service. The contact list enables the owner to manage his/her namespace group and to choose a friend or a device to have SMP service. The namespace group is maintained by namespace sync module (NSM), which collaborates with NMM at SMPS. The SMP functions are done in SMP service module (SSM), which issues and accepts service requests through SMPP, as well as interacts with the video application.

C. SMPP

SMPP consists of two planes, control plane and data plane. The control plane performs the signaling of video sharing and coordinates the operation of service migration. After receiving each request, it finds and locates the target device using the SMPS resolution service. For video sharing, it then informs the device's SMPA of the sharing request through its SSM. For service migration, it then asks the SMPA to prepare for the migrated service through the SSM, and will manipulate the behavior of the data plane through its update module to switch the service to the target.

The data plane acts as a bridge between two ends of each session by relaying packets from either end to the other based on a forwarding table. When the video application requests a video service through SMPP, it will get the service on behalf of the application. Once a video session is set up, a new entry will be created for the session in the forwarding table. Each forwarding entry includes the video information and the corresponding addresses of the server, SMPP and the client, so that it can bridge the session by modifying the addresses of its packets and forwarding them. To support service migration, the update module is provided to update a session's forwarding entry while it is ongoing.

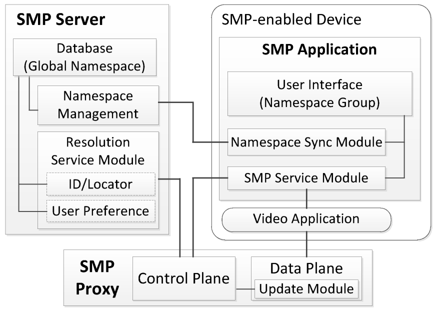

Fig. 2. SMP system architecture.

V. Service Migration Protocol for Video Service

SMP aims to deliver service to one's most appropriate device(s) during service initialization and perform service migration among devices. To achieve them, we need to address the following four issues. First, how to enable one's most appropriate device(s) to receive the shared video? Second, when and how will service migration be triggered? Third, how to migrate an ongoing session? Last, what is the timing for switching service from the old device to the new one?

Before addressing these issues, we introduce four control messages, which are sent via HTTP or directly over TCP connections: Video Sharing Request (VSR), SMP Resolution Request (SRR), Migration From Request (MFR) and Migration To Request (MTR). VSR is used by SSM when a user wants to share a video clip with another, so it should include a video URL and a sharing target, which can be user identity (UID) or device identity (DID). We will detail the namespace design in the next section. The SMPP control plane uses SRR to resolve a device's locator or a user's preferred device(s) through the SMPS resolution service by attaching either a UID or a DID. SSM employs MFR to request the migration of one of its ongoing sessions, whereas the SMPP control plane uses MTR to ask the target device's SMPA to prepare for receiving a migrated session. Both MFR and MTR should contain the information of the migrated service and the target device.

A. Best Delivery of Service

When receiving a VSR message, the SMPP control plane finds and locates the target devices using SRR messages with SMPS, and then forwards the VSR to them. If a DID is provided in the VSR, the device's care-of-address (CoA) will be resolved. If only a UID is attached, a list of the user's preferred devices and their CoAs will be resolved. After locating all the target devices, it forwards the VSR to them. For each device which receives the request, its SMPA SSM will invoke the video application to connect to SMPP with the input of a service identity (SID), which is the concatenation of the device's DID and the shared video URL. Each SID, which is globally unique, is used to identify an ongoing service. Then, the SMPP data plane gets the video URL from the SID and requests the video service on behalf of the application.

B. Service Migration Triggers

Service migration can be triggered by user or network. For the user-triggered, each user can request service migration through a command of the SMPA with choosing an ongoing service and a target device. Then, SSM sends a MFR message to SMPP to trigger migration. For the network-triggered, the SMPP data plane examines the reception quality feedback of each session, and triggers migration if a specified threshold is achieved. When we here consider video services delivered within RTP sessions, it continually checks the receiver reports of each session's RTCP packets to see if the fraction of the lost packets exceeds the threshold.

C. How to Migrate an Ongoing Session?

After a migration is triggered, the control plane asks the new device to prepare for service reception, and then switches the service from the old device to it. With the DID in MFR, it resolves the new device's CoA, and sends a MTR with the migrated session's SID to its SMPA. Then, the SSM invokes the device's video application with a temporary SID, which is the concatenation of the old SID and the device's DID, to connect to SMPP. This connection represents that the application needs to get the session description to set up a session. SMPP always caches the session description of each ongoing session so that it can respond a new version description with some necessary changes to it. Based on the temporary SID, the data plane updates the migration information, which includes a new SID, and the new device's DID and address, to its update module. The new SID can be generated by excluding the old device's DID from the temporary SID. At the end, the update module will commit this update into the forwarding table based on the rules about when to switch service.

D. When to Switch Service?

The timing for switching a service to the new device by committing the update information into its forwarding entry depends on the service type. We consider MPEG4-encoded video services in this work. In MPEG4, a large portion of packets cannot be successfully decoded without their neighboring packets. Its video content is partitioned into multiple groups, each of which is a Group of Pictures (GOP) and can be decoded individually. Each GOP has three types of frame: I-frame, P-frame and B-frame. An I-frame starts a GOP and does not depend on any frame, whereas P-frame and B-frame, which depend on others, follow it. For avoiding transient frame loss, the timing of switching service should be right before the first packet of a GOP, or the new device would not be able to decode the first GOP it receives. Once the data plane gets a session's update, it starts to monitor its data packets until the update is committed. When it receives the first packet of the following first GOP, it commits the update into the forwarding table and the session's data packets accordingly start to be forwarded to the new device.

E. SMP Service Procedure

We present the procedure of SMP service in this section.

1) Best Delivery of Service: Figure 3 shows that Alice wants to share a video clip with Bob by clicking "Share video" button, selecting "Bob" from her contact list and inputting the video's URL on her laptop's SMPA. The SMPA then

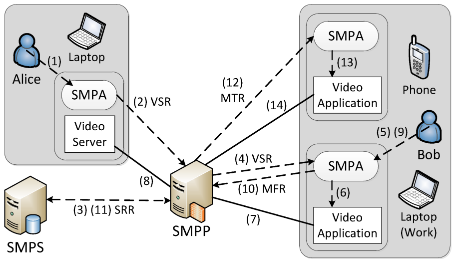

Fig. 3. Alice shares a video clip with Bob through SMP system. The video service is initially delivered to Bob's laptop, and then switched to his phone.

sends a VSR to SMPP, and SMPP discovers the Bob's most appropriate device using the SRR with SMPS. The VSR is then forwarded to the SMPA at Bob's office laptop, which has the highest priority during daytime and is online. A confirmation is then shown up on his laptop's SMPA. After Bob clicks the "Accept" button, SMPA invokes the video application to connect to SMPP. SMPP then requests the video on behalf of it and bridges the corresponding session.

2) User-triggered Service Migration: Before leaving the office, Bob wants to switch the video service from his laptop to his phone. From Step 9 in Figure 3, he clicks "Service Migration" button, selects his phone from his contact list and chooses this ongoing service on SMPA. SMPA then sends a MFR with the service's SID to SMPP. After locating Bob's phone, SMPP sends a MTR to its SMPA. The SMPA invokes the video application to connect to SMPP once this request is received. The service will be switched to the phone after some delay and the original session of the laptop will be interrupted.

VI. Naming and Namespace Management

In this section, we introduce the namespace design in SMP and several fundamental management functions.

A. Naming Principles

The namespace is designed based on the ID/Locator split approach. We organize it into three layers: Name, ID and Locator. They are joined with two-dimensional (2D) mapping: Name to ID to Locator, User ID (UID) to Device IDs (DIDs).

1) Name/ID/Locator: The SMP system maintains a namespace group for each user, which the contact list in the SMPA is based on. In the list, friends (users) and devices are recognized with user name (UN) and device name (DN) respectively. In each namespace group, the names, which are changeable and human-readable, are assigned by its owner. Two identities, UID and DID, are introduced to identify user and device respectively. DID substitutes for the identity role of IP address so that IP address serves as only the locator. Both of UID and DID are globally unique and persistent. The email address used to register the SMP system is considered as UID. A device's DID, which is represented in DNS-like dotted notation, is generated by combining its owner's UID with its name. The initial device name has to be unique in the owner's device set so that DID can achieve global uniqueness with UID. For example, Bob registers his UID as *bob@ucla.edu* and the DID of his laptop, named *laptop* at its registration, would be *laptop.bob@ucla.edu*.

2) 2D Name Mapping: Each locally unique UN or DN is associated with a globally unique UID or DID respectively, and each DID is mapped to its care-of-address (CoA). The former mapping is maintained in each namespace group, whereas the latter is managed in the global namespace at SMPS. Devices can discover each other with the peer's DID through the DNS-like resolution service at SMPS. Another dimension of mapping is between a UID and (multi-)DID as a user may own more than one devices. It can be done by the identity itself because each DID contains its owner's UID.

B. Namespace Management

Each user has a namespace group in the SMPAs of his/her devices, in which (s)he manages his/her own devices and keeps the information of his/her friends and friends' devices. SMPS manages the global namespace, which contains all the namespace groups and users' preference settings, as well as both the CoA and the status of each device. A user's preference setting is the ranking of preferred device(s) by the user for different situations, whereas a device's status can be online, offline, busy or away. A namespace group is constructed and maintained by three functions: service registration, user and device introduction, and namespace state synchronization.

1) Service Registration: Each user needs to register the SMP system with his/her email through an installed SMPA at any of his/her devices before using SMP service. His/her namespace group will then be created, which initially contain only the information of the device used for registration.

2) User and Device Introduction: Users can introduce with each other and so can devices, using two schemes: Local Rendezvous and Centralized Coordination. The owner(s) of two devices can connect both of them to a common local area network such as WiFi, and apply the local rendezvous tool in SMPA, which is similar to Apple's Bonjour [12], to find each other. Both user and device introduction can also be done by issuing requests through the SMPS coordination.

3) Namespace State Synchronization: SMPA uses periodic heartbeat messages to maintain its device's status and employs the latest modification timestamp to check if it should synchronize its namespace with SMPS. SMPS responds to each heartbeat message with the information of all the changes of the devices' status in the SMPA's namespace group, and SMPA then updates these changes in its contact list. When SMPS detects a lack of several consecutive heartbeat messages after a time period, the device's status will become off-line. To reduce the overhead of namespace synchronization, only the latest modification timestamp of the SMPA's namespace group is included in the heartbeat messages. If it is different from the timestamp in the SMPS database, SMPA will synchronize its namespace group with SMPS.

VII. Evaluation

In SMP, service migration may incur delay so as to influence users' satisfaction. We thus examine its performance by evaluating how much delay service migration would incur in various settings. The evaluation is conducted in NS2 and considers

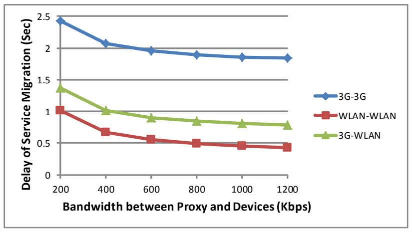

Fig. 4. Migration delay varies with devices' bandwidth.

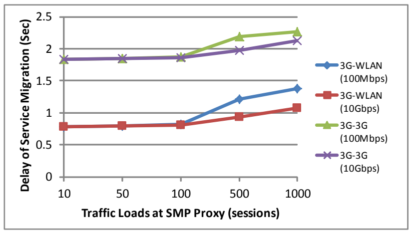

Fig. 5. Migration delay varies with the scaling number of sessions.

three different combinations of 3G and WLAN networks, which are involved in the migration process.

A. Experimental Setup

Our deployed topology contains SMPS, SMPP, a server and a pair of the clients which are involved in service migration, as well as multiple pairs of server and client, each of which has an ongoing service in SMP. All of them are implemented in NS2 except for the global namespace, which is maintained in MySQL database [13]. We use the Evalvid tools [14] to generate video traces and then feed them into NS2 based on the evaluation framework [15]. Based on the traces including the information of each frame and its corresponding packets, the server sends video traffic to the client. We input the values of some network parameters into NS2. The processing delay of SMPS is 200ms per request based on the statistics of Twitter servers [16]. The processing delay of SMPP is set to 380ns per packet [17]. The network latency between SMP devices and SMPP is set to 740ms and 38ms for 3G and WLAN networks respectively, based on the measured latency between our devices and Google server. Moreover, the latency between SMPP and SMPS is set to 1ms, and the bandwidth between them is 10Gbps. The bandwidth between each device and SMPP is set to 1Mbps if it is not specified. To measure the delay of service migration, we calculate the time between the action that MFR is sent by the old client and the action that the new client can play video by receiving the first video frame. We assume that video applications are always ready for use, because how long it takes for SMPA to invoke an application depends on different applications and platforms.

B. Migration Delay

We examine migration delay by varying the bandwidth between SMPP and SMP devices. The number of concurrent sessions is set to 100. As shown in Figure 4, the delay in the WLAN-WLAN scenario, which performs less delay than others, can achieve below 0.5 second when the bandwidth is large. When the bandwidth is low, the performance becomes worse due to network congestion, which slows down migration process. Moreover, the delay increases with the network latency between SMPP and SMP devices because the migration process contains several exchanges of control messages. As a result, 3G network results in longer delay than WLAN. We leave the customization of video service for heterogeneous access technologies during migration to the future work.

We conduct scaling scenarios by varying the number of concurrent sessions from 10 to 1000 at SMPP. Figure 5 shows that the 3G-WLAN scenario still incurs lower delay than the 3G-3G does. There is a minimum migration delay each scenario needs due to network latency, and the delay increases with the traffic loads of SMPP. Therefore, the processing power of SMPP also has an impact on the migration performance. We can adjust it based on the loads of SMPP to guarantee the migration delay to be below a certain number of seconds.

VIII. Conclusion

We are marching toward the post-PC era with the proliferation of various portable devices owned by a user. How to adapt network protocols to such "single-user, multi-device" scenarios becomes a new challenge. The goal is to allow for users to communicate with others anytime, anywhere, and from any device. In this paper, we have described our initial effort along this direction. The main feature of SMP is to place most new functions at the proxy middlebox, while imposing no changes on the existing client and server-side protocols.

References

[1] Vlc. [Online]. Available: http://www.videolan.org/

[2] Youtube. [Online]. Available: http://www.youtube.com/

[3] D. A. Maltz and P. Bhagwat, "Msocks: An architecture for transport layer mobility," in *IEEE INFOCOM*, 1998.

[4] M. Bernaschi, F. Casadei, and P. Tassotti, "Sockmi: A solution for migrating tcp/ip connections," in *PDP*, 2007.

[5] C. Li *et al.*, "A tcp service migration protocol for single user multiple devices," in *IARIA UBICOMM*, 2011.

[6] C2dm: Google android cloud to device messaging. [Online]. Available: http://code.google.com/android/c2dm/

[7] Apns: Apple push notification service. [Online]. Available: http://developer.apple.com/library/ios/

[8] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM*, 2007.

[9] P. Shankar *et al.*, "Sbone: Personal device sharing using social networks," Rutgers University, Tech. Rep. DCS-TR-666, Feb. 2010.

[10] B. Ford *et al.*, "Persistent personal names for globally connected mobile devices," in *OSDI*, 2006.

[11] R. Moskowitz and P. Nikander, "Host identity protocol (hip) architecture," RFC 4423, May 2006.

[12] Apple bonjour. [Online]. Available: http://developer.apple.com/networking/bonjour/

[13] Mysql. [Online]. Available: http://www.mysql.com/

[14] Evalvid: a video quality evaluation tool-set. [Online]. Available: http://www.tkn.tu-berlin.de/research/evalvid/

[15] C.-H. Ke *et al.*, "An evaluation framework for more realistic simulations of mpeg video transmission," *J. Inf. Sci. Eng.*, pp. 425–440, 2008.

[16] Scaling twitter. [Online]. Available: http://highscalability.com/scaling-twitter-making-twitter-10000-percent-faster

[17] G. Liao, X. Zhu, and L. Bhuyan, "A new server i/o architecture for high speed networks," in *IEEE HPCA*, 2011.