```
Internet Engineering Task Force                                    C. Li
Internet-Draft                                                      UCLA
Intended status: Informational                                     B. Li
Expires: May 9, 2013                                             C. Peng
                                                                W. Zhang
                                                                  Huawei
                                                                   S. Lu
                                                                    UCLA
                                                        November 5, 2012
```

A TCP Service Migration Protocol for Single User multiple Devices

draft-li-tsvwg-tcp-service-migration-00

Abstract

This document describes a new transport protocol TSMP, which seeks to support data transfer for the emerging usage paradigm of "single user, multiple device" in a TCP compatible manner. Through its novel proxy-based design, TSMP is able to retain the current client and server protocol operations of the legacy TCP protocol and TCP-based applications while placing new functions at the proxy.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

In this document, we design protocol solutions to an emerging usage scenario of "single user, multiple devices." In recent years, it has become increasingly popular 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 or iPad wherever (s)he goes. This emerging single-user, multi-device setting opens new venue for networking protocol design and operations.

TCP has been the dominant transport protocol for most Internet applications, and many popular applications such as web-based video streaming, and Instant messaging (e.g., MSN) are based on its operation. In the single-user, multiple-device context, there are two main design challenges. First, the protocol operations should support TCP-based data transfer among multiple devices of the same user. TCP sessions should seamlessly migrate among the devices owned by the same user. For example, a user uses instant messaging or video streaming on his laptop when he is in his office. When he walks out for lunch, he proceeds the ongoing messaging or video session via his iPhone or iPad. Second, users can continue to run the legacy TCP and applications with minimal changes at both sides of the client and the server while supporting the notion of single-user, multiple-device during data communication. This will enable reuse of most existing Internet applications. Existing protocols can achieve one of these two goals, but not both.

In this document, we describe a novel solution, called TCP Service Migration Protocol (TSMP), which supports "single-user, multi-device" TCP communications. The TCP connection is associated with the user and can seamlessly migrate among devices belonging to the same user. A key innovation in TSMP is the proxy bridging the client and the server in the current client-server communication model. The proxy offers two critical services of naming and TCP control/data plane functions. By carefully designing the proxy, TSMP is able to reuse existing TCP and TCP-based applications at both the client and the server without changes.

2. Terminology

This document uses the following terms to refer to the entities or functions that are required in service migration protocol. Readers are expected to be familiar with RFC 3753 "Mobility Related Terminology" [RFC3753].

Instant Messaging (IM): A form of real-time, direct, text-based chatting communication between two or more people.

TCP Service Migration Protocol (TSMP): A protocol that can be used to migrate an ongoing TCP service from one device to another. Both devices belong to the same owner.

TCP Service Migration Protocol Server (TSMPS): A system that maintains a global namespace, and performs namespace management and namespace resolution.

TCP Service Migration Protocol Proxy (TSMPP): A system that is interposed between the server and the client to support TSMP via coordinating control signaling and forwarding data.

TCP Service Migration Protocol Application (TSMPA): An application that is installed on each TSMP-enabled device.

Migration From Request (MFR): A control message that is used to issue the request of service migration by the originator device.

Migration To Request (MTR): A control message that is used to inform the target device regarding the request of service migration.

3. Single User Multiple Devices

This section illustrates an example scenario of single-user, multi-device, the requirements for our design, and the applications to which our protocol can be applied.

3.1. An Example Scenario

Bob has three networking devices: PC at home, Laptop in the office, and Smartphone that he uses while roaming. He chats with his friend, Alice, over an IM application using his smartphone while he is on his way back home. Meanwhile, Alice wants to share video clips with Bob using HTTP streaming from her web server running on her PC. After arriving at home, Bob switches both the IM session and the HTTP progressive downloading of the remaining videos to his home PC because of its larger screen size and network bandwidth. Bob then chats with Alice and watches the latter part of the video on his PC. Moreover, the service migration among Bob’s devices is transparent to Alice.

3.2. System Requirements

In the above scenario, we need to address the following two issues.

- o How to keep the intended connection open during migration and prevent the end that is not involved from perceiving the migration?

- o How to transfer TCP connection states from one device to another and make the overhead incur as little impact as possible on the connection?

The system needs to consider both control and data planes to support service migration. The control plane is used to coordinate the operation of service migration, including triggering the migration process, discovering the device to which the service is migrated, and informing the new device to accept the migration. During service migration, the data plane should be able to cache the transient packets that have been sent by the sender but have not been acknowledged, and make these packets as few as possible to reduce potential overhead. Once the migration is completed, it sends all cached packets to the new receiver and resumes the original TCP connection. Another important task is to avoid retransmission timeout to retain the same value of Congestion_Threshold. Service migration may happen from one device with low bandwidth to another with high bandwidth or from the latter to the former. In the former case, we need to make the Congestion_Threshold value as large as possible so that the sender’s Congestion_Window grows quickly with the slow-start algorithm, to reach the appropriate size associated with the larger bandwidth setting. If the Congestion_Threshold value becomes too small, the size of Congestion_Window would increase very slowly because it increases linearly with the additive increase/ multiplicative decrease mechanism once exceeding the threshold. However, the Congestion_Threshold cannot increase without data transmission; therefore, a feasible option is to retain the same value of Congestion_Threshold. As for the latter case, the Congestion_Window can shrink quickly to the appropriate size due to the multiplicative decrease mechanism.

4. TCP Service Migration Protocol

In this section, we first present the protocol architecture, and then describe our proxy-based solution to TCP migration.

4.1. Architecture

We employed a proxy-based solution to achieving TCP service migration. As shown in Figure 1, the TSMP architecture is composed of three main components: TSMP Proxy (TSMPP), TSMP Server (TSMPS), and TSMP Application (TSMPA). TSMPP is interposed between the client and the server to relay packets from either side to the other and mediate the sub-connections of each TCP connection. To avoid modification of the existing systems and applications, it collaborates with TSMPS and TSMPA to support the service migration process. TSMPA provides users an interface to make use of the TSMP service. It offers a channel for TSMPP to interact with the TCP-based applications at devices. TSMPS provides the service of DNS-like name resolution and enables the proxy to locate devices.

```
     +------------------------+       +------------------------------+
     |       TSMP Server      |       |      TSMP-enable Device      |
     | +--------------------+ |       | +--------------------------+ |
     | |     Resolution     | |       | |     TSMP Application     | |
     | |   Service Module   | |       | | +----------------------+ | |
     | +----------+---------+ |   +---+-+-|  TSMP Service Module | | |
     +------------+-----------+   |   | | +-----------+----------+ | |
                  |  +------------+   | +-------------+------------+ |
     +------------+--+-----------+    |               |              |
     | TSMP  +----+--+-+ +------+|    | +-------------+------------+ |
     | Proxy | Control | | Data ++----+-|  TCP-based Applications  | |
     |       |  Plane  | | Plane||    | +--------------------------+ |
     |       +---------+ +------+|    |                              |
     +---------------------------+    +------------------------------+
```

Figure 1: TSMP Architecture.

4.2. Proxy-based Solution

TSMPP consists of both the control plane and the data plane. The former coordinates the service migration process. The latter forwards packets between two ends, and emulates as a TCP sender to set up a new sub-connection to the new receiver upon TCP migration request.

4.2.1. Control Plane

The control plane coordinates the operation of service migration using two control messages: Migration From Request (MFR) and Migration To Request (MTR). MFR is always sent by TSMPA to request

TSMPP to migrate a TCP connection from the device where it resides to another device. It includes both the identity of the intended device and the information of the migrated connection so that TSMPP can resolve the device’s IP address by querying TSMPS and identify the connection. The other control message, MTR, is used by TSMPP to let TSMPA invoke its local application and set up a connection to TSMPP. TSMPP then hooks up this new sub-connection to the old sub-connection of the other end, thus recovering the migrated TCP connection.

4.2.2. Data Plane

TSMPP bridges between the two ends for each TCP connection by forwarding packets from one end to the other. Each connection is divided into two sub-connections that are glued by a mapping table in TSMPP. The mapping entry of a connection contains two-tuple of each end: IP address and port number. When TSMPP receives packets from one end’s sub-connection, it replaces the source and destination information with the TSMPP address and the other end’s address, respectively, and then forwards them to the other sub-connection.

4.3. TCP Migration

When TSMPP receives a MFR request, it starts the migration process of the requested TCP connection. The main concept is that, it temporarily halts the TCP flow until the connection between the new device and TSMPP is established, and then resumes it. The process hence consists of two phases: transient pause phase and resumption phase. The pause phase freezes the sending process, caches all outstanding packets that have not be forwarded by TSMPP, and keeps the value of Congestion_Threshold unchanged to prevent unnecessary congestion control invocations at the sender. The goal of the former two actions is to minimize transient loss and keep the connection open, whereas the last action seeks to decrease the overhead of increasing Congestion_Window to the appropriate size of the new sub-connection after the migration completes. In the resumption phase, TSMPP emulates a TCP end to set up a connection to the new device, flushes the cached packets to it, and then recovers the sending process. After the connection is resumed, TSMPP continues the forwarding process and the old sub-connection is halted.

4.3.1. Transient Pause Phase

This phase is launched once MFR is received by TSMPP. It does not end until the migration is complete. It is mainly composed of three tasks: advertising the size of the receiver’s window to be zero, stopping forwarding data packets but caching all of them, and responding to the zero-window probing.

In the TCP flow control mechanism, the receiver can advertise its window with the size zero to stop the sender from sending data.

The sender does not resume sending until the advertised window is larger than zero. We leverage this feature to stop the sending process by resetting the window size of the TCP headers to be zero in the ACK packets that are forwarded once this phase begins. TSMPP keeps on forwarding the ACK packets that acknowledge the data packets it has forwarded to the old receiver before this phase. The sender thus halts its sending process, and does the zero-window probing by sending at least one octet of new data periodically. Its purpose is to attempt recovery and ensure that re-opening of the window can be reliably reported. During the migration period, TSMPP should generate and send an ACK packet, which shows the next expected sequence number and zero window size, in response to each probe segment. Therefore, the TCP sender would allow the connection to stay open and temporarily freeze the sending process without shrinking the value of Congestion_Threshold. We can use the maximum sequence number of the cached packets plus one to be the expected sequence number.

Another task for this phase is to cache the transient packets that have not been forwarded. TSMPP starts to cache data packets and stops forwarding them once this phase begins. These cached data packets have been sent out by the sender so that the retransmission timeout would be triggered if they were not acknowledged. Accordingly, TSMPP needs to generate and send their ACK packets to the sender in advance on behalf of the new receiver. These ACKs should also contain the same information of the expected sequence number and the window size. TSMPP needs to ensure that it caches the data segments with all the sequence numbers between the expected sequence number and the acknowledge number of the last ACK packet that the old receiver sends. There may be a case that the old receiver does not acknowledge all its received packets before tearing down its connection. However, these packets would not be cached by TSMPP because they have been forwarded. Intuitively, TSMPP can just send ACKs to trigger retransmission at the sender and cache them, but the side effect is that Congestion_Threshold will be reduced. We may enable TSMPP to cache a certain amount of packets to handle this case no matter whether it is in the migration state or not. If there are still missing packets, relying on the retransmission would not be avoided. We can estimate the cache size based on half of the RTT between the sender and the receiver.

4.3.2. Resumption Phase

When the new device requests for a new connection due to its TSMPA’s invocation, the resumption phase begins. TSMPP emulates a TCP end to conduct three-way handshaking with the device, and starts to send its cached packets to it. As a TCP sender, TSMPP maintains some connection states: Congestion_Window, and Congestion_Threshold, etc. It uses the slow-start algorithm when the sending process is initialized or the connection times out, and employs the AIMD algorithm after Congestion_Window reaches Congestion_Threshold. TSMPP does not forward their ACK packets to the sender. After all cached packets are acknowledged, TSMPP resumes the sender's sending process by forwarding the new receiver's last ACK it receives. The transmission is thus recovered due to the last ACK with a non-zero receive window. TSMPP then returns to the normal forwarding phase, and discards the emulated TCP states. An issue we need to address is that the initial sequence number that is chosen at random may result in different sequence number systems between the old sub-connection and the new sub-connection. For this reason, TSMPP should add the mapping information of their sequence numbers into the mapping entry of this connection, and modify each packet's sequence number before forwarding it.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

The security aspects of the proxy-based applications also apply to this memo. TBC.

7. Informative References

[RFC3753] Manner, J. and M. Kojo, "Mobility Related Terminology", June 2004, <RFC 3753>.

Authors' Addresses

Chi-Yu Li
UCLA
4681 Boelter Hall
Los Angeles, CA 90095
USA

Phone: +1 323 528 1039
Email: lichiyu@cs.ucla.edu

Bojie Li
Huawei
Shenzhen,
P.R.C.

Phone: +86 755 2878 0808
Email: libojie@huawei.com

Chenghui Peng
Huawei
Shenzhen,
P.R.C.

Phone: +86 755 2878 0808
Email: pengchenghui@huawei.com

Wei Zhang
Huawei
Shenzhen,
P.R.C.

Phone: +86 755 2878 0808
Email: wendy.zhangwei@huawei.com

Songwu Lu
UCLA
4731C Boelter Hall
Los Angeles, CA 90095
USA

Phone: +1 310 794 9289
Email: slu@cs.ucla.edu