Computer Networks 105 (2016) 1–21

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Is multi-path transport suitable for latency sensitive traffic?

Kiran Yedugundla[a,*], Simone Ferlin[b], Thomas Dreibholz[b], Özgü Alay[b], Nicolas Kuhn[c,d], Per Hurtig[a], Anna Brunstrom[a]

[a] Dept. of Computer Science, Karlstad University, Karlstad, Sweden
[b] Simula Research Laboratory, Oslo, Norway
[c] IMT Télécom Bretagne, IRISA, Cesson-Sévigné, France
[d] Centre National d'Etudes Spatiales (CNES), Toulouse, France

A R T I C L E I N F O

Article history:
Received 3 December 2015
Revised 13 April 2016
Accepted 11 May 2016
Available online 21 May 2016

Keywords:
Internet
Latency
Multi-path communication
Transport protocols
MPTCP
CMT-SCTP

A B S T R A C T

This paper assesses whether multi-path communication can help latency-sensitive applications to satisfy the requirements of their users. We consider Concurrent Multi-path Transfer for SCTP (CMT-SCTP) and Multi-path TCP (MPTCP) and evaluate their proficiency in transporting video, gaming, and web traffic over combinations of WLAN and 3G interfaces. To ensure the validity of our evaluation, several experimental approaches were used including simulation, emulation and live experiments. When paths are symmetric in terms of capacity, delay and loss rate, we find that the experienced latency is significantly reduced, compared to using a single path. Using multiple asymmetric paths does not affect latency – applications do not experience any increase or decrease, but might benefit from other advantages of multi-path communication. In the light of our conclusions, multi-path transport is suitable for latency-sensitive traffic and mature enough to be widely deployed.

1. Introduction

Live and interactive applications are sensitive to latency, as the user experience is negatively affected when data is delayed. For instance, freezing a live video just 1% of the video duration is sufficient to turn away 5% of the viewers [1]. Similarly, a latency of 60 ms suffices to degrade user experience in Internet gaming [2]. Multiple ways of improving the user experience of latency sensitive applications are active subjects of research. However, as far as we know, a weakly explored area is to determine whether utilizing all available network interfaces at the end host could improve such experience. In recent times, deployed devices such as tablets and smartphones are often equipped with both Wireless LAN (WLAN) and cellular 3G or 4G interfaces.

Multi-path transmission has been proposed to guarantee better resilience to link failures and a better use of resources. For instance, consider a connection using two interfaces simultaneously; if one of the interfaces (or underlying links) fails, the transmission can simply continue over the other interface. In a single-interface scenario, the transmission would be stalled and maybe require a connection re-establishment. It has also been shown that simultaneous transmission of data over multiple interfaces can increase the throughput, due to capacity aggregation [3]. Even if multi-path protocols have been shown to be more resilient to link failures and able to aggregate capacity to provide increased throughput, the impact of using multiple paths on latency has not been thoroughly investigated.

This paper fills this gap by assessing whether multi-path approaches are suitable transport protocols for applications transmitting latency-sensitive traffic, e.g., video, gaming and web traffic. Recent efforts within the Internet Engineering Task Force (IETF) include designing Multi-path TCP (MPTCP) [4] extensions to TCP [5] to enable end-to-end connections to span multiple paths simultaneously. Similarly, Concurrent Multipath Transfer for SCTP (CMT-SCTP) [6–8] is an extension to the Stream Control Transmission Protocol (SCTP) [9], enabling simultaneous multi-path communication. We therefore evaluate their suitability to carry out latency sensitive traffic.

In our experiments we consider both symmetric multi-path communication (e.g. WLAN-WLAN) as well as asymmetric (e.g. WLAN-3G). For the actual evaluations we use a combination of simulations, emulations and real experiments to ensure a correct assessment.

The remainder of this paper is structured as follows. Section 2 presents an overview of CMT-SCTP and MPTCP, and how

* Corresponding author.
E-mail addresses: kiran.yedugundla@kau.se, kirayedu@kau.se (K. Yedugundla), ferlin@simula.no (S. Ferlin), dreibh@simula.no (T. Dreibholz), ozgu@simula.no (Ö. Alay), nicolas.kuhn@cnes.fr (N. Kuhn), perhurt@kau.se (P. Hurtig), annabrun@kau.se (A. Brunstrom).

http://dx.doi.org/10.1016/j.comnet.2016.05.008
1389-1286/

these protocols solve the core issues inherent in transport-level multi-path communication. Section 3 describes the applications used in our evaluation and their latency requirements. In Section 4, the experimental setup is detailed. Section 5 presents and explains the results obtained. In addition to that, Section 6 provides an in-depth discussion of the results. Section 7 discusses related work on multi-path transport. Finally, Section 8 concludes the paper and discusses possible future work in this area.

2. Multi-path transport

This section introduces CMT-SCTP and MPTCP, the current key multi-path transport protocols. The core issues of multi-path communication, and how these are addressed by CMT-SCTP and MPTCP, are then described.

2.1. CMT-SCTP

SCTP [9,10] is a transport protocol originally developed by the IETF Signaling Transport (SIGTRAN) Working Group [11], as part of an architecture to provide reliable and timely message delivery for Signaling System No. 7 (SS7) [12] telephony signaling information, on top of the Internet Protocol (IP) [13]. While motivated by the need to carry signaling traffic, SCTP was designed as a general purpose transport protocol on par with TCP [5] and UDP [14]. While SCTP can offer functionality similar to TCP, such as ordered and reliable transmission or congestion controlled transport, its options can be easily set so that SCTP rather features unordered transmission or multi-homing. This flexibility is one main advantage of SCTP as opposed to TCP.

The multi-homing feature of SCTP allows a single association (or connection) between two endpoints to combine multiple source and destination IP addresses. These IP addresses are exchanged and verified during the association setup, and each destination address is considered as a different path towards the corresponding endpoint. Using the Dynamic Address Reconfiguration protocol extension [15], it is also possible to dynamically add or delete IP addresses, and to request a primary-path change, during an active SCTP association.

While SCTP multi-homing [9,10] targets robustness and uses only one active path at a time, several researchers have suggested the concurrent use of all paths for sending data. Budzisz et al. [16] provides a survey of these approaches. In this paper, we consider the most complete of these proposals, Concurrent Multipath Transfer for SCTP (CMT-SCTP) [6–8]. CMT-SCTP improves the internal buffer management procedures of SCTP, transmission over multiple paths and reordering with its single sequence-number space. Assuming disjoint paths, CMT-SCTP applies the original SCTP congestion control [9] for each path independently.

2.2. MPTCP

Multi-Path TCP (MPTCP) [17] is a set of extensions to TCP [5,18] developed by the IETF MPTCP working group [19] to enable simultaneous use of multiple paths between endpoints. The motivation behind MPTCP is more efficient resource usage and improved user experience through improved resilience to network failure and higher throughput.

To use the MPTCP extensions the initiator of a connection appends a “Multipath Capable” (MP_CAPABLE) option in the SYN segment, indicating its support for MPTCP. When the connection is established, it is possible to add one TCP flow, or subflow, per available interface to this connection by using a “MPTCP Join” (MP_JOIN) option in the SYN segment. Once the MPTCP connection has been fully established, both end hosts can send data over any of the available subflows.

While MPTCP transparently divides user data among the subflows, simultaneous transmission may cause connection-level packet reordering at the receiver. To handle such reordering, two levels of sequence numbers are used. Apart from the regular TCP sequence numbers that are used to ensure in-order delivery at subflow level, MPTCP uses a 64-bit data sequence number that spans the entire MPTCP connection and can be used to order data arriving at the receiver. To ensure fairness [20] on bottleneck links shared by subflows of a MPTCP flow and other TCP flows, MPTCP extends the standard TCP congestion control. Running existing TCP congestion control algorithms independently would give MPTCP connections more than their fair share of the capacity if a bottleneck is shared by two or more of its subflows. To solve this MPTCP uses a coupled congestion control [21] that links the increase functions of each subflows’ congestion control and dynamically controls the overall aggressiveness of the MPTCP connection. The coupled congestion control also makes resource usage more efficient as it steers traffic away from more congested paths to less congested ones.

2.3. Core issues

This section presents the core issues that are related to the use of multiple paths and how they are addressed by CMT-SCTP and MPTCP.

2.3.1. Path management

As shown in Fig. 1, a path is a sequence of links between a sender and a receiver [4], over which it is possible to open a subflow. A multi-path protocol must define a path management strategy. The strategy needs to find suitable paths to open subflows over and decide whether one or more subflows should be opened over a specific path. For short or extremely time-sensitive flows, the choice of path for the initial connection establishment might be very important. For example, if (i) two paths (p_1 and p_2) are available, (ii) both paths have the same capacity and (iii) the RTT of p_1, r_1, is significantly higher than the RTT of p_2, r_2, (e.g. $r_1 > 10 \times r_2$), then whether the first subflow will be opened over p_1 or p_2 would seriously impact the latency. The number of subflows to open over a path is a problem that is not very well studied. While the Linux implementation of MPTCP supports this using its ndiffports path manager, as described later in this section, it is often regarded as unnecessary to open more than one subflow per path as they typically would traverse the same links and compete for the same network resources. However, in some specific environments, e.g. datacenters, the network might conduct load balancing between subflows, routing them over disjoint subpaths. In such situations there might be benefits of creating several subflows per path, as shown in [22].

For CMT-SCTP a path is defined by the destination IP address and port number. To manage paths, CMT-SCTP employs a simple

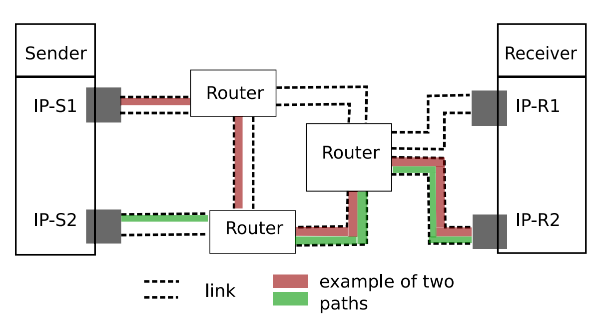

Fig. 1. Definition of a path as a sequence of links between a sender and a receiver.