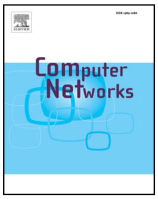
Computer Networks 91 (2015) 548–563

Contents lists available at ScienceDirect

Computer Networks

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



Constraint-based proactive scheduling for MPTCP in wireless networks

Bong-Hwan Oh, Jaiyong Lee*

UbiNet Lab, Department of Electrical and Electronic Engineering, Yonsei University, 134 Shinchon-Dong, Seodaemoon-Gu, Seoul 120-749, Republic of Korea

ARTICLE INFO

Article history:
Received 20 March 2015
Revised 8 July 2015
Accepted 3 September 2015
Available online 10 September 2015

Keywords:
Multipath TCP
Packet scheduling
Buffer blocking problem

ABSTRACT

Multipath TCP (MPTCP) is one of the leading protocols that support multipath operation in a transport layer. However, depending on the network and the receiver buffer, the original MPTCP can experience throughput degradation, underutilizing the network capacity compared to the regular TCP. Furthermore, MPTCP can result in a large packet interval. In this paper, we propose a new scheduling scheme for MPTCP that performs packet scheduling according to the receiver buffer and network delay. Our scheme estimates out-of-order packets according to performance differences between subflows and assigns data packets to subflows by comparing the estimated out-of-order packets and the buffer size. Moreover, our scheme can adjust the trade-off between throughput and delay performance using a delay constraint. We implement the proposed scheduling in the Linux kernel and evaluate its performance over a virtual network framework using NS-3 and real networks. The results show that the proposed scheduling scheme performs efficient packet transmission regardless of the performance differences of multiple paths and buffer size. Moreover, the proposed scheduling can complement and cooperate with an existing non-scheduling-based solution.

1. Introduction

New approaches for using access networks have been proposed with the rapid advancement of portable devices. Unlike older devices, which used only a selected network, current portable devices are equipped with multiple radio interfaces that can select and use a specific access network. In addition to the selection of a specific network, new initiatives to concurrently use multiple access networks seek to provide better network services. One instance such work is the announcement by SK Telecom and KT (the main network operators in South Korea) that they will launch the commercial services Hybrid Network Integration System (SK telecom) and KT GIGA path, which use long-term evolution (LTE) and WiFi networks simultaneously.

Unfortunately, the current dominant transport layer protocols such as TCP and UDP cannot use and control multiple paths autonomously. The main problem with the existing protocols is that applications cannot change a communication session to another path. Thus, in order to manage multiple paths with current protocols, applications should perform multiple interface management, which increases the workload of applications.

One promising solution to this problem is the application of multipath TCP (MPTCP) [1], which is a TCP extension version that is being standardized by the Internet Engineering Task Force (IETF). MPTCP provides concurrent data transmission over multiple paths and supports compatibility with a single TCP. Thus, by using MPTCP, applications can utilize multiple interfaces without additional burden.

---

* Corresponding author. Tel.: +82 221232873.
E-mail addresses: crusader27@yonsei.ac.kr (B.-H. Oh), jyl@yonsei.ac.kr (J. Lee).

http://dx.doi.org/10.1016/j.comnet.2015.09.002
1389-1286/

One of the concerns caused by the extension from single TCP to MPTCP is packet scheduling for multiple paths. This affects the reordering problem, which is one of the challenges of multipath transmission. Thus, in order to provide efficient transmission in MPTCP, a scheduling scheme should be considered in order to maintain the number of reordering packets, which influences the overall transmission performance. Currently, two basic scheduling cases have been proposed to fully utilize all paths [1,3]. The first is a round-robin scheduler that utilizes subflows in sequence. The second is lowest-round-trip time (RTT)-first scheduling, which first assigns packets to the fastest subflow until its congestion window is filled with data, after which packets are then allocated to the other subflows. However, these cases present problems when applied to current wireless networks, and the results might not be viable. The current scheduling methods can generate a lower throughput (goodput) performance than single TCP when MPTCP with a limited buffer simultaneously uses multiple paths that have different delay characteristics [4,6]. This phenomenon occurs frequently, because promising wireless networks, including 3G (LTE) and WiFi, have quite different delay performances, and mobile devices do not always have sufficient memory. Furthermore, a large delay difference can generate a large packet interval, which affects the quality of service (QoS) of real-time applications.

In this paper, we propose a new scheduling scheme for MPTCP, which performs packet scheduling according to the receiver buffer and the network delay. The proposed scheme can prevent performance degradation by using multiple asymmetric paths and sufficiently utilizes the capacity of multiple symmetric paths. As a result, MPTCP with the proposed scheduling can perform efficient packet transmission regardless of the performance difference of multiple paths and buffer size. For the delay performance, the proposed scheme can maintain the packet interval by restricting the utilization of multiple paths. Moreover, the proposed scheduling scheme can be used with other solutions that are not scheduling-based and is sufficiently simple for practical implementation.

The rest of this paper is organized as follows. Section 2 describes a data transfer problem in MPTCP and discusses related work. Section 3 introduces the proposed scheduling scheme. Section 4 presents the results of performance evaluation, robustness of the proposed scheme, and cooperation with an existing solution. Section 5 presents the measurement results in real networks. Finally, we offer our conclusions in Section 6.

2. Background and related work

Fundamentally, MPTCP consists of two sub-layers: the MPTCP layer and the subflow layer. The MPTCP layer provides reliability and application compatibility by preserving the TCP-like semantics of global ordering of the application data, whereas the subflow layer provides network compatibility by appearing and behaving as a TCP flow in the network [2]. First, the MPTCP layer establishes the connection over the subflow layer and discovers available paths. The MPTCP layer receives a byte stream from an application and allocates application data to the subflows. Each subflow transmits the allocated data segment and performs congestion control according to its network condition. Although the data segment allocated to a subflow is managed in the subflow, all of the application data is managed using a single buffer and data sequence number (DSN) at the MPTCP connection in order to support ordered and reliable delivery. In other words, although each subflow operates independently regarding network-dependent functions, it can affect the other subflows through their combination with application-dependent functions.

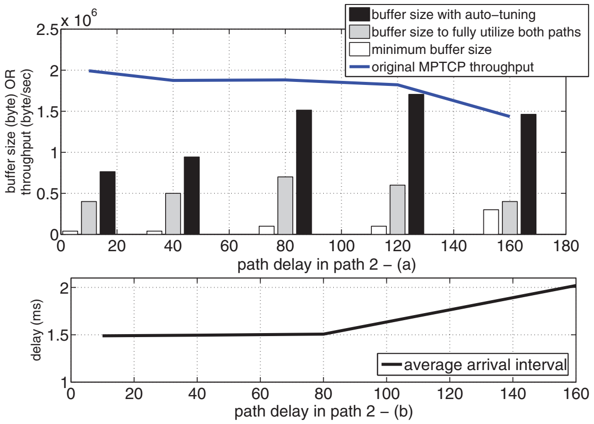


Fig. 1. Buffer and arrival interval consideration in original MPTCP.

Flow control is one of the application-dependent functions that are influenced by all subflows according to an extension from single TCP to MPTCP. Basically, the flow control operation in single TCP is the solution to one of the problems caused by a difference in processing performance between end devices. However, in MPTCP, flow control can be performed due to the difference in network delay performance, because the packets transmitted to a slow path hinder transmission in a fast path. Although packets transmitted to a fast path are sent later than packets transmitted to other paths, they can be out-of-order due to the difference in delay among multiple paths, which reduces the receiver window (RWND). Consequently, the sender performs flow control, which interrupts transmissions on the fast path, which is called the buffer blocking problem in this paper. Although this phenomenon does not occur when MPTCP has a sufficient buffer [4], the required buffer size can be extremely large when MPTCP simultaneously uses multiple paths that have different delay characteristics [5,6].

One of the main considerations for solving the buffer blocking problem is the required buffer size needed to utilize multiple paths. Fig. 1a shows the effect of delay asymmetry on MPTCP buffer size. In this simulation environment, the original MPTCP uses two paths: path 1 has a fixed performance (8 Mbps bandwidth and 10 ms path delay), and path 2 has a fixed bandwidth (8 Mbps) but a path delay that varies between 10 ms and 160 ms. The original MPTCP does not employ additional buffer-blocking solutions such as opportunistic retransmission or penalization [6,7]. The gray bar indicates the required buffer size needed to fully utilize both paths, and the white bar shows the minimum buffer size needed to achieve the same or better throughput than single TCP over the fastest path. The black bar indicates the