ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths

Yeon-sup Lim
University of Massachusetts Amherst
ylim@cs.umass.edu

Erich M. Nahum
IBM Research
nahum@us.ibm.com

Don Towsley
University of Massachusetts Amherst
towsley@cs.umass.edu

Richard J. Gibbens
University of Cambridge
richard.gibbens@cl.cam.ac.uk

ABSTRACT

Multi-Path TCP (MPTCP) is a new standardized transport protocol that enables devices to utilize multiple network interfaces. The default MPTCP path scheduler prioritizes paths with the smallest round trip time (RTT). In this work, we examine whether the default MPTCP path scheduler can provide applications the ideal aggregate bandwidth, i.e., the sum of available bandwidths of all paths. Our experimental results show that heterogeneous paths cause under-utilization of the fast path, resulting in undesirable application behaviors such as lower video streaming quality than can be obtained using the available aggregate bandwidth. To solve this problem, we propose and implement a new MPTCP path scheduler, ECF (Earliest Completion First), that utilizes all relevant information about a path, not just RTT. Our results show that ECF consistently utilizes all available paths more efficiently than other approaches under path heterogeneity, particularly for streaming video.

1 INTRODUCTION

One significant factor that affects MPTCP performance is the design of the path scheduler, which distributes traffic across available paths according to a particular scheduling policy. The default path scheduler of MPTCP is based on round trip time (RTT) estimates, that is, given two paths with available congestion window space, it prefers to send traffic over the path with the smallest RTT. While simple and intuitive, this scheduling policy does not carefully consider path heterogeneity, where available bandwidths and round trip times of the two paths differ considerably. This *path heterogeneity* is common in mobile devices with multiple interfaces [2, 4, 5, 9, 12] and can cause significant reorderings at the receiver-side [1–3, 7, 13]. To prevent this, MPTCP includes opportunistic retransmission and penalization mechanisms along with the default scheduler [10]. In long-lived flows, e.g., large file transfer, MPTCP is able to enhance performance using these mechanisms. However, a large number of Internet applications such as Web browsing and video streaming usually generate traffic which consists of multiple uploads/downloads for relatively short durations. We find that in the presence of path heterogeneity, the default MPTCP scheduler is unable to efficiently utilize some paths with such a traffic pattern. In particular it does not fully utilize the highest bandwidth paths, which should be prioritized to achieve the highest performance and lowest response time.

In this work, we propose a novel MPTCP path scheduler to maximize fast path utilization, called ECF (Earliest Completion First). To this end, ECF monitors not only RTT estimates, but also the current subflow bandwidths (i.e., congestion windows) and the amount of data available to send (i.e., the send buffer). By determining whether using a slow path for the injected traffic will cause faster paths to become idle, ECF more efficiently utilizes the faster paths, maximizing throughput, minimizing download time, and reducing out-of-order packet delivery. Our experimental results demonstrate that ECF successfully avoids undesirable idle periods, achieving greater throughput with higher path utilization than the default scheduler. At the same time, it performs as well as other schedulers under symmetric path conditions.

2 THE EFFECT OF PATH HETEROGENEITY

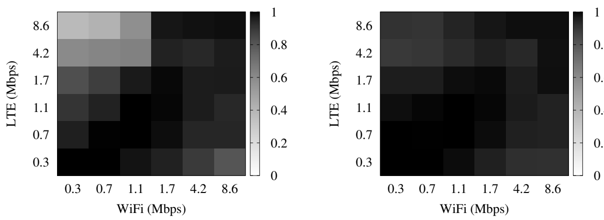
We examine the effect of path heterogeneity on application performance using adaptive video streaming, since it is currently one of the dominant applications in use over the Internet [11]. We measure the average video bit rate obtained by an Android DASH (Dynamic Adaptive Streaming over HTTP) streaming client while limiting the bandwidth of the WiFi and LTE subflows on the server-side using the Linux traffic control utility tc [8]. The streaming client uses a state-of-art adaptive bit rate selection (ABR) algorithm [6]. The choice of ABR does not significantly affect the results in this experiment as we use fixed bandwidths for each interface. The opportunistic retransmission and penalization mechanisms are enabled by default. Each experiment consists of five runs, where a run consists of the playout of the 20 minute video of which available resolutions are 144p to 1080p. Table 1 presents the bit rates corresponding to each resolution. We choose bandwidth amounts slightly larger than those listed in Table 1, i.e., {0.3, 0.7, 1.1, 1.7, 4.2, 8.6} Mbps, to ensure there is sufficient bandwidth for that video encoding.

Figure 1(a) presents the ratio of the average bit rate achieved versus the ideal average bit rate available, based on the bandwidth combinations, when using the default MPTCP path scheduler. The figure is a grey-scale heat map where the darker the area is, the closer to the ideal bit rate the streaming client experiences. The closer the ratio is to one, the better the scheduler does in achieving the potential available bandwidth. The values are averaged over five runs. In a streaming workload, we define the ideal average bit rate as the minimum of the aggregate total bandwidth and the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGMETRICS'17, Urbana-Champaign, Illinois, USA
© 2017 ACM. 123-4567-24-567/08/06...$15.00
DOI: 1234567898

Resolution	144p	240p	360p	480p	760p	1080p
Bit Rate (Mbps)	0.26	0.64	1.00	1.60	4.14	8.47

Table 1: Video Bit Rates vs. Resolution



(a) Default (b) ECF

Figure 1: Ratio of Measured Average Bit Rate vs. Ideal Average Bit Rate (darker is better)

bandwidth required for the highest resolution at that bandwidth. For example, in the 8.6 Mbps WiFi and 8.6 Mbps LTE pair (the upper right corner in Figure 1(a)), the ideal average bit rate is 8.47 Mbps, since the ideal aggregate bandwidth ($8.6+8.6 = 17.2$ Mbps) is larger than the required bandwidth for the highest resolution of 1080p (8.47 Mbps). Since the full bit rate is achieved, the value is one and the square is black.

Figure 1(a) shows that, when significant path heterogeneity exists, the streaming client fails to obtain the ideal bit rate. For example, when WiFi and LTE provide 0.3 Mbps and 8.6 Mbps, respectively (the upper left box in Figure 1), the streaming client retrieves 480p video chunks, which requires only 2 Mbps, even though the ideal aggregate bandwidth is larger than 8.47 Mbps. Thus, the value is only 25% of the ideal bandwidth and the square is light grey. This problem becomes even more severe when the primary path (WiFi) becomes slower (compare the 0.3 Mbps & [0.3 − 8.6] Mbps and 8.6 Mbps & [0.3 − 8.6Mbps] pairs), as shown by the grey areas in the upper left and lower right corners.

3 ECF SCHEDULER

To solve the performance degradation problem with path heterogeneity, we propose a new MPTCP path scheduler, called ECF (Earliest Completion First). An MPTCP sender stores packets both in its connection-level send buffer and in the subflow level send buffer (if the packet is assigned to that subflow). Assume that there are k packets in the connection level send buffer, which have not been assigned (scheduled) to any subflow. If the fastest subflow in terms of RTT has available CWND, the packet can simply be scheduled to that subflow. If the fastest subflow does not have available space, the packet needs to be scheduled to the second fastest subflow.

We denote the fastest and the second fastest subflows as x_f and x_s, respectively. Let RTT_f, RTT_s and $CWND_f$, $CWND_s$ be the RTTs and CWNDs of x_f and x_s, respectively. If the sender waits until x_f becomes available and then transfers k packets through x_f, it will take approximately $RTT_f + \frac{k}{CWND_f} \times RTT_f$, i.e., the waiting and transmission time of k packets. Otherwise, if the sender sends some packets over x_s, the transmission will finish after RTT_s with

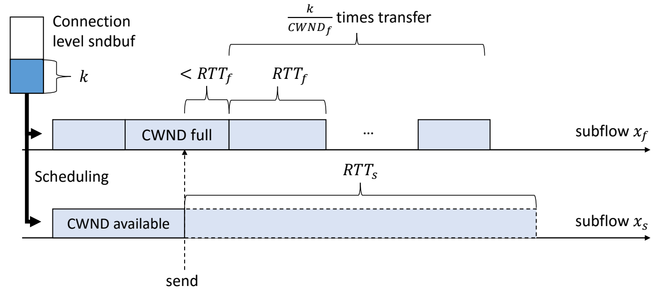


Figure 2: The case for waiting for the fast subflow

or without completing k packet transfers. Thus, as shown in Figure 2, in the case of $RTT_f + \frac{k}{CWND_f} \times RTT_f < RTT_s$, using x_f after it becomes available can complete the transmission earlier than using x_s immediately. If $RTT_f + \frac{k}{CWND_f} \times RTT_f \geq RTT_s$, there are sufficient number of packets to send, so that using x_s at that moment can decrease the transmission time by utilizing more bandwidth than just by using x_f.

ECF checks the above inequality to decide whether it will wait for x_f or immediately use x_s. Figure 1(b) shows that ECF successfully enables the streaming client to obtain average bit rates closest to the ideal average bit rate, and does substantially better than the default when paths are not symmetric.

REFERENCES

[1] Y.-C. Chen, Y.-S. Lim, R. J. Gibbens, E. Nahum, R. Khalili, and D. Towsley. A measurement-based study of Multipath TCP performance in wireless networks. In *Proc. of ACM IMC*, pages 455–468, Nov 2013.
[2] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan. WiFi, LTE, or both? Measuring multi-homed wireless Internet performance. In *Proc. of ACM IMC*, 2014.
[3] S. Ferlin, O. Alay, O. Mehani, and R. Boreli. BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks. In *Proc. of IFIP Networking*, pages 1222–1227, 2016.
[4] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan. MP-DASH: Adaptive video streaming over preference-aware multipath. In *Proc. of ACM CoNEXT*, 2016.
[5] J. Huang, Q. Feng, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *Proc. of ACM MobiSys*, pages 225–238, 2012.
[6] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proc. of ACM SIGCOMM*, pages 187–198, 2014.
[7] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli. DAPS: Intelligent delay-aware packet scheduling for multipath transport. In *Proc. of IEEE ICC*, pages 1222–1227, 2014.
[8] Linux Foundation. Linux advanced routing and traffic control. http://lartc.org/howto/.
[9] A. Nikravesh, Y. Goo, F. Qian, Z. M. Mao, and S. Sen. An in-depth understanding of multipath TCP on mobile devices: Measurement and system design. In *Proc. of ACM MobiCom*, 2016.
[10] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley. How hard can it be? Designing and implementing a deployable multipath TCP. In *Proc. of USENIX NSDI*, pages 399–412, 2012.
[11] Sandvine. Global Internet phenomena report, Latin America and North America 2016. https://www.sandvine.com/downloads/general/global-internet-phenomena/2016/global-internet-phenomena-report-latin-america-and-north-america.pdf.
[12] J. Sommers and P. Barford. Cell vs. wifi: on the performance of metro area mobile connections. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 301–314. ACM, 2012.
[13] F. Yang, P. Amer, and N. Ekiz. A scheduler for multipath TCP. In *Proc. of ICCCN*, pages 1–7, 2013.