Computer Communications 80 (2016) 82–100

Contents lists available at ScienceDirect

Computer Communications

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

Living with congestion: Digital Fountain based Communication Protocol

Sándor Molnár, Zoltán Móczár*, Balázs Sonkoly

High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary

ARTICLE INFO

Article history:
Received 5 August 2014
Revised 13 January 2016
Accepted 15 January 2016
Available online 2 February 2016

Keywords:
Fountain coding
Congestion control
Transport protocol
Network architecture
Performance evaluation

ABSTRACT

In this paper, we propose a networking paradigm built upon a fountain code based data transfer mechanism. We advocate that, instead of controlling the congestion as it is applied in recent Internet by the Transmission Control Protocol (TCP), we can utilize congestion and neglect control algorithms with all of their drawbacks. We present our envisioned network architecture relying on a novel transport protocol called *Digital Fountain based Communication Protocol (DFCP)* and highlight some potential application areas. We have implemented DFCP in the Linux kernel and we provide its validation results gained from three different testing platforms including our laboratory testbed, the Emulab network emulation environment and the ns-2 network simulator. Moreover, we present and discuss a comprehensive performance evaluation of DFCP in comparison with widely used TCP versions.

1. Introduction

A fearful phenomenon called *congestion* has been observed from the beginning of computer networks when an aggregate demand for a resource exceeds its available capacity. Congestion can result in long delays, lost packets and consequently degraded quality of the service and wasted resources of the network. The phenomenon became even more dreadful in the early days of the Internet by causing *congestion collapse* [1] when the increase in network load led to the decrease of useful work in the entire network. It was the reason to introduce *congestion control*, which is a set of mechanisms to limit the demand-capacity mismatch by controlling traffic sources when such a mismatch occurs.

In the history of the Internet, closed-loop congestion control was the successful paradigm to avoid congestion collapse and the related performance degradation due to the overload of network resources. Congestion control is mostly performed by the *Transmission Control Protocol (TCP)*, which transports more than 80% of Internet traffic. The basic idea behind the congestion control mechanism of TCP is that the sender gradually increases the transmission rate until a packet loss is detected, and when it happens, the sending rate is cut in half starting the cycle again. Packets can be lost due to both congestion and environmental factors (e.g. lossy wireless links), which is typically indicated by an explicit feedback signal, or inferred by the sender from the reception of three duplicate acknowledgments. In this paper, we focus on this traditional loss-based approach, however, we note that some TCP variants adjust their transmission rate by measuring round-trip delay variations, or use hybrid solutions. The success of TCP was not even questioned until the fast development of networks, mobile devices and user applications resulted in heterogeneous and complex environments over the last decades. In order to fit these changes, significant research was carried out to further develop TCP, and therefore, several different TCP versions have been proposed [2–5]. However, it turned out that it will be very difficult to modify TCP to work efficiently as a universal transport protocol.

In this paper, we propose a different data transfer paradigm, namely, *instead of avoiding congestion we advocate to allow it*. The main idea is that it might be possible to perform efficient communication even in the presence of congestion. First, we enable *sources to transmit their data at the maximum possible rate*, which will shift the operation to the *overloaded regime*. Theoretically, such a solution can be considered as the most efficient one, because the network would always be fully utilized by hosts sending at maximal rates, and therefore, each additional capacity would immediately be consumed. In practice, however, this greedy behavior can lead to bandwidth waste that needs to be addressed by a proper rate limitation mechanism (see Sections 3.2 and 7.6 for the discussion). Moreover, we suggest to apply *efficient digital fountain based coding schemes* for encoding the application-level data. The proposed approach implies that packet loss becomes inconsequential, which can considerably simplify network routers

* Corresponding author. Tel.: +36 205215496.
E-mail addresses: molnar@tmit.bme.hu (S. Molnár), moczar@tmit.bme.hu (Z. Móczár), sonkoly@tmit.bme.hu (B. Sonkoly).

http://dx.doi.org/10.1016/j.comcom.2016.01.008
0140-3664/

and can result in highly reduced buffer sizes closely aligned to the concept of all-optical networking. Concerning stability, we can also expect improvements since transmitting at the maximum possible rate would result in more predictable traffic patterns by avoiding the high extent of rate variation often seen in TCP transmissions. This property would make network design and dimensioning easier tasks as well. Moreover, we suggest to use *fair schedulers* in network nodes to ensure fairness among competing flows. Since simple scalable schedulers are already available in routers, the seamless deployment of the concept can be feasible.

In order to take research on alternative data transfer paradigms forward, we studied the possibility of applying a *fountain code* [6] based transport protocol instead of the congestion control based TCP. To the best of our knowledge, such a transport protocol has not been developed and implemented with a comprehensive performance evaluation study yet, thus we made our prototype implementation and carried out a thorough analysis. First, we introduced the *Digital Fountain based Communication Protocol (DFCP)* in [7] with some preliminary analytical, simulation and testbed results. In this paper, the whole idea of the new architecture with a detailed description and performance evaluation of DFCP is presented. Our intention is to deliver the message that the proposed transport mechanism can be a potential alternative to TCP for future networks in several application areas.

In summary, the main contributions of this paper are

- the introduction of our envisioned network architecture and data transfer paradigm with discussions on possible future applications and challenges,
- a detailed description of our proposed transport protocol (DFCP) including both design and implementation aspects,
- the validation of our prototype implementation performed on multiple platforms including our laboratory testbed, the Emulab network emulation environment [8] and the ns-2 network simulator [9],
- as well as a comprehensive performance evaluation of DFCP in comparison with TCP
  - on both simulation and testbed platforms,
  - on multiple network topologies,
  - by investigating two widely used TCP versions in various network conditions and
  - focusing on important performance metrics such as goodput, flow completion time and fairness.

The paper is organized as follows. First, in Section 2 we give a brief overview about the evolution of TCP and digital fountain codes. We also review the recent ideas proposed in the field of data transfer methods and protocols. In Section 3, we introduce and discuss the envisioned network architecture built upon our newly developed transport protocol. Section 4 gives a detailed description of DFCP including the main design principles and implementation issues. In Section 5, we describe the evaluation methodology used for validation and comparative performance analysis. In Section 6, we present a multi-platform validation framework and confirm the operability of our protocol. A comprehensive performance evaluation study of DFCP and different TCP versions is presented in Section 7, then the future applications and challenges are discussed in Section 8. Finally, Section 9 concludes the paper.

2. Related work

2.1. The evolution of TCP

TCP is a connection-oriented transport protocol that provides reliable data transfer in end-to-end communication. It means that lost packets are retransmitted, and therefore, each sent packet will be delivered to the destination. One of the most important features of TCP is its congestion control mechanism, which is used to avoid congestion collapse by determining the proper sending rate and to achieve high performance. TCP maintains a congestion window that controls the number of outstanding unacknowledged data packets in the network. Continuously evolving network environments have made significant research demand on designing more and more efficient transport protocols in the last decades. As a result, several TCP versions have been developed in order to fit the ever-changing requirements of communication networks.

In this paper, we investigate two widely used TCP variants, namely TCP Cubic [10] and TCP NewReno with SACK [11]. In the case of TCP Reno, a lost packet is detected and retransmitted when triple duplicate acknowledgments are received or a timeout event occurs at the sender. TCP Reno is effective to recover from a single packet loss, but it still suffers from performance problems when multiple packets are dropped from a window of data [12]. TCP NewReno is a slight modification of TCP Reno intended to improve its performance when a burst of packets is lost [11]. TCP Cubic is an enhanced version of its predecessor, BIC TCP [13]. BIC TCP was originally designed to solve the well-known RTT (round-trip time) unfairness problem by combining two schemes called additive increase and binary search. TCP Cubic simplifies the window control of BIC and it applies a cubic function in terms of the elapsed time since the last loss event, which provides good stability and scalability [10].

Beyond the protocols described above, many other solutions have been worked out to improve the performance of traditional TCP. One of the main issues is that it takes a long time to make a full recovery from packet loss for high-bandwidth, long-distance connections, because the congestion window builds up very slowly. In order to cope with this limitation, HighSpeed TCP (HSTCP) [14] was proposed to achieve better performance on high-capacity links by modifying the congestion control mechanism of TCP for use with large congestion windows. Scalable TCP (STCP) [15] applies a multiplicative increase and multiplicative decrease algorithm to obtain performance improvement in high-speed networks and it can also guarantee the scalability of the protocol. TCP Westwood [16] is a sender-side modification of the congestion control mechanism that improves the performance of TCP Reno both in wired and wireless networks. The main problem is that Reno cannot distinguish between random and congestion losses, thus equally reacts to them. In fact, TCP Westwood shows moderate sensitivity to random errors, therefore the improvement is the most significant in wireless networks with lossy links. FAST TCP [17] is a congestion avoidance algorithm especially targeted for long-distance, high-latency links. FAST determines the current congestion window size based on both round-trip delays and packet losses over a path. The algorithm estimates the queuing delay of the path using RTTs and if the delay falls below a threshold, it increases the window aggressively. If it gets closer to the threshold, the algorithm slowly reduces the increasing rate. MultiPath TCP (MPTCP) [18] is a recent approach for enabling the simultaneous use of several IP addresses or interfaces by a modification of TCP that presents a regular TCP interface to applications, while in fact spreading data across several subflows.

Traditional TCP and its variants were primarily designed for wired networks, but emerging wireless networks and the increasing demands motivated researchers to develop new versions and optimize them for different network environments. The performance issues experienced in such environments stem from the unique characteristics of wireless links and the packet loss model used by TCP. The problems manifest in various applications as degradation of throughput, inefficiency in network resource utilization and excessive interruption of data transmissions. Modification of standard TCP to remedy its deficiency in wireless