Multimedia Tools and Applications (2021) 80:16763–16780
https://doi.org/10.1007/s11042-020-08856-z

An intelligent long-lived TCP based on real-time traffic regulation

Mohammad Al Shinwan[1] · Laith Abualigah[1] · Nguyen Dinh Le[2] · Chulsoo Kim[2] · Ahmad M. Khasawneh[1]

Received: 4 October 2019 / Revised: 28 February 2020 / Accepted: 13 March 2020 /
Published online: 24 April 2020

Abstract

Transmission control protocol (TCP) transaction is one of the chief protocols of Internet protocols. It is divided into two categories of data flow. (1) A small fraction of the TCP connections carried a small portion of the traffic called short-lived TCP. (2) A small fraction of the remaining TCP connections carried a large portion of the traffic called long-lived TCP. The main problem here is the transmission time; other data usually harm its flows, such as User Datagram Protocol (UDP) or short-lived TCP, and cause the unfairness in the network. In this paper, a novel framework is proposed to improve network throughput and to reduce the impact of long-lived TCP to other data flows. In this framework, each TCP connection passed by an edge network device and it will be observed for determining the long-lived TCP flows. Then, the detected long-lived TCP flow will be regulated based on predicting real-time traffic levels. Moreover, to highlight the benefits of the proposed framework, an analytical model is proposed to compare the proposed framework with the conventional TCP in terms of network performance. Experiments are conducted using the ns-2 benchmark in order to verify the results of the analytical model. The results showed that the analytical outcomes are promising and matched well with the outcomes of the ns-2 experiments. In the case of a high error rate, the proposed framework achieves higher reliability and reveals lower resource consumption.

Keywords Long-lived TCP · Isolation · Segment-by-segment · Network performance · A novel framwork

1 Introduction

The transmission control protocol (TCP) conveys the vast majority of the traffic with more than 90% of bytes. The reliability of data transmission keeps TCP remains the dominating transport protocol in the internet network [14, 27]. However, with the increase of applications

✉ Mohammad Al Shinwan
mohmdsh@aau.edu.jo

Extended author information available on the last page of the article

running on top of TCP and the demand for a high-speed network, TCP reveals some shortcomings such as low network throughput due to slow ramp-up mechanism or unfairness in network resource sharing with other data flows [11, 25]. For these reasons, researchers are encouraged to investigate TCP in more detail in order to provide a better solution for solving these existed problems [3, 4, 16, 20, 25].

In order to understand the root causes of unfairness problem, the complexity of TCP traffic is needed to be classified. In this study, TCP traffic is divided into short-lived and long-lived TCP based on their size of data and their life transmission time [23, 32]. In another study, by investigating the lifetime and transfer size of each TCP flow, L. Guo and I. Matta discovered TCP flows follow a heavy-tailed distribution in which only a small percentage (e.g., less than 20%) of flows are long-lived TCP (e.g., more than 20 packets), but they carry more than 80% of the total traffic in bytes [6, 17]. The vulnerability of long-lived TCP can state that the unfairness problem flows in sharing network resources [5]. Because of its significant amount of data, long-lived TCP flows need a long transmission time to finish its transaction. Hence, it is usually affected by other flows in network resource sharing. In detail, UDP flows usually occupy more than their fair share of the bandwidth. Short-lived TCP affects long-lived TCP by reducing its network throughput by up to 10% as the study in [9, 15, 34].

The basis of TCP congestion control lies in the additive increase multiplicative decrease mechanism (AIMD) [10]. The congestion window is halved for every window containing a packet loss and is increased roughly by one segment per round-trip time (RTT). By this mechanism, standard TCP is successful at low speeds; however, it is not suitable for high-speed communication due to its slow grow-up of congestion window size when congestion occurs. In other words, standard TCP is not efficient for high speeds of data communication, which is currently a high demand for all TCP applications [31, 35]. For these reasons, it is crucial to develop a new transport mechanism for improving network performance and reduce the impaction between long-lived TCP flows and other network flows [30].

In this paper, a new transport mechanism is proposed with a novel mechanism to solve the stated unfairness problems and to improve the network throughput of long-lived TCP by modifying the edge network Equationuipment in the following process. First, edge routers will be responsible for classifying flows and marking packets, whether they belong to long-lived or other flows. Second, once a flow is classified into a long-lived TCP flow, edge routers will be able to control the flow by providing a set of traffic regulation mechanisms. A mechanism for predicting the traffic level is firstly defined. Based on the predicted traffic level, we can decide to regulate the long-lived TCP by either reducing its data rate or splitting its connection to increasing the data speed. In order to highlight the advantage of the proposed transport mechanism, we develop two probability models for estimating the packet loss rate and several traversed hops, which respectively denote for the reliability and resource consumption and then compare with the conventional TCP. Furthermore, we perform an extensive set of ns-2 simulation experiments in a variety of scenarios in order to verify the analytical models. The analytical results and the supporting experimental outcomes show the following:

- When the error rate is high, ALL-TCP achieves higher reliability than the conventional system.
- In the proposed model, the lost packets are retransmitted from the edge router instead of retransmitting from the source node; hence the number of traversal hops will be reduced. In other words, the proposed model will be better than conventional in terms of resource consumption.

- The experiments show that our proposed model obtains higher throughput than the conventional. The reason is that the RTT of each segment is shorter than the end-to-end RTT, and the transmission process in each segment is independent of the others.

The rest of the paper is organized as follows. Section 2 outlines related works and positions our analysis concerning prior studies. In Section 3, we present the proposed framework in the viewpoint of architecture and operation mechanism. In Section 4, we develop the analytical models to highlight the advantages of our new transportation mechanism in terms of reliability and resource consumption and further conduct an extensive of ns-2 experiments to validate the accuracy of our analytical model. Finally, Section 5 concludes the paper with final remarks.

2 Related work

TCP has been recognized as the most important transport layer protocol for the Internet. It is distinguished by its reliable transmission, flow control, and congestion control [8, 26]. The literature on TCP protocol is considerable; however, it is out of the scope of this paper to present a comprehensive survey. Hence, the most related papers have been reviewed in this section in terms of long-lived TCP regarding network performance and unfairness issues. There is a real unfairness issue amongst long-lived TCP, UDP, and short-lived TCP in sharing network resources. As mentioned earlier in the introduction of this paper, UDP flows usually use more than their fair share of the bandwidth. However, there is no existing solution for solving the impact of UDP to other network flows because UDP is connectionless; it does not provide mechanisms to control the end-to-end data flow [24]. Hence, most of the researches focus on TCP traffics, connection-oriented protocol. The complexity of TCP unfairness is not only reflected by the competition in network resources between UDP and TCP traffics but also between the short and long-lived TCP. Short-lived TCP flows may reduce up to 10% in the throughput of the long-lived flows [17]. In order to resolve the unfairness problems, several types of research have been proposed. We can classify previous works into two aspects. First, the solutions that are given for general network traffic (i.e., TCP and UDP). Second, specific consideration will explore the unfairness of TCP traffics in detail regarding long-lived and short-lived TCP flows.

The general unfairness problems were early recognized and mentioned in [13, 28]. These papers focused on improving network fairness by either modifying TCP protocol or by employing non-tail-drop buffer management at routers. In particular, in [2], Morris proposes solutions to regulate the data traffics by providing per-flow information at all routers. In [28], Seddigh et al. propose a new packet dropping mechanism. These solutions show significant results; however, they require modifications on all routers which are unfeasible in deployments. The studies in [1, 7, 19, 33] aim to provide traffic metering and dropping mechanisms at the edge routers only with expected bandwidth or fairness guarantees. These researches are considered as an improvement to different services [22], which only provides several class priorities. Their dropping mechanism will base on the user profiles, which supplied vary of dropping level. Their result outcomes show the number of benefits to the network service provider. However, it is challenging to deploy in the overall network because the idea of the open internet is nowadays considered a significant stream in the network society. Many research aims to provide free internet service with high performance. In this paper, we aim to improve the network performance of long-lived TCP and reduce the unfairness in the network but not aiming at any specific users or any individual services. Most of the routers nowadays are green routers, which can modify their power plan to balancing their service rate according to network traffic, switching, and transmission capacity.

These are also several researches focused on analyzing the fairness between long-lived and short-lived TCP. The competition between long-lived and short-lived is early mentioned in [36]. The solution for solving this problem is proposed by I. Matta and L. Guo in [18]. In this paper, they provide a mechanism for isolating low and long TCP flows and then enhance response time and fairness of low flows. The significant results are given. However, the bandwidth (load) control is needed to perform at each core router. X. Wu and I. Nikolaidis proposed algorithms to classify schemes based on lifetime (short us. long-lived flows) and RTT attributes, as well as their combinations in order to provide a better TCP classification schemes. The double objective served by the classification schemes is to satisfy the need for reduced response time, which is the primary concern of short-lived flows, while at the same time ensuring fairness among long-lived flows [4]. In [12, 17], they try to improve the network fairness and to improve the network performance of short-lived TCP flows by suggesting to either using a considerable initial window value or sharing the network measurement information from previous records. These papers require modifying the TCP protocol at the end-host terminals and may lead to congestion collapse.

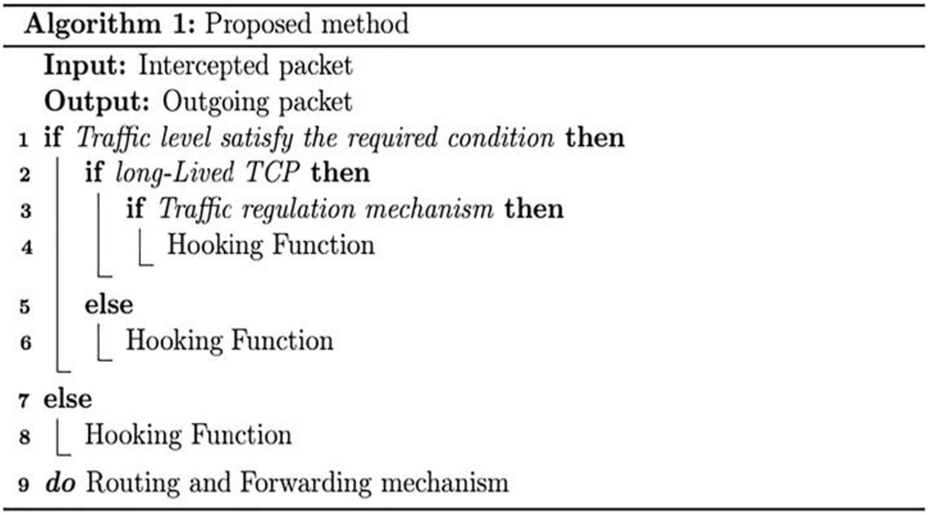
From different approaches, we provide a new framework to improve the network performance of long-lived TCP flow and to reduce its impaction to other network traffic flows. The following proposed mechanisms highlight our advantages by comparing them with previous studies. First, in order to avoid the unfairness problem, long-lived TCP is controlled separately by classifying the TCP traffics based on the amount of data passing by edge routers. Second, the FARIMA technology, a well-known predicting mechanism in an internet network, is used for determining the traffic level, which is a critical input factor for regulating network traffic. Third, based on the predicted traffic level, the edge device will decide whether it will activate the splitting mechanism for increasing the data transmission process or activate the slow-down mechanism for reducing the transmission speed of controlled, long-lived TCP flows when our edge network equipment is nearly overflowed. We believe that those upgraded mechanisms will improve the performance of long-lived TCP and significantly reduce its impaction to other data flows.

3 Methodology

In this section, the proposed methodology is represented to show the general procedure in solving the mentioned problem. Figure 1 presents an overview of our proposed framework. Several modifications are prerequisites in the network edge routers. Because in the current operation of almost the router operating system, all IP packets received from the Ethernet driver will be intercepted and processed by the routing mechanism for selecting the next forwarding hop. Hence, in the first step, a hooking function is provided in order to intercept all IP packets and forward them into our provided ALL-TCP layer. Hooking function refers to a variety of techniques employed to prevent calls to pre-existing employment and wind around them to change the function’s behaviour at runtime. Second, in the proposed provided layer, a traffic level prediction function is proposed for estimating the traffics state at the next time interval, which is an essential input factor for traffic regulation mechanisms. Next, the packet classification mechanism is provided for selecting long-lived TCP flows from network traffics. In this stage, we can separate TCP traffics from other traffics based on the protocol field in the IP header and then determine the long-lived flow by counting the number of packets of each flow passing the network edge router (i.e., more significant than 30 packets). Finally, the traffic regulation mechanism is designed for regulating the long-lived TCP flows based on the traffic level predicted in the second step [33]. Algorithm 1 shows the algorithm of the proposed method. The complexity of the proposed algorithm approximates O (n*t), where n denote the number of packet and t denote the complexity of the hooking function. The follows descriptions regarding the proposed method process:

- Hooking function: The hooking function is provided to intercept all IP packets and forward them into the suggested ALL-TCP layer.
- Traffic level prediction function: A traffic level prediction function is presented for estimating the traffics state at the next time interval, which is an essential input factor for traffic regulation mechanisms.
- Packet classification mechanism: The packet classification mechanism is provided for selecting long-lived TCP flows from network traffics. In this stage, we can separate TCP traffics from other traffics based on the protocol field in the IP header and then determine the long-lived flow by counting the number of packets of each flow passing the network edge router (i.e., more significant than 30 packets).
- Traffic regulation mechanism: The traffic regulation mechanism is designed for regulating the long-lived TCP flows based on the traffic level predicted in traffic level prediction function [33].

In order to implement these mechanisms, two solutions are used depending on the operating system. If the internetwork operating system (IOS) is developed in the edge router, the device is upgraded by adding the ALL-TCP layer in its upper protocol stacks. A hooking function is also needed to integrate into the IOS kernel in order to intercept all IP packets and forward them to the ALL-TCP layer for further processing. After processing in our provided layer, packets are redirected to the routing and forwarding module for transmitting to the next hop. In the second solution, the edge router implements a like-Unix operating system, so a modified IP and TCP layers are needed to comfort with our framework. The following sub-sections discuss the following two main mechanisms, traffic level prediction and traffic regulation in more detail.



Algorithm 1: Proposed method

```
Input: Intercepted packet
Output: Outgoing packet
if Traffic level satisfy the required condition then
    if long-Lived TCP then
        if Traffic regulation mechanism then
            Hooking Function
    else
        Hooking Function
else
    Hooking Function
do Routing and Forwarding mechanism
```

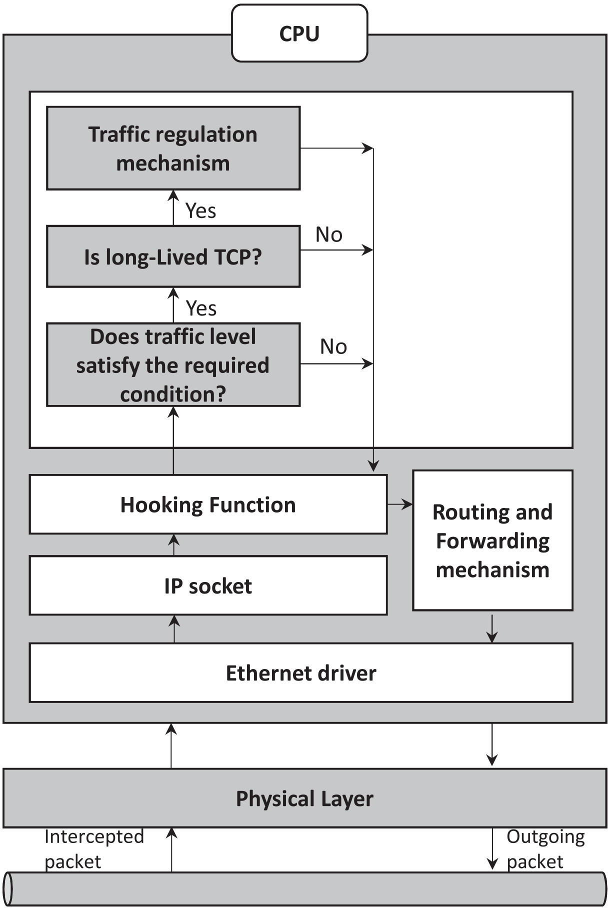


Fig. 1 The overview of proposed software architecture

3.1 Traffic level prediction

It is important to choose the right prediction model amongst several current prediction techniques (i.e., artificial neural network, autoregressive integrated moving average (ARIMA), fractional ARIMA (FARIMA), and wavelet-based predictors). Because each prediction technology requires matched input parameters based on the characters of analytical data. In the field of the internet network, Feng and Shu [36] compared amongst these prediction technologies by using mean square error and normalized mean square error as performance metrics [29]. The result shows that FARIMA obtains a better accuracy result than others. Also, it may be noted that internet traffic is mainly based on the normal distribution which can be presented by: $Y(t) = X(t) + \mu$, where μ is the mean rate, and $X(t)$ is a stochastic process with a continuous spectrum and zero mean. The FARIMA process is also in this form. Due to the two above reasons, FARIMA is chosen for predicting the network traffic level in our edge device.

FARIMA is a prediction model, which is fitted by time series data either to better understand the data or to predict future points in the series. The goal of FARIMA is to estimate network traffic $Y(t + \tau)$ from the measured traffic history $\{Y(r) \mid r\epsilon(-\infty, t]\}$. If given τ as the next control time interval. τ is a significant importance factor because it effects the trade-off between the processing time and the accuracy of prediction decision. For reducing the overhead of the network node processor and the update message in the routing protocol, $Y(t+\tau)$ must be predicted with an acceptable value of τ.

Assume that a confident prediction requires that the normalized interval τ -step prediction error ($\overline{\text{error}}(\tau) = \frac{\widehat{Y}(t+\tau)-Y(t+\tau)}{\widehat{Y}(t+\tau)}$) should not exceed a percentage ε (e.g. 20%) with a probability P_ε. The maximum prediction interval (MPI) is defined as the follows Equation:

$$\text{MPI} = \max\left\{\tau|P_{\overline{\text{error}}}(\tau,\varepsilon)\le P_\varepsilon\right. \tag{1}$$

where $P_{\overline{\text{error}}} = \Pr(\overline{\text{error}}(\tau) > \varepsilon)$. $P_{\overline{\text{error}}}(\tau,\varepsilon)$ is Equation to $P(Z>0)$ where Z is a random variable of the Gaussian distribution with a probability density function $N\left(-\varepsilon\mu, \sigma^2_{\varepsilon,\tau}\right)$.

Based on the Gaussian assumptions identified in this paper, the traffic could be presented by: $Y(t) = X(t) + \mu$, where μ is the mean rate, and $X(t)$ is a stochastic process with a continuous spectrum and zero mean. By applying the wold's decomposition theorem, Gaussian process $Y(t)$ can be represented by one-sided moving average as the follows Equation:

$$Y(t) = \sum_{u=0}^{+\propto} h_u n(t-u) + \mu \tag{2}$$

where n(t) and h_u denote Gaussian white noise and the possibly infinite vector of moving average weights (coefficients or parameters), respectively. The optimal $\tau-$step predictor of the Gaussian process $X(t+\tau)$ is $\widehat{X}(t+\tau) = E[X(t+\tau)|X(s), s\le t]$. Applying Kolmogorov's approach, $\widehat{X}(t+\tau)$ can expressed as the follows Equationuation:

$$\widehat{X}(t+\tau) = \sum_{u=0}^{+\infty} h_{u+\tau} n(t-u). \tag{3}$$

Comparing to Eqs. (2) and (3), the unpredictable part in $X(t+\tau)$ is $\sum\limits_{u=0}^{\tau-1} h_u n(t+\tau-u)$. So $\tau-$step predictor variance $\widehat{\sigma^2_\tau}$ can be expressed in Eq. (4).

$$\widehat{\sigma^2_\tau} = \sum_{u=0}^{\tau-1} h_u^2 = \sigma^2 - \sum_{u=0}^{+\infty} h_u^2 \tag{4}$$

where σ^2 is the variance of X(t). $\sigma^2_{\varepsilon,\tau}$can be calculated by using Eq. (5).

$$\sigma^2_{\varepsilon,\tau} = \sum_{u=0}^{\tau-1} h_u^2 + \varepsilon^2 \sum_{u=\tau}^{+\infty} h_u^2 = \left(1-\varepsilon^2\right)\widehat{\sigma^2_\tau} + \varepsilon^2\sigma^2. \tag{5}$$

From Eqs. (3) and (4), $P_{\overline{\text{error}}}(\tau,\varepsilon)\le P_\varepsilon$ in Eq. (1) is equivalent to $\sigma^2_{\varepsilon,\tau}\le\frac{\varepsilon^2\mu^2}{\Phi^2(1-P_\varepsilon)}$ where $\Phi(x)$ is the inverse CDF of N(0,1). From Eqs. (1) and (5), MPI can be given as:

$$\text{MPI} = \max\left\{\tau|\ \frac{\widehat{\sigma}_r^2}{\sigma^2}\le\left(\frac{1}{C^2\Phi^2(1-P_\varepsilon)}-1\right)\frac{\varepsilon^2}{1-\varepsilon^2}\right\} \tag{6}$$

where C=σ/μ is the variance coefficient of Y(t).

3.2 Traffic regulation

Based on the traffic level, $Y(t+\tau)$ which is predicted at the next interval of the control time of τ, ALL-TCP agent will decide to either accept or refuse the new long-lived TCP flows [21].

In the first case, when the predicted traffic level is lower than the minimum threshold α. The new long-lived TCP flows will be accepted to enter the proposed traffic regulation module, where the long-lived TCP flow will be split into two sub-connections that work independently. From the functionality point of view, each edge network router operates as a destination host by sending acknowledgment messages for every received packet from the preceding node. Also, it contains functions that operate similarly to those in the conventional source host, in which it follows existing algorithms for sending data packets in response to incoming ACKs, multiple duplicate acknowledgments, or retransmit timeouts.

In the second case, if the predicted traffic level excesses the maximum threshold β, ALL-TCP will reduce the data rate of incoming traffic to avoid congestion state instead of accepting new connections. Random early detection protocol (RED) [11] is used to reduce the data rate of incoming traffic with a few modifications needed. In which, the algorithms for calculating the average queue length is replaced by FARIMA. The behavior of RED with UDP and short-lived TCP flows are remained the same as it does in original version. Howeve, we control the long-lived TCP flows is changed where the long-lived TCP flow that has the highest amount of buffered data in amongst existed long-lived TCP flows in edge router is selected for reducing data rate. This behavior will guarantee the fairness between long-lived TCP flows in resource sharing. For selecting the flow that has highest amount the stored data in buffer, we need to measure the incoming and outgoing packets rate of each long-lived TCP at the next time interval, $t+\tau$ separately. If the predicted incoming and outgoing packets of the preceding and the next segment at the next time interval are $Y_i(t+\tau)$, $Y_o(t+\tau)$ and the current stored data in the buffer is δ, then expected buffering data for the long-lived TCP flow can be measured as the follows Equation:

$$B(t+\tau) = Y_i(t+\tau) - \Big(\delta + Y_o(t+\tau) \tag{7}$$

The predicted incoming and outgoing packets of preceding and next segments are similarly determined based on the congestion window size at the next time interval of τ. For example, when the current congestion window size of the preceding segment is $cwnd_i$ and its incoming packets arrive the edge router at the time before the predicted time interval, then $Y_i(t+\tau) = cwnd_{i+1}$, otherwise $Y_i(t+\tau) = 0$. If W_{max} presents for the maximum window size and an Ack message are sent to the sender after receiving a number of η data frames, then the value of the next congestion window size in the case of congestion free can be obtained by the follow Equation:

$$cwnd_{i+1} = \begin{cases} W_{max,} & \text{if } cwnd_{i+1} \geq W_{max} \\ cwnd_i + cwnd_i/\eta, & \text{Otherwie} \end{cases} \tag{8}$$

In order to reduce the jitter problem at the end of the host terminal and to keep the allocated memory for each flow in the edge router at a minimal value, the data transmission rate between sub-segments of each long-lived TCP flow should be regulated based on the following ratio:

$$\gamma = \frac{\delta + cwnd_{i+t}}{cwnd_{o+t}} \tag{9}$$

This ratio should be regulated at a value that higher or at least equal to 1 and does not exceed a defined maximum threshold (i.e., $\gamma \leq 2$). When γ is out of the allowed range, the mechanism need to adjust the bias by delay sending packets or acknowledgment message of the sub-segment which has higher throughput. This mechanism is similar to TCP delayed acknowledgment that is proposed in [13].

4 Analysis and simulation

In the proposed traffic regulation module, each long-lived TCP is running based on segment-by-segment transportation (SBST). Hence, in Section A, we explore the advantages of SBST by comparing it with conventional end-to-end transportation (ETET) regarding reliability and resource consumption. In section B, simulated experiments by using the ns-2 tool are conducted to verify the analytical model and to investigate the network throughput of long-lived TCP further.

It's hard to tune the SBST and ETET, because of the SBST is related to the data transformation to segments at the source node and reconstructed at the receiver. On the other hand, the ETET is related to different feature such as the flow controls rate and error detection, which make the tuning is hard to implement.

4.1 Fundamental of SBST and ETET

In this section, the network reliability and resource consumption of each transportation mechanism is measured by modeling the behavior of a packet. Which is when sent from the sender to the receiver. The reliability is presented by overall packet loss probability, while resource consumption can be measured based on the number of traversed routers that a packet needs to pass by in order to reach the receiver. The following assumptions are used in the proposed model:

- p_i: Packet loss probability at router i.
- p_{si}: Probability of a packet passing by router number i successfully.
- p_e: End-to-end packet loss probability.
- p_s: Probability of successfully sending a packet from the source to the destination.
- n: The total number of routers in the network.
- n_1: The number of routers of the first segment.
- n_2: The number of routers of the second segment.
- R: Maximum number of retransmissions.

4.1.1 Packet loss probability in ETET

The probability of sending a packet successfully from the sender to the receiver is defined as:

$$p_s = \prod\nolimits_{i=1}^{n} (1-p_i). \tag{10}$$

Hence, the probability of sending a packet unsuccessfully after R retransmission times is donated as follows:

$$p_e = [1-p_s]^R = \left[1-\prod\nolimits_{i=1}^{n} (1-p_i)\right]^R. \tag{11}$$

Assume that a packet is dropped at a random position k when transmitting to the receiver, then the probability of this event can be described as:

$$p_{e_k} = p_k \prod\nolimits_{i=1}^{k-1} (1-p_i). \tag{12}$$

In one transmission time, the number of traversed routers of event that a packet may not reach the receiver is defined as the follows Equation:

$$h_f = \sum_{k=1}^{n-1} \frac{k p_{e_k}}{1-\prod_{i=1}^{n}(1-p_i)} = \sum_{k=1}^{n-1}\left[\frac{k\ p_k\prod_{i=1}^{k-1}(1-p_i)}{1-\prod_{i=1}^{n}(1-p_i)}\right]. \tag{13}$$

Hence, the expected of total traversed routers of event that a packet cannot reach the receiver after a maximum number of retransmissions, R, is given as follows Equation:

$$h_R = Rh_f = R\sum_{k=1}^{n-1}\left[\frac{k\ p_k\prod_{i=1}^{k-1}(1-p_i)}{1-\prod_{i=1}^{n}(1-p_i)}\right]. \tag{14}$$

The expected of traversed routers of event that a packet reaches a receiver after a number of r retransmissions time (r < R) are obtained by the follows Equation:

$$h_r = (r-1)h_f + n. \tag{15}$$

The probability of event that a packet may not reach the receiver after a maximum number of retransmissions, R, is presented by the follows Equation:

$$p_R = \left[1-\prod_{i=1}^{n}(1-p_i)\right]^R. \tag{16}$$

The probability of event that a packet reaches to the receiver after a number of retransmissions r (r < R) is given as the follows Equation:

$$p_r = \prod_{j=1}^{r}\left\{\left[1-\prod_{i=1}^{n}(1-p_i)\right]\right\}^{j-1}\prod_{i=1}^{n}(1-p_i). \tag{17}$$

From Eqs. (14), (15), (16), and (17), the total expected of traversed routers when a packet is transmitted through an end-to-end connection is defined as the follows Equation:

$$h_t = \sum_{r=1}^{R-1}(p_r h_r) + p_R h_R. \tag{18}$$

4.1.2 Packet loss probability in ETET

Packet loss probability in SBST is measured by aggregating the packet loss probability in the first and the second segments. The probability of event that a packet is transmitted successfully through the first segment is given as the follows Equation:

$$p_{s_{i1}} = \prod_{i=1}^{n_1}(1-p_i). \tag{19}$$

The probability of event in which a packet cannot reach the receiver after a maximum number of retransmissions, R, can then be:

$$p_{e_1} = \left[1-p_{s_{i1}}\right]^R = \left[1-\prod_{i=1}^{n_1}(1-p_i)\right]^R. \tag{20}$$

Hence, the probability of an event in which a packet is transmitted successfully by passing through the first and second segments is defined sequentially as:

$$p_{s_1} = 1-p_{e_1} = 1-\left[1-\prod_{i=1}^{n_1}(1-p_i)\right]^R \tag{21}$$

$$p_{s_2} = 1-p_{e_2} = 1-\left[1-\prod_{i=1}^{n_2}(1-p_i)\right]^R \tag{22}$$

From Eq. (21) and (22), the probability of an event in which a packet is transmitted successfully from the sender to the receiver is described as follows:

$$p_s = p_{s_1}\ p_{s_2} \tag{23}$$

Therefore, the probability of an event in which a packet cannot reach the receiver after a maximum number of retransmissions, R can be obtained as follows:

$$p_e = 1-p_s = 1-\left\{1-\left[-\prod_{i=1}^{n_1}(1\ \ p_i)\right]^R\right\}\left\{1-\left[1\prod_{i=1}^{n-n_1}(1-p_i)\right]^R\right\} \tag{24}$$

4.1.3 Resource consumption in SBST

Similar to Eq. (18), the expected total traversed routers that a packet needs to pass through the first segment (h_{t_1}) can be calculated as:

$$h_{t_1} = \sum_{r=1}^{R-1}\left(p_{r_1}h_{r_1}\right) + p_{R_1}h_{R_1}, \tag{25}$$

where p_{r_1}, h_{r_1} represent for the packet loss probability and number of traversed routers of event that a packet is transmitted successfully though the first segment, respectively. While, p_{R_1}, h_{R_1}denote for the packet loss probability and number of traversed routers of event that a packet cannot reach the receiver after a maximum number of retransmission R. The probability of an event that a packet is transmitted successfully though the second segment after a number of retransmissions; k can be defined as the follows Equation:

$$p_{r_2} = \left\{1-\left[1-\prod_{i=1}^{n_1}(1-p_i)\right]^R\right\}\left[1-\prod_{i=1}^{n_2}(1-p_i)\right]^{(k-1)}\left[\prod_{i=1}^{n_2}(1-p_i)\right]. \tag{26}$$

The probability of an event that a packet cannot reach the destination after a maximum number of retransmissions, R, is given as follows Equation:

$$p_{R_2} = p_{r_1}\left[1-\prod_{i=1}^{n_2}(1-p_i)\right]^R. \tag{27}$$

Similar to the analyses in (14) and (15), the number of traversed routers of events that a packet transmits through the second segment in both cases success and failure after a maximum number of retransmissions, R, can be represented as $h_{r_1}$and h_{r_2}, respectively. From Eqs. (26) and (27), the overall traversed routers of event that a packet is transmitted through the second segment can be obtained as follows:

$$h_{t_2} = \sum_{r_2=1}^{R-1}\left(p_{r_2}h_{r_2}\right) + p_{R_2}h_{R_2}. \tag{28}$$

From Eqs. (25) and (28), the overall number of traversed routers (h_t) through which a packet passes in SBST is identified as:

$$h_t = h_{t_1} + h_{t_2} = \sum_{r=1}^{R-1}\left(p_{r_1}h_{r_1}\right) + p_{R_1}h_{R_1} + \sum_{r_2=1}^{R-1}\left(p_{r_2}h_{r_2}\right) + p_{R_2}h_{R_2}. \tag{29}$$

4.1.4 Analytical results

The comparison results can be obtained based on the established Equations, which is given above by using MATLAB software package. The configure ration of the given parameters are

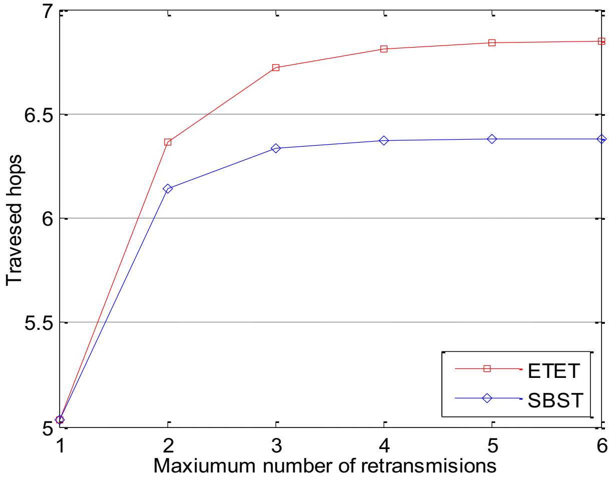


Fig. 2 Network reliability comparison

set to $n = 8$, pi = 0.05, and R = 10 according to [3]. The packet loss probability of SBST and ETET are obtained by using Eqs. (11) and (24), respectively. The results in Fig. 2 show that the SBS is more reliable than the conventional ETET. The number of traversed routers is determined by using Eqs. (18) and (29). The results shown in Fig. 3 present that, in SBST, a packet needs to pass by a fewer number of routers to reach the receiver than in ETET. In other words, SBST consumes less network resource than the conventional ETET.

4.2 Simulation experiments

We simulate ALL-TCP by modifying the implementation of the current Tahoe TCP in ns-2, which generally can be summarized as follows.

First, in order to provide mechanisms for sending packets, we redesign the packet sending module by driving from the TcpAgent class. Second, for acknowledging the received packet, an acknowledgment module is provided by driving from the TcpSink class. Each long-lived

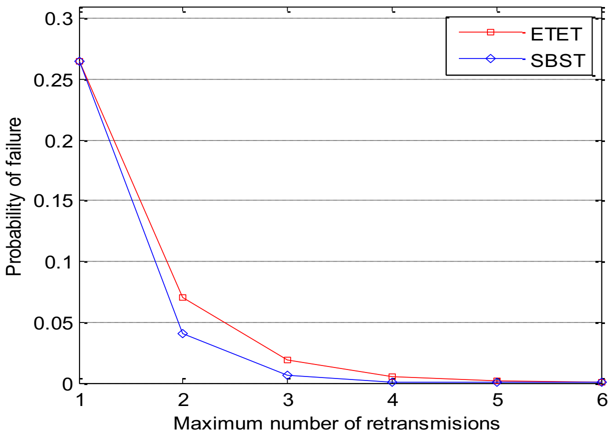


Fig. 3 Resource consumption comparison

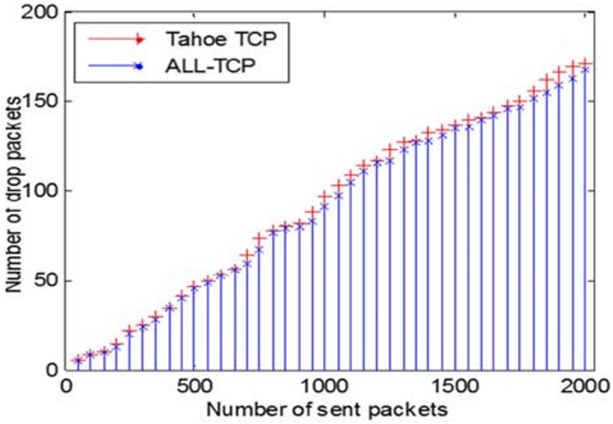


Fig. 4 Experiment configuration

TCP flow is maintained and controlled by each flow separately, which includes both sending and acknowledgment modules. Third, because ns-2 currently provides a single point of packet entrance called entry (Which is a connector object). Each packet that enters the Node entry is directly forwarded to the address classifier module (i.e., *Classifier instvar*). If the Node is not the final destination, the address classifier will forward the packet to the link specified in the routing table. Therefore, we need to provide a hooking function.

In order to intercept packets and forward them to our provided ALL-TCP agent for further processing, our framework lie on the change in transport mechanism without modifying the original TCP protocol implemented in the end-host terminals. Hence, in a real implementation, the edge router is responsible for adapting with a different version of TCP, such as Reno, New-Reno, or SACK.

As we proposed in the Tcl simulated code, a network with four routers that connects two host terminals is created. Each link is configured to 4 Mb of bandwidth capacity, and 50 ms link-delay. We use link error instead of creating congestion in routers in order to create packet losses. The link error probability is similar to packet loss probability, p_i, which is used in our analytical model. The ALL-TCP agent is attached to router number 2

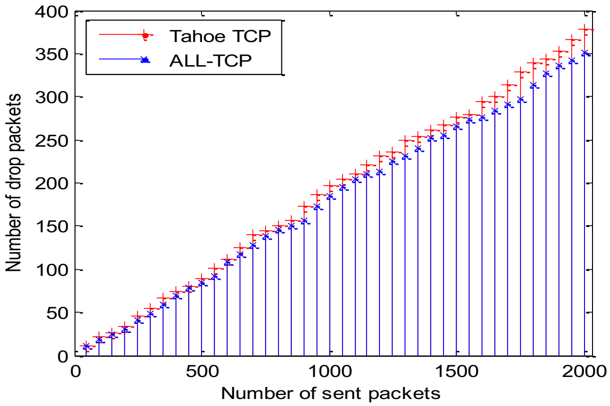


Fig. 5 Drop packets with link error 0.5

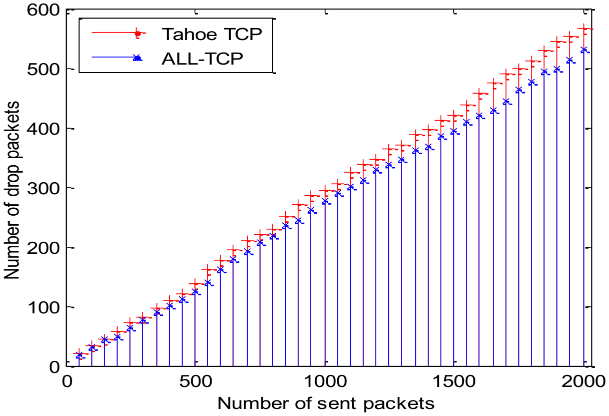


Fig. 6 Drop packets with link error 0.75

when we simulate for the proposed framework. The overview of our experiment configuration is shown in Fig. 4.

The first experiment aims to confirm the analytical model in section Packet Loss Probability in ETET, where the reliability of transmitting a packet via ALL-TCP is proved better than the conventional TCP. In this experiment, the amounts of dropped packets are measured by sending 2.000 packets via the network with different link error. The results from Figs. 5, 6, 7, and 8 show the amounts of dropped packets for link error: 0.025, 0.05, 0.075, and 0.1, respectively. The results show that the packet drop probability of the proposed framework is always smaller than the conventional Tahoe TCP. This conclusion confirms to our analytical model in section Packet Loss Probability in ETET.

In the ALL-TCP framework, each long-lived TCP connection is split into sub-connections that work simultaneously in the case that network traffic is lower than the minimum threshold value. It is because the round trip time of each sub-connection is shorter than the end-to-end RTT. Hence, ALL-TCP is expected to gain higher network throughput than the conventional TCP. In the second experiment, we try to measure the network throughput at the TCP Sink based on the number of received bytes in a fixed 0.8 s time scale. The results in Fig. 9 showed ALL-TCP achieves higher network throughput than the conventional Tahoe TCP and finishes

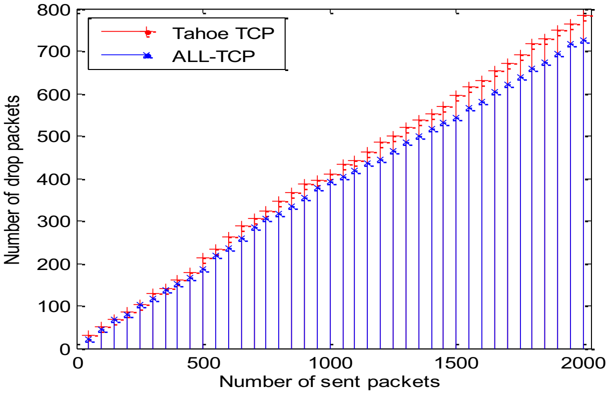


Fig. 7 Drop packets with link error 0.1

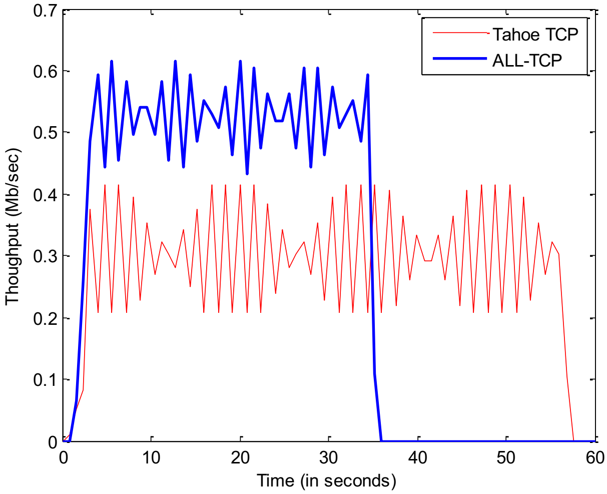


Fig. 8 Throughput comparison in normal state with one ALL-TCP agent in deployment

the transmission of 2000 packets in a shorter time. The results (as shown in Fig. 10) are even better than if there are two ALL-TCP agents are deployed in the network.

In order to highlight the adaptive ability of the proposed framework, we measure the recovery time of congestion window size from the time of multiplicative decreasing until it recovers to the highest value. The results in Fig. 11 shows ALL-TCP quickly retrieve the highest congestion window size each time it incurs congestion. This advantage helps ALL-TCP not only get higher network throughput but also it is better in adaption to congestion. As well as, this Figure shows that ALL-TCP always get higher network throughput than Tahoe TCP in the case of network configuration with link error set to 0.1.

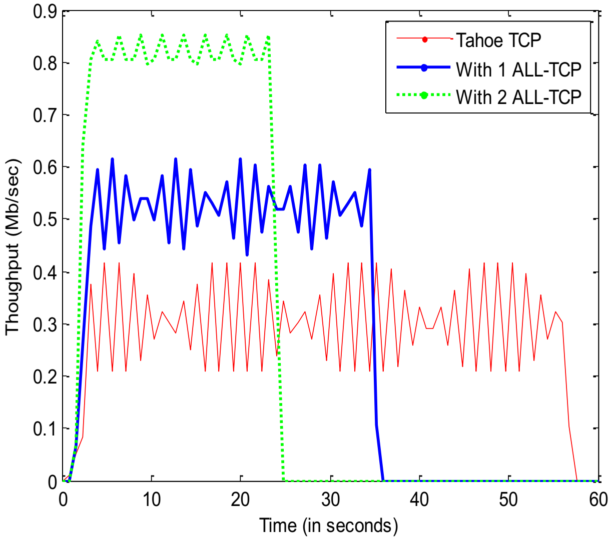


Fig. 9 Throughput comparision in normal state with two ALL-TCP agents in deployment

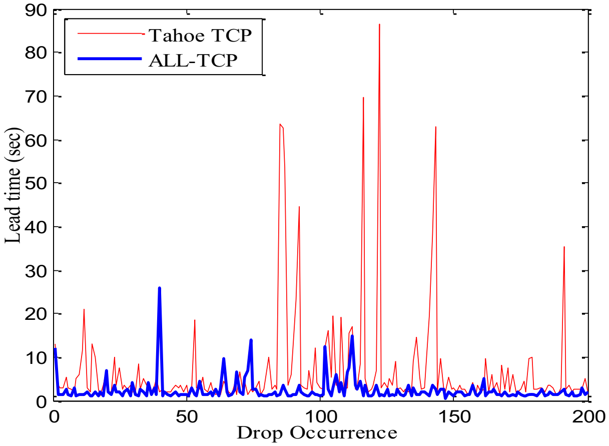


Fig. 10 Lead time to recover congestion window size after congestion

5 Conclusion

In this paper, a novel protocol design is proposed to improve the network performance of long-lived TCP (ALL-TCP). ALL-TCP firstly separates long-lived TCP from other data flows. Secondly, the traffic regulation mechanism is proposed in which a number set of functions is provided to regular, and control long-lived TCP connections in an edge router. The results of the analytical models and an extensive set of ns-2 simulation showed the follows: 1) ALL-TCP is more reliable and consumes less energy than the conventional TCP in the case of high error rate in the network. 2) It always presents higher network throughput in both cases of congestion state or normal operation due to its shorter RTT. 3) Our modification is on the edge network equipment only, hence these are no effect to the end-host terminals. It could be deployed gradually in current TCP/IP networks with high scalability. 4) The network fairness is obtained because the long-lived TCP is separated from other data flows. For future work, the

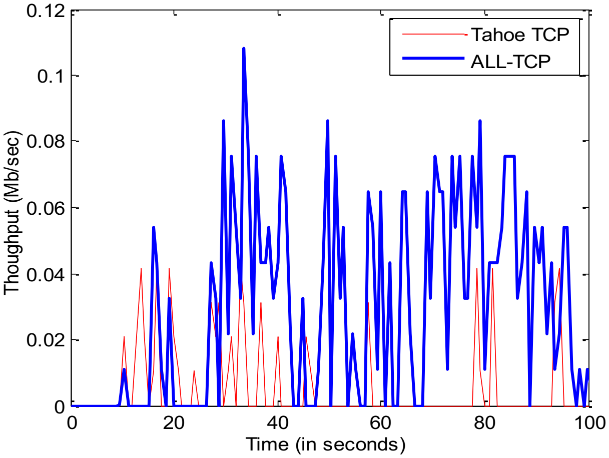


Fig. 11 Throughput comparison with error link 0.1

proposed mechanism can be modified to improve other TCP versions like BIC TCP or compound TCP, in hope to improve the enhance performance and fairness of long-lived TCP flows.

References

1. Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer, Berlin, pp 1–165
2. R. Braden, “REquationuirements for Internet Hosts – Communication Layers, ” RFC 1122, Oct.19.
3. Carlucci G, De Cicco L, Holmer S, Mascolo S (2017) Congestion control for web real-time communication. IEEE/ACM Trans Networking 25(5):2629–2642
4. Carlucci G, De Cicco L, Mascolo S (2018) Controlling queuing delays for real-time communication: the interplay of E2E and AQM algorithms. ACM SIGCOMM Computer Communication Review 46(3):1–7
5. Carlucci G, De Cicco L, Mascolo S (2018) Controlling queuing delays for real-time communication: the interplay of E2E and AQM algorithms. ACM SIGCOMM Computer Communication Review 46(3):1
6. Cheng, J., & Grinnemo, K. J. (2017). Telco distributed DC with transport protocol enhancement for 5G Mobile networks: a survey.
7. Clark DD, Fang W (1998) Explicit allocation of best-effort packet delivery service. IEEE/ACM Trans Networking 6(4):362–373
8. Douga Y, Bourenane M, Mellouk A, Hadjadj-Aoul Y (2016) TCP based-user control for adaptive video streaming. Multimed Tools Appl 75(18):11347–11366
9. Ebrahimi-Taghizadeh S, Helmy A, Gupta S (2005) TCP vs. TCP: a systematic study of adverse impact of short-lived TCP flows on longlived TCP flows. In: Proceedings of INFOCOM, vol 2, Miami, pp 926–937
10. S. Floyd and V Jacobson, “Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, ” pp. 397–413, 1993.
11. Floyd S, Jacobson V (1993) Random early detection gateways for congestion avoidance. IEEE/ACM Trans Networking 1(4):397–413
12. Heidemann J (1997) Performance interactions between P-HTTP and TCP implementations. ACM Computer Communication Review 27(2):65–73
13. P Hurley , J. L. Boudec, and P Thiran, “A note on the fair-ness of additive increase and multiplicative decrease, ” in 16th international T eletraffic congress (ITC-16), Edinburgh, Scotland, 1999.
14. Kakhki AM, Jero S, Choffnes D, Nita-Rotaru C, Mislove A (2017) Taking a long look at QUIC: an approach for rigorous evaluation of rapidly evolving transport protocols. In: proceedings of the 2017 internet measurement conference, pp 290–303 ACM
15. Kennedy, J., Armitage, G., & Thomas, J. (2017). Household bandwidth and the'need for speed': evaluating the impact of active queue management for home internet traffic. Journal of Telecommunications and the Digital Economy, 5(2), 113.
16. Kua J, Nguyen SH, Armitage G, Branch P (2017) Using active queue management to assist IoT application flows in home broadband networks. IEEE Internet Things J 4(5):1399–1407
17. Liu J, Chi Y, Liu Z, He S (2019) Ensemble multi-objective evolutionary algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps. CAAI Transactions on Intelligence Technology 4(1):24–36
18. Matta and L. Guo, “Differentiated Predictive Fair Service for TCP Flows,” inProc. ICNP’2000, Osaka, Japan, Nov. 2000.
19. Mehdi H, Pooranian Z, Vinueza Naranjo PG (2019) Cloud traffic prediction based on fuzzy ARIMA model with low dependence on historical data Transactions on Emerging Telecommunications Technologies:e3731
20. G. L. Monoco, F. Azeem, and S. Kalyanaraman, “TCP-friendly marking for scalable best-effort services on the internet, ” ACM computer communication review, 2001.
21. Padmanabhan and R. Katz, “TCP Fast Start: a Technique for Speeding Up Web Transfers,” inProc. IEEE Globecom’98 Internet Mini-Conference, Nov. 1998.
22. Pukkala T (2019) Optimized cellular automaton for stand delineation. J For Res 30(1):107–119
23. Real Ehrlich C, Blankenbach J (2019) Indoor localization for pedestrians with real-time capability using multi-sensor smartphones. Geo-spatial Information Science 22(2):73–88
24. Rejaie R, Handley M, Estrin D (1999) RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the internet. In INFOCOM'99. Eighteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE 3:1337–1345 IEEE

25. Roberts, J., Skandalakis, J., Foard, R., & Choi, J. (2016). A comparison of SDN based TCP congestion control with TCP Reno and CUBIC. Technical Report.
26. Saldana J (2016) On the effectiveness of an optimization method for the traffic of TCP-based multiplayer online games. Multimed Tools Appl 75(24):17333–17374
27. N. Seddigh, B. Nandy, and P . Pieda, "Study of TCP and UDP Interaction for the AF PHB, " Internet Draft, draft-nsbnpp-diffserv-tcpudpaf-01.pdf, 1999.
28. N. Seddigh, B. Nandy, and P Pieda, "Study of TCP and UDP Interaction for the AF PHB, " Internet Draft, draft-nsbnpp-diffserv-tcpudpaf-01.pdf, 1999.
29. S. Shenker, "A theoretical analysis of feedback flow control, " in ACM/SIGCOMM'90, 1990.
30. Soleimani MHM, Mansoorizadeh M, Nassiri M (2018) Real-time identification of three Tor pluggable transports using machine learning techniques. J Supercomput 74(10):4910–4927
31. Sunny A, Panchal S, Vidhani N, Krishnasamy S, Anand SVR, Hegde M, … Kumar A (2017) A generic controller for managing TCP transfers in IEEE 802.11 infrastructure WLANs. J Netw Comput Appl 93:13–26
32. Tsai CS, Wang YC, Wang HK (2015) A. Transmission Control Protocol with High Throughput of Using Low Earth-Orbit Satellite to Collect Data from the Floats on Sea Surface 26(2):78–96
33. S. Yilmaz, I. Marta, "On class-based isolation of UDP, short-lived and long-lived TCP flows", 9th Int. Symp. MASCOTs 2001, pp 415–422.
34. Zaidi, S. M. R., Hassink, B. J., & Grover, L. (2017). U.S. Patent No. 9,729,454. Washington, DC: U.S. Patent and Trademark Office.
35. Y. Zhang, L. Qiu, and S. Keshav, "Speeding Up Short Data Transfers: Theory, Architecture Support, and Simulation Results," in Proc. NOSSDAV 2000, Chapel Hill, NC, Jun. 2000.
36. Zhang Y, Hossain MS, Ghoneim A, Guizani M (2019) COCME: content-oriented caching on the mobile edge for wireless communications. IEEE Wirel Commun 26(3):26–31

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Mohammad Al Shinwan[1] · Laith Abualigah[1] · Nguyen Dinh Le[2] · Chulsoo Kim[2] · Ahmad M. Khasawneh[1]

Laith Abualigah
Aligah.2020@gmail.com

Nguyen Dinh Le
nguy.le@inje.ac.kr

Chulsoo Kim
charles@inje.ac.kr

Ahmad M. Khasawneh
a.khasawneh@aau.edu.jo

[1] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

[2] Department of Computer Engineering, Inje University Gimhae, Seoul, Republic of Korea