Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
MED2016-2021

Design and Evaluation of Future 5G Mobile Network Transport Protocol

Ivan Petrov
Makedonski Telekom AD
FYROM

Toni Janevski
Professor
Ss. Cyril and Methodius University
FYROM

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. This paper has been peer reviewed by at least two academic members of ATINER.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Petrov, I. and Janevski, T. (2016). **"Design and Evaluation of Future 5G Mobile Network Transport Protocol",** Athens: ATINER'S Conference Paper Series, No: **MED2016-2021.**

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209 Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm
Printed in Athens, Greece by the Athens Institute for Education and Research.
ISSN: 2241-2891
25/10/2016

Design and Evaluation of Future 5G Mobile Network Transport Protocol

Ivan Petrov

Toni Janevski

Abstract

The past two decades TCP development was elevated at higher level mainly because of its poor performances in high speed networks. The Congestion Avoidance phase was studied and identified as the main reason for this behavior. A variety of TCP proposals were developed with the aim to improve protocol utilization in a high speed environment, most of them made changes of the initial TCP's congestion control mechanism, with or without network layer relations. In this paper we present the novel transport protocol, designed to assure efficient data usage in future 5G mobile networks. This unique transport protocol is designed to prevent congestion collapses of the backhaul network traffic and to assure ultra-high data rates of up to 400Gbps.

Keywords: 5G mobile networks; Congestion control; TCP; Throughput.

Introduction

From the very beginning until today TCP is evolving and the document roadmap describing its development is frequently updated [1]. One of the key triggers in the development process was its poor performance in high speed networks.

The congestion avoidance phase was studied and identified as the generator of this unpleasant behavior. Numbers of TCP versions were created with primary aim to improve the protocol utilization in a high speed environment and most of them were addressing and redesigning the congestion control mechanism. It is important to notice that during the design phase a variety of constraints must be carefully observed especially when creating a protocol for usage in future 5G mobile networks [2-5].

Today we distinguish three main groups of high speed protocols: loss based [6-12], delay based [13-16] and loss based with bandwidth estimation [17-22]. High speed TCP proposals use approach of more aggressive network probing with the aim to create a congestion control mechanism that will be aggressive enough in case of underutilized network and will remain gentle in case of utilized resources.

There are two main reasons for poor performance of standard TCP protocols in high speed networks:

- Cwnd linear increase by one packet per round trip time in congestion avoidance phase is too slow and the multiplicative decrease in case of loss is too drastic.
- Maintaining large average congestion windows at flow level require extremely small equilibrium loss probability and a stable flow dynamic design.

In the next section, we discuss future 5G network design and we express the need for development of TCP Ohrid. TCP Ohrid is defined in the “TCP Ohrid Protocol” section. In the “Simulation Scenario” section we present the simulation environment and discuss the results. The last section concludes the paper.

Future Networks

Next Generation Networks (NGN) and Future Networks [2] are all-IP networks, meaning that all data, control and signaling will be carried through IP-based communication and based on Internet technologies form network protocol layer up to the application layer, with heterogeneous access technologies (fixed and mobile) on the lower layers of the protocol stack. TCP Ohrid meets 5G characteristics which are set prior to the standardization, expected to be finalized around 2020.

Most researchers are focused on the development of 5G network that will provide improved future communications. Fundamental design, requests and drivers that are defined and argued in [3]. 5G Network design scenarios and evaluations are conducted in [4].

Ideas for mobile cloud computing in order to offload most energy consuming tasks to nearby fixed servers and to develop mobile cloud computing are reality and are subject of future development [5].

As we know, small cells and their ultra-dense deployment could become a main feature of 5G networks indicating that the number of small cells will increase in unit area so the corresponding backhaul gateway traffic will increase exponentially if the conventional centralized control model is adopted as 5G backhaul network architecture. If this network design is chosen during the selection process we have to notice that massive backhaul traffic will create congestion which might result to the collapse of the backhaul network. Therefore, there is a need for development of ultra-fast protocol similar to the TCP Ohrid, able to assure data transfer rates up to 400 Gbps or lower rates in case of heavy mobility of the terminals, making the protocol more efficient than the traditional one. This protocol is able to reduce and dynamically manage the backhaul congestion.

Hybrid backhaul networks using wireless and fiber access technologies can be used to increase the energy efficiency of the backhaul network.

We identify several requests that should be satisfied by future networks. 5G networks will have to employ energy harvesting mechanisms at core base station network. For example, “harvest-then-transmit” mechanism or similar can be used in order to save energy [23].

Optimization at terminal application level with offload of most energy consuming tasks to nearby fixed servers presented as mobile cloud computing is additional value that can be added and employed at the application layer in order to save terminal battery. Both centralized and decentralized approaches can be used in order to provide this network function.

5G Networks must represent an all IP mobile platform that will provide infrastructure as a service, platform as a service and software as a service. (Unified access of the platform will be provided regardless the medium access control, e.g., WiFi, 2G, 3G, LTE, Copper, Fiber, RF, etc.). The network must offer higher data transfer rates in static and mobile (nomadic) mode of operation.

TCP Ohrid Protocol

The development of TCP Ohrid is inspired by the HSTCP protocol. We have noticed that HSTCP response function can be modified but the main request to remain convex should be preserved. It is possible to define different increase and decrease parameter functions by the usage of different interpolation mechanisms. Equation (1) describes all loss based TCP protocols and in combination with (2) it defines HSTCP.

$$\begin{aligned} w &\leftarrow w + \frac{\alpha}{w} && \text{CA phase, when ACK received} \\ w &\leftarrow w - \beta w && \text{in case of drop} \\ w &\leftarrow w + \gamma && \text{Slow Start} \end{aligned} \tag{1}$$

$$\alpha(w) = \frac{w^2 2\beta(w) p(w)}{2 - \beta(w)} \tag{2}$$

α is window increase parameter with default value of 1 Maximal Segment Size, β is decrease parameter with default value of 0.5 MSS, w denotes the congestion window size, p is packet loss probability and γ is slow start parameter with default value of 1 MSS. When default values of α, β and γ are used equation (1) describes the standard TCP (Reno, Newreno) protocol versions.

TCP Ohrid is defined with several response functions where linear interpolation is used for six known points. Usage of several response functions for different speed improves protocol friendliness at lower rates. The algorithm combines response functions in order to provide improved protocol friendliness at low and high rates. We know that the fiber optic error rate is 1 error in 10^{10} bits meaning that the probability of bit error in fiber is 10^{-10}. In our scenario we use packets with a size of 1500 bytes that limits the packet loss probability at $P=12000x10^{-10}=1.2x10^{-6}$. TCP Ohrid preserves usage of switch point defined with w=38 packets, p=0,0015 and β parameter set at 0.5 in order to represent fair protocol that will be friendly with competing TCP. If we want to create a protocol that will achieve Gbps throughput we have to modify the basic definition (1) in order to get faster window increase. This implies that we cannot use existing congestion mechanism. TCP Ohrid is defined with several response functions in dependence of the sending rate. The main idea is to use one primary switch point and five end/switch points instead of using one switch and one endpoint at a log-log scale. It is defined with points that correspond at speed of 5Gbps, 10Gbps, 40Gbps, 100Gbps and 400Gbps, throughput that is supported by the transport medium. Knowing that packet loss probability of fiber networks is in range of 10^{-7} (when 1500 byte packets are used) we decided to observe real packet loss probability values for optical networks and future packet loss probability of 5G packet transmission networks. TCP Ohrid is a 5G protocol that can provide data rates higher than 5Gbps under heavy user mobility and 400 Gbps backhaul data rates in the core network. Today there are patented solutions (at physical layer) that assure ultra-high speed transmission with a capacity of 10^{15} bits/s. In 2005, 100Gbps Ethernet was tested and in 2010 commercial active and passive equipment was available. IEEE Taskforce has completed standardization of local area network interfaces for 100Gbps Ethernet. We can recall that IEEE P802.3bj supports BER better or equal at 10^{-12} and 100 Gbps (over copper), IEEE P802.3bs supports BER better or equal at 10^{-13} and MAC data rate of 400Gbps. IEEE 802.3ba supports BER better than or equal at 10^{-12} and MAC data rates of 40Gbps and 100 Gbps. Considering all stated above we can decide which packet loss probability values and data rates to be used in order to define the protocol switch points. Note that suggested bit error probabilities are real if we use 1500 byte packets. TCP Ohrid is defined with maximal speed of 400Gbps, $p=10^{-12}$ and β =0.1 or 0.4 in dependence of the used decrease parameter equation. Interpolation can be conducted with help of the following points defined with 5Gbps, $p=10^{-7}$; 10Gbps, $p=10^{-8}$; 40Gbps, $p=10^{-9}$ and 100Gbps, $p=10^{-10}$. In the future networks BER is expected to be over 10^{-15} which represents main reason why we have decided to use this value as upper constraint. Equations of line are defined among following points w-w1, w1-w2, w2-w3, w3-w4, w4-w5 and w-w2, w-w3, w-w4, w-w5 as presented at Figure. 1. Five basic response functions denoted as TCPO1, TCPO2, TCPO3, TCPO4, TCPO5 and additional four TCPO6, TCPO7, TCPO8 and TCPO9 are defined.

All of the response functions are uniquely defined with an equation of line joining two points with different slope. Several algorithms can be defined. The protocol will start with a standard TCP Reno/NewReno mechanism, afterwards it will use TCPO1 and in case of lossless environment after passing w1 value it will use TCPO2 as a response function, if loss is detected than TCPO6 is used (even if the calculated window size after loss is below the switch point value, after passing w=38, TCPO6 will be used as a protocol response function because it will provide improved window growth). After cwnd reaches w2 value TCPO3 will be used and if loss is detected TCP07 will become actual response function. When w3 value is reached TCPO4 is used and if loss is detected TCPO8 will be used or TCP05 will be used after w4 value is reached. When loss is detected the protocol switches to use TCPO9 as a response function, after passing w5 value current response function is preserved. In case of three consecutive timeouts the protocol counters are restarted and it restarts the initial usage of the response functions according the actual congestion window value (cwnd). It is interesting to notice that this protocol modifies the initial behavior of Reno/Newreno protocol used during slow mode of operation. In order to calculate the window value after loss we use:

$$w_{new} = \frac{w_{\max} + w_{\min}}{2} \tag{3}$$

Where w_{max} denotes the highest window value achieved prior to the loss that occurs and w_{min} is the decreased window value after the loss that corresponds with the second line of the equation (1). Obtained value places the cwnd in the middle between the highest achieved value before the loss and the calculated value that should be obtained after the loss.

A similar window calculation algorithm is used by BIC protocol [8]. We find usage of (3) more interesting because it impacts β calculation. Till now we have defined the protocol response function variation and its loss window calculation. Packet loss probability p can be calculated for a given value of cwnd and α parameter is defined with equation (2). The β parameter is:

$$\beta_{calculated}(w) = (0.1 - 0.5)\frac{\log(w) - \log(38)}{\log(w_5) - \log(38)} + 0.5 \tag{4}$$

Where w represents the present value of the congestion window of interest and w5 is the congestion window size that defines the fifth switch point. Calculated value of β should be calibrated having in mind that the window calculation after the loss is modified, when equation (3) is used to calculate the window value after loss in slow mode of operation. If we substitute all known values in (3) we obtain

$$w_{new} = (1 - \frac{\beta}{2})w_{\max} \tag{5}$$

Equation that indicates new β value is:

$$\beta_{new} = \frac{\beta_{calculated}}{2} \tag{6}$$

This equation is used only in case of time out (case when NewReno/Reno protocols are used, slow mode of operation), afterwards the protocol proceeds with standard β calculation. Since we have defined all important protocol parameters we can implement them in Network Simulator 2 (ns2). We have decided TCP Ohrid to use β calculation according to the following equation:

$$\beta_{new} = 0.5 - \beta_{calculated} \qquad (7)$$

With this equation we achieve increased concavity of β function that provides low cwnd decrement for low values and high cwnd decrement when timeout occurs for large window values. $\beta_{calculated}$ is obtained with (4). Usage of (7) provides improved protocol performance for large cwnd values when congestion occurs because the window will be decreased for 40% of the previous value. Lowest β_{new} value is bounded by 0.1 and the highest with 0.4. If the result of equation (7) is lower than 0.1 than β_{new} value is rounded at 0.1.

Figure 1. *Response Function Choice of TCP Ohrid*

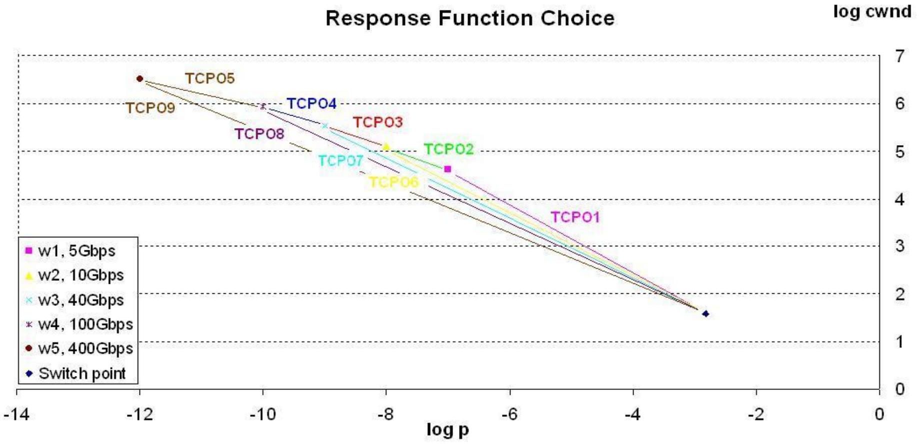
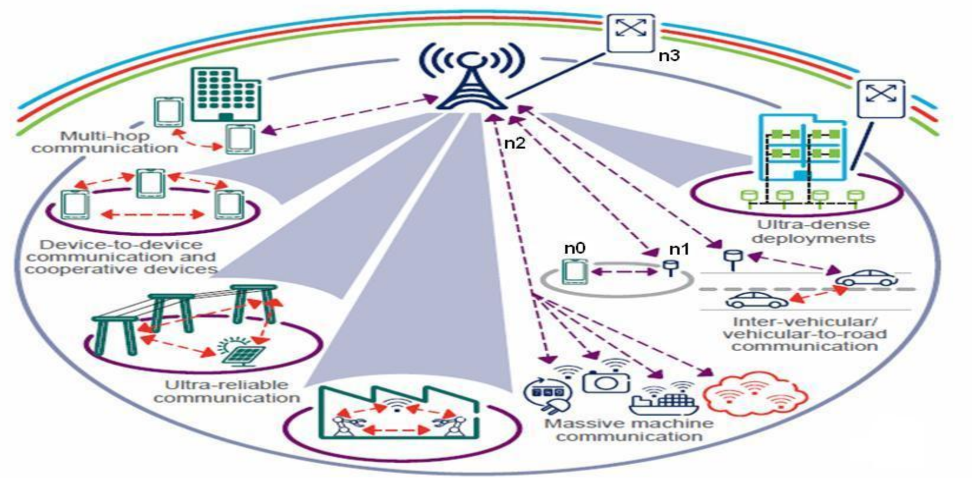


Figure 2. *Simulation Scenario. High Speed Backhaul Link is used between Base Station and Core Network Gateway. We Define Wireless Medium between n0 and n1, n1 is Wirelessly Connected with Macro Base Station n2 and Optical Link n2-n3 is Defined between the Macro Base Station n2 and the Network Gateway n3.*



Simulation Scenario

We study the simple simulation environment presented at Figure 2 assured by Network Simulator 2 (ns2). We simulate mobile user (n0) communication with micro base station (n1) and macro base station (n2) through backhaul high speed link n2-n3 connected with adequate network gateway (n3).

Thr parameter of interest is congestion window size variation; we know that it is directly related with throughput values. Packet size is set at 1500 bytes. Simulation lasts 230 sec. Buffers are Drop Tail; buffer size is set at 100% of the product of bottleneck capacity by the largest RTT divided by packet size. Link speed can vary and high speed link has RTT value of 100 ms. We have conducted simulations for 5, 10, 40 and 100 Gbps and adequate buffer size limitations. Since the max window size is predefined, similar results are obtained regardless the speed and buffer size constraints. Obtained results are presented at Figure 3 and Figure 4.

Figure 3. *Cwnd Change when 5, 10, 40 and 100 Gbps Links are used with Adequate Buffer Sizes (Max ns2 Window Bound Set at 20 pkts)*

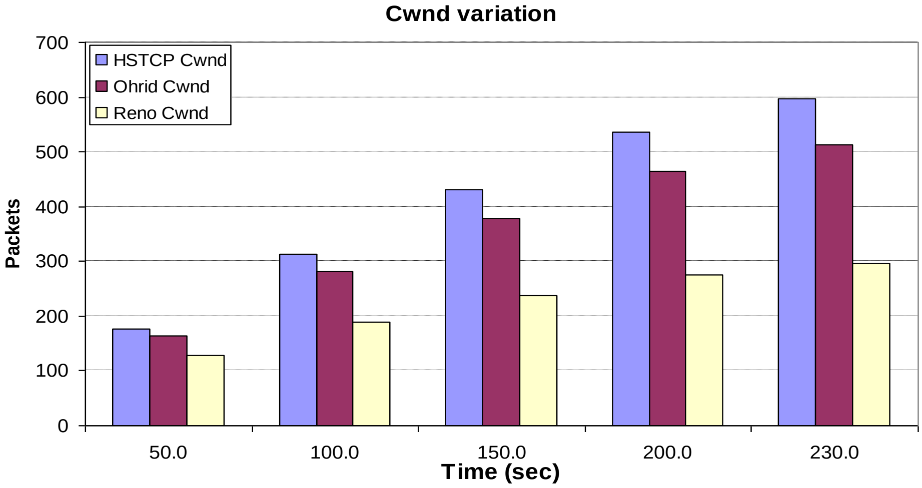
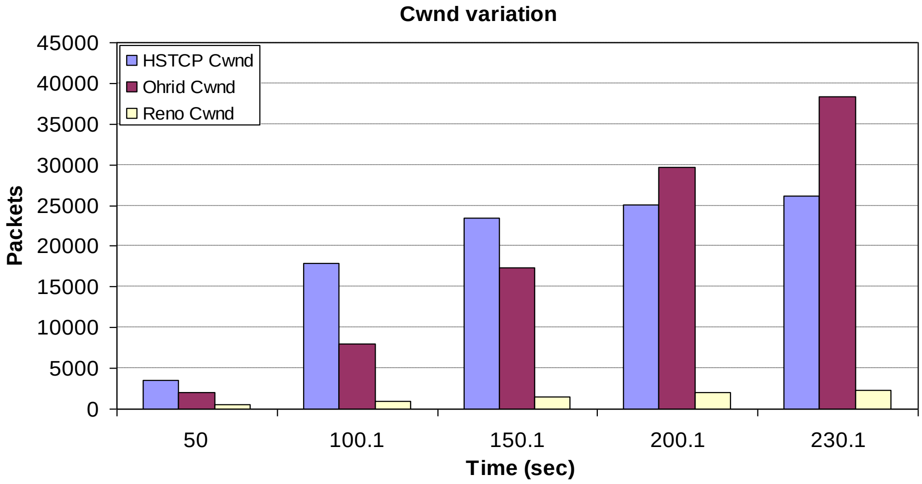


Figure 4. *Cwnd Change when 5, 10, 40 and 100 Gbps Links are used with Adequate Buffer Sizes (Max ns2 Window Bound Set at 80000 pkts)*



At Figure 3 we have presented cwnd change when link capacity is set at 5, 10, 40 and 100 Gbps. Simulation lasts 230 sec and the window size is set at ns2 default value of 20 packets. This figure justifies the defined parameters of Ohrid protocol. It can be noticed that Reno has the smallest window growth which directly impacts the throughput that can be achieved when this protocol is used. Ohrid has shown larger window growth compared with Reno and smaller than HSTCP. This proves that Ohrid is friendlier with existing protocols while sustaining its high speed performances than HSTCP. This protocol behavior assures that the TCP Ohrid can be used to buffer the negative impact of wireless medium especially in case of heavy user mobility. At Figure 4 we have presented results when maximal ns2 window parameter is set at 80000 pkt. At the beginning of the simulation Reno protocol achieved the lowest cwnd values while HSTCP cwnd change is better that the one obtained by Ohrid protocol mainly because of the rapid growth of the HSTCP response function. TCP Ohrid has presented improved performances which justifies the novel protocol design. The protocol uses more efficient set of response functions in case when the window size has larger value than w1 packets. It is obvious that the Ohrid protocol provides higher data rates than HSTCP. Figure 4 assures us that the protocol is robust in case of increased terminal mobility, cwnd growth will be lower, hence making the protocol capable to cope wireless medium impact at the given throughput when the user is mobile and away from the base station. When the terminal is static or near the base station and the number of active users of the network is low than higher data rates can be achieved. Therefore, this protocol is appropriate for use in cell communication and for backhaul data transfer.

Conclusions

In this paper we have presented and evaluated the TCP Ohrid, protocol that has shown improved cwnd behavior. Increased friendliness and higher data rate performances of TCP Ohrid are justified. The protocol is defined to be used in the future 5G data networks regardless of the engaged MAC Layer. It is defined with several response functions for speeds of 5, 10, 40, 100 and 400 Gbps. Calculation of β parameter is done with help of equation (7) which guarantees improved protocol behavior especially in high congested networks. Equation (3) is used to improve Reno/Newreno protocol performance till reaching cwnd value of 38 packets that provides a faster increase when loss occurs during a slow mode of operation. All used parameters can be fine tuned in order to properly design the protocol. Development of high speed protocols that will sustain 5G network capacity is essential hence user can benefit the network capacity. TCP Ohrid is a good candidate since it can support high transmission speeds and provide efficient usage of network capacity up to speeds of 400 Gbps.

References

[1] M.Duke, R.Braden,W.Eddy, E.Blanton, A.Zimmermann, “A Roadmap for Transmission Control Protocol (TCP)”. Internet Engineering task Force (IETF), RFC7414, February 2015.

[2] T. Janevski, “NGN Architectures, Protocols, and Services”, John Wiley and Sons, UK, 2014.

[3] J. Rodriguez, “Fundamentals of 5G Mobile Networks”, John Wiley and Sons, UK, 2015.

[4] X. Ge, H. Cheng, M. Guizani, T. Han, “5G wireless backhaul networks: challenges and research advances”, Network IEEE, Nov.-Dec. 2014.

[5] S. Barbarossa, S. Sardellitti, P. Di. Lorenzo, “Communicationg While Computing”, IEEE Signal Processing magazine, Oct. 2014.

[6] T. Kelly, “Scalable TCP: improving performance in highspeed wide area networks,” Computer Communications Review, vol. 32, no. 2, April 2003.

[7] S. Floyd, “HighSpeed TCP for large congestion windows,” RFC 3649, 2003.

[8] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestion control for fast, long distance networks,” in Proc. IEEE INFOCOM, vol. 4, March 2004, pp. 2514–2524.

[9] D. Leith, “H-TCP:TCP congestion control for high bandwidth-delay product paths,” IETF Internet Draft, http://tools.ietf.org/html/draftleith-tcp-htcp-06, 2008.

[10] I. Rhee and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” SIGOPS Operating Systems Review, vol. 42, no. 5, pp. 64–74, July 2008.

[11] G. Marfia, C. Palazzi, G. Pau, M. Gerla, M. Sanadidi, and M. Roccetti, “TCP Libra: Exploring RTT-Fairness for TCP,” UCLA Computer Science Department, Tech. Rep. UCLA-CSD TR-050037, 2005.

[12] C. Caini and R. Firrincieli, “TCP Hybla: a TCP enhancement for heterogeneous networks,” International J. Satellite Communications and Networking, vol. 22, pp. 547–566, 2004.

[13] A. Baiocchi, A. P. Castellani, and F. Vacirca, “YeAH-TCP: yet another highspeed TCP,” in Proc. PFLDnet, ISI, Marina Del Rey (Los Angeles), California, February 2007.

[14] R. King, R. Baraniuk, and R. Riedi, “TCP-Africa: an adaptive and fair rapid increase rule for scalable TCP,” in Proc. IEEE INFOCOM, vol. 3, March 2005, pp. 1838–1848.

[15] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “A compound TCP approach for high-speed and long distance networks,” July 2005.

[16] S. Liu, T. Basar, and R. Srikant, “TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks,” in Proc. First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), 2006.

[17] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, “FAST TCP: motivation, architecture, algorithms, performance,” IEEE/ACM Trans. Netw., vol. 14, no. 6, pp. 1246–1259, 2006.

[18] J. Sing and B. Soh, “TCP New Vegas: Improving the Performance of TCP Vegas Over High Latency Links,” in Proc. 4th IEEE International Symposium on Network Computing and Applications (IEEE NCA05), 2005, pp. 73–80.

[19] K. Yamada, R. Wang, M.Y. Sanadidi, and M. Gerla ” TCP Westwood with Agile Probing: Dealing with Dynamic, Large, Leaky Pipes” IEEE Communication Society, 2004.

[20] D. Kliazovich, F. Granelli, and D. Miorandi, “Logarithmic window increase for TCP Westwood+ for improvement in high speed, long distance networks,” Computer Networks, vol. 52, no. 12, pp. 2395–2410, August 2008.
[21] H. Shimonishi, T. Hama, and T. Murase “TCP-Adaptive Reno for Improving Efficiency-Friendliness Tradeoffs of TCP Congestion Control Algorithm”, in Proc. 4th International Workshop on Protocols for Fast Long Distance Networks, 2006.
[22] K. Kaneko, T. Fujikawa, Z. Su, and J. Katto, “TCP-Fusion: a hybrid congestion control algorithm for high-speed networks,” in Proc. PFLDnet, ISI, Marina Del Rey (Los Angeles), California, February 2007.
[23] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, “Designing intelligent energy harvesting communication systems,” IEEE Commun.Magazine, vol. 52, no. 1, pp. 210 - 216, Jan. 2014.