Dynamic ACK skipping in TCP with Network Coding for Power Line Communication Networks

著者	Nguyen Viet Ha, Le Van Hau, Tsuru Masato
journal or publication title	2020 22nd International Conference on Advanced Communication Technology (ICACT)
year	2020-04-09
URL	http://hdl.handle.net/10228/00007968

doi: http://dx.doi.org/10.23919/ICACT48636.2020.9061477

Dynamic ACK skipping in TCP with Network Coding for Power Line Communication Networks

Nguyen Viet Ha[1], Le Van Hau[2,3], Masato Tsuru[1]
[1]Kyushu Institute of Technology, Fukuoka, Japan
[2]University of Science, Ho Chi Minh City, Vietnam
[3]Vietnam National University, Ho Chi Minh City, Vietnam
nguyen.viet-ha503@mail.kyutech.jp, lvhau2016@gmail.com, tsuru@cse.kyutech.ac.jp

***Abstract*—Transmission Control Protocol (TCP) still plays an essential role in various user applications for end-to-end reliable data transmission. However, TCP cannot get a high goodput performance in the lossy networks because it considers any packet loss to be a congestion signal and decreases the congestion window mistakenly. Therefore, TCP with Network Coding (termed TCP/NC) was proposed to recover the packet loss at the sink without retransmission if the number of coded packets is enough. However, the ACK packet needs to be sent for any arriving coded packet as a feedback of the end-to-end channel condition, resulting in a lower transmission performance in half-duplex networks, e.g., Power Line Communication. Therefore, we propose the ACK-Skipping scheme for TCP/NC to limit the number of ACK packets but still retain the necessary information, e.g., for channel estimation. The simulation result on ns-3 (Network Simulation 3) shows that the proposal achieves a higher goodput on PLC environment compared to TCP with Selective Acknowledgment and TCP Westwood+ as well as the recent variant of TCP/NC.**

***Keywords*—TCP/NC, Network Coding, ACK-Skipping, PLC, Half-duplex, ACK Scheduling**

I. INTRODUCTION

Transmission Control Protocol (TCP) with its advantages on connection-oriented and congestion control is still a preferred option for reliable data transmission. However, its congestion control feature does not work correctly in non-congestion packet loss environments (e.g., wireless networks) when it considers any packet loss to be a congestion signal and decreases the sending rate by reducing the congestion window (CWND). Instead, the CWND should be maintained if the packet loss is from a channel to overcome the temporary lossy conditions. The mistaken decreasing of the CWND results in low transmission performance caused a bad user experience. Although some TCP variants try to distinguish congestive and non-congestive losses based on Round Trip Time (RTT) monitoring, such as TCP Veno [1], TCP Westwood+ [2], they are only effective at a small link loss rate, e.g., lower than 0.01 in Wifi, WiMAX [3], and lower than 0.001 in the satellite environment [4]. However, in networks with heavy non-congestion loss (e.g., due to environmental errors), all TCP variants based only on passive retransmission are unable to get a high goodput. Therefore, a forward erasure-packet correction was introduced by combining Network Coding (NC) with TCP (termed TCP/NC) [5]. TCP/NC introduces a new NC layer between TCP and IP layer shown as Fig. 1 to handle the lost packets. At the source, NC layer receives n TCP segments (referred to as original packets) from TCP layer, combines them to m combination packets (referred to as combinations) with $m>n$ for lossy links, and forwards them to IP layer. The sink is expected to recover all n original packets if the number of lost combinations is no more than $m-n$. While TCP/NC can keep a stable sending rate, it will work negatively in network congestion. Therefore, TCP/NC is considered not necessarily universal but essentially useful in certain environments with particular settings.

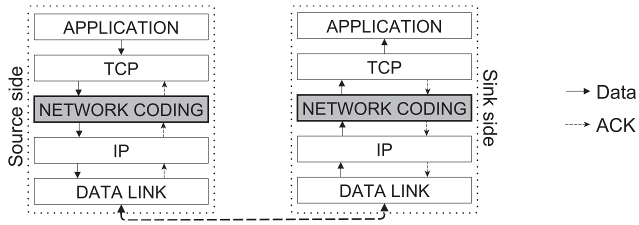

Figure 1. NC layer in TCP/IP model.

Some TCP/NC variants were introduced. Network Coding TCP (NCTCP [6]), TCP-NC with adaptive redundancy factor (TCP-NCAR [7]), and Dynamic Coding (DynCod [8]) focus on link loss rate estimation and NC parameters adaptation (n and m) to work well in overtime frequently changed channels. TCP/NC with Loss Rate and Loss Burstiness Estimation (TCP/NCwLRLBE [9]) estimates the channel condition in burst loss environments and flexibility adjusts the NC parameters without adverse effect on the current settings. It also adopts enhanced retransmission schemes to reduce the retransmitted packet loss to avoid TCP timeout (TCP TO). Acknowledgment (ACK) packets are usually used as a feedback of the end-to-end channel conditions. TCP/NCwLRLBE uses Packet Identification (*Pid*) to identify each combination and *Pid-Echo-Reply* in NC-ACK header added inside the regular TCP ACK packet to help the source determine the lost packets. In another side with TCP-NCAR, the redundancy factor $R=\frac{m}{n}$ is increased or decreased based on the number of the consecutive duplicate ACKs. As a result, all these protocols

The research results have been achieved by the "Resilient Edge Cloud Designed Network (19304)," the Commissioned Research of National Institute of Information and Communications Technology (NICT), and by JSPS Grant-in-Aid for Scientific Research (KAKENHI) Grant number 19K21535, Japan.

need to return as much as ACK packets, e.g., returning the ACK packet for every arrival data packet to estimate the accurate NC parameters. Therefore, most of the protocols do not use the TCP Delayed-ACK function by default. In TCP Delayed-ACK, several ACK packet responses may be combined together into a single response. On other words, one ACK packet will be returned in every d data packets (default of 2) to improve network performance. In the half-duplex network, only one packet can transfer on the link at a time. Even though the ACK packet size is small, but it still occupies the link due to the media access control protocol (e.g., Carrier-sense multiple access - CSMA). Therefore, the TCP/NC variants will not work well in such networks, e.g., Power Line Communication network (PLC).

Besides, the bi-directional loss, which ACK packet is also lost, often appears in non-congestion loss conditions is a strong obstacle for reliable and efficient data transmission in severe environments, especially with TCP/NC variants which need to receive enough ACK packet. A new information field in NC-ACK header was introduced in our previous study [10], [11] by which the sink notifies the source about arrivals of recent combination in one ACK packet. Thus, even if some ACK packets are sparsely missed, the source can correctly know the combination loss, including its burstiness and properly estimate the NC parameters. With this proposal when TCP/NC no need to receive all ACK packets, TCP Delayed-ACK can be re-enabled on TCP/NC variants. However, the constant of the number of ACK packets that can be skipped (s) will not achieve the best transmission performance. There are some strategies for control s such as ACK filtering, TCP Sender Adaptation [12] but most of them consider for minor of no non-congestion packet loss environments. In this paper, therefore, we continue our preliminary work of [13] to propose a new scheme called ACK-Skipping for TCP/NC (TCP/NCwAS) to send only one ACK packet in every s combinations. And s can be estimated based on the channel conditions e.g., both directional packet loss rate.

The remainder of this paper is organized as follows. Sect. II introduces the overview of TCP/NC. Sect. III reviews the PLC system and its channel. Sect. IV explains the TCP/NCwAS scheme. The simulation evaluation is presented in Sect. V. And the conclusion is finally given in Sect. VI.

II. TCP/NC Overview

A. Network coding in protocol stack

NC layer handles the packet flow between the TCP and IP layer to recover the packet loss. If the NC layer can recover all packet losses, the TCP layer is unaware of the loss events; thus, the goodput is not affected by the lossy channel. Besides, NC layer returns ACK packet with ACK number determining based on the degree of freedom and the seen/unseen definition [5]. The sink can return the different ACK number for every received packet without waiting for decoding all the packets. When the sink receives enough combination packet, all original packets will be decoded. Therefore, the CWND is kept increasing even though some combination packets are lost. Thus, the goodput performance is stable through lossy channels.

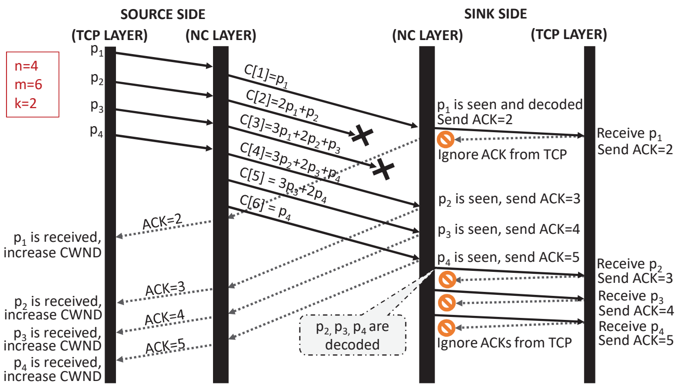

Figure 2. Network coding process.

B. Coding process

TCP/NC allows the source to send m combinations (C) created from n original packets (p) with $m \geq n$ using Eq. (1) where α is the coefficient (encoding process). If the number of lost combinations is less than $k=m-n$, the sink can recover all the original packets without retransmission except for the case of the linearly dependent combinations (decoding process). TCP/NC combines original packets into a combination packet. TCP/NC uses a sliding method to combine no more than $k+1$ original packets into one combination. Besides, α is selected randomly, i.e., by random linear coding. All operators are performed on a Galois field with XOR operations and a lookup table; hence, the complexity of computation is small enough to apply to real systems. In the example in Fig. 2, $C[2]$ and $C[3]$ are lost when sending through a lossy network. Until $C[6]$ is received, the sink has four equations and four unknown variables. packets can be decoded even though two combinations are lost.

$$C_i = \sum_{j=a}^{b} \alpha_{ij} p_j \ ; \quad i=1,2,3,...,m \ ; \qquad (1)$$

$$a=\begin{cases} 1 & (i \leq k) \\ i-k & (i>k) \end{cases} ; \quad b=\begin{cases} i & (i \leq n) \\ n & (i>n) \end{cases}$$

C. TCP functionality

TCP functionality has been studied and worked stably in a long history; thus, TCP/NC takes all these advantages, such as retransmission and congestion control mechanisms. The source must retransmit the packets when the number of packet losses is larger than k. In that situation, both the TCP and NC layer at the source receives many duplicated ACK equaling the oldest "unseen" packet. The NC layer only needs to wait for the retransmission from the TCP layer. Increasing or decreasing the CWND is also controlled by TCP layers, not NC layer. Going back the example in Fig. 2, when combination $C[i]$ is received, p_i is always expressed to the set of p_k with $k>i$; thus, all packets are seen.

D. TCP with Loss-rate and Loss-burstiness Estimation

We proposed TCP/NCwLRLBE [9] to automatically adjust the NC layer behaviors for adapting to time-varying channel in both random and burst loss conditions. The loss conditions are estimated from the continuous observation of the packet transmission between the source and the sink. TCP/NCwLRLBE calculates the NC parameters based on the probability distribution of the number of lost packets and the loss burstiness in one *CW* and able to update NC parameters in the coding system promptly.

III. Power Line Communication Overview

A. PLC and IoT

Nowadays, broadband PLC technology with its merit in speedy and stability that emerges as one of the top wire communication candidates for IoT applications (e.g., smart grid, smart city, smart home) when the electrical system presents everywhere and be connected. Broadband PLC works at high-frequencies in range of 1.8 to 250 *MHz*, high data rates up to 100 *Mbps*, and is mostly implemented for shorter-range applications. Depending on requirements in data rate, supported applications, and the specifics of grid topologies, the certain frequency bands are taken into account consideration for applying. However, the biggest challenge for most PLC systems is the complexity of channel environment where many components such as internal factors (e.g., propagation constant of cable, the reflection coefficient of a circuit) and external factors (e.g., interference) which significantly affect to the quality of signal and transmission performance. In order to overcome existing difficulty as mentioned above and bring good user experience in IoT networks, it is necessary to have innovations to customize functions not only for MAC and PHY layers but also for upper layers. In this paper, broadband PLC version we choose for the simulation is IEEE Standard for Medium Frequency (less than 12 *MHz*) Power Line Communications for Smart Grid Applications (IEEE Std. 1901.1-2018) which approved in May 2018 [14].

B. PLC channel

PLC channel can be modeled based on Transmission Line Theory where lines and passive power-grid components (e.g., loads, transformers) connected to it and establishing a two-port network. The Channel Transfer Function (CTF) is presented for internal factor, that significantly reduces the power of the signal from the transmitter (Tx) to receiver (Rx). CTF is obtained relying on impedance, propagation constant, and cable length. Considering on each edge (an unit to connect two nodes), CTF of the i^{th} edge is calculated by Eq. (2) [15], where γ_i is the propagation constant of the i^{th} line piece, ρ_i is the reflection coefficient of the i^{th} edge, and l_i is the length of the i^{th} line, thereby the overall transfer function from Tx to Rx ($H(f)$) is calculated by multiplication of the CTF of all edges linked together to connect the source to the sink as Eq. (3).

$$H_i(f) = \frac{1+\rho_i}{e^{\gamma_i l_i} + \rho_{i+1} e^{-\gamma_i l_i}} \tag{2}$$

$$H(f) = \prod_i H_i(f) \tag{3}$$

As a result, an attenuation of power of the signal from Tx to Rx which is constituted by the CTF follow as Eq. (4), where

TABLE 1. Parameters of the log-linear approximation for physical abstraction model with modulation.

Channel Model	Modulation Coding Scheme	A	B
"excellent"	QPSK[1]; 1/2	-5.1	2.5
	16-QAM[2]; 1/2	-4.3	8.4
	64-QAM; 16/21	-4.7	16.3
"good"	QPSK; 1/2	-5.3	2.7
	16-QAM; 1/2	-4.1	8.1
	64-QAM; 16/21	-4.5	17.2
"medium"	QPSK; 1/2	-4.9	2.7
	16-QAM; 1/2	-4.2	8.0
	64-QAM; 16/21	-4.6	17.8
"bad"	QPSK; 1/2	-5.5	2.9
	16-QAM; 1/2	-4.4	7.0
	64-QAM; 16/21	-5.1	18.0

[1] QPSK: Quadrature Phase Shift Keying
[2] QAM: Quadrature Amplitude Modulation

Psd_{tx} is power spectral density (PSD) of the signal at the transmitter, Psd_{rx} is PSD of the signal at the receiver.

$$Psd_{rx} = Psd_{tx} \times |H(f)|^2 \tag{4}$$

The Signal-to-Interference-plus-Noise Ratio (SINR) is an element that mainly impacts to the packet loss rate of the channel and decides to quality of the signal. SINR is calculated by Eq. (5) where Psd_{rx} is addressed by Eq. (4); Psd_I and Psd_N is PSD of interference and background noise relatively.

$$SINR = \frac{Psd_{rx}}{Psd_I + Psd_N} \tag{5}$$

The study [16] describes the physical abstraction methodology suitable to predict the physical layer error performance in a realistic PLC channel environment. They proposed a simple log-linear approximation of PER vs. Capacity performance shown in Eq. (7) where *C* is capacity performance and value of *A* and *B* is chosen in Table 1. The capacity performance is calculated by Eq. (6), where k is the k^{th} sub-carrier.

$$C = \frac{1}{N}\sum_{k=1}^{N} log_2(1 + SINR_k) \tag{6}$$

$$PER_{appr}(C) = 10^{A \times C + B} \tag{7}$$

IV. TCP/NC with ACK-Skipping (TCP/NCwAS)

Since the source needs to receive all ACK packets returning for all data packets to calculate the channel conditions (e.g., packet loss rate, burst loss size), ignoring some ACK packets will affect the accurateness of estimation processes. However, in half-duplex communication like PLC, sending all ACK packets affect the transmission performance. In [11], we proposed to add a new field in NC-ACK header called "Packet loss Sequence" (PLS) to send the recent packet arrival information from the sink to the source. PLS includes a 32-bit field representing the status (received or lost) of recent thirty-two combinations until the latest successfully received combination identified by the current *Pid*. Therefore, even

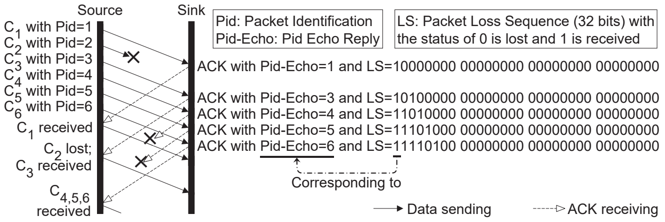

Figure 3. Illustration of Packet Loss Sequence updating at sink.

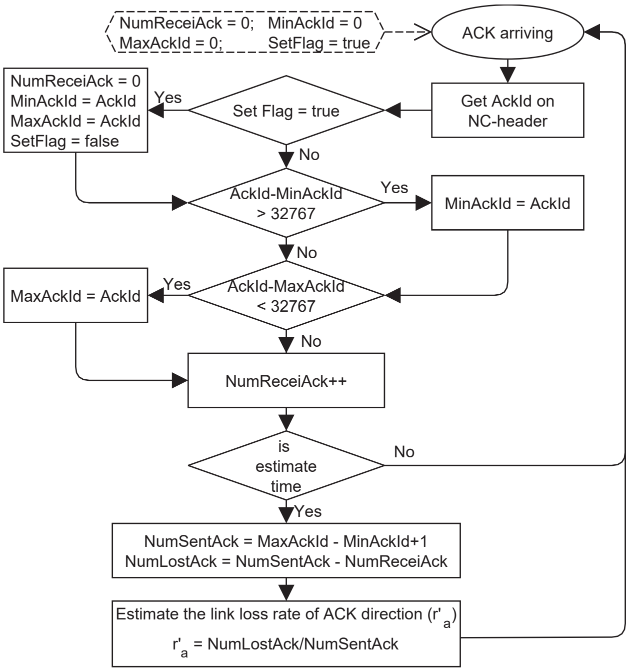

Figure 4. Flowchart: Estimating the link loss rate of ACK receiving direction.

though some previous ACK packets are lost, the source is able to collect all information to estimate the channel condition. It is the same behavior for ignoring some ACK packets if the number of ignored ACK packets no more than 32. Fig. 3 illustrates how the source exactly knows the packets received at the sink when two ACK packets are missed.

Taking advantage of [11], in this paper, we proposed the scheme to let the sink returns only one ACK packet and skips other s ACK packets to optimize the network performance. The skipped ACK packet will be sent if no ACK packet sent in 200 ms which follows the setting of TCP Delayed-ACK [17]. Instead of keeping s constantly like TCP Delayed-ACK (default of 2), s can be dynamically estimated by the source based on the channel conditions, e.g., the link loss rate of data sending direction (r_d) and the link loss rate of ACK receiving direction (r_a). Note that s must be less than 32 due to the limitation of the PLS field size. All the estimation processed can be done by the source. Then, the source will send the s related information (2 bytes unsigned integer) to the sink via the NC header. We added a new two-bytes field to the NC-ACK header called "ACK identification" (*AckId*) to estimate the r'_a as shown in the flowchart of Fig. 4. The smoothing

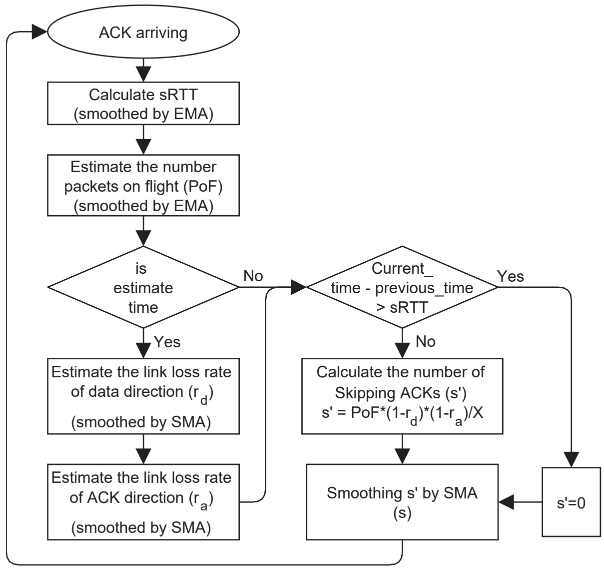

Figure 5. Flowchart: Estimating the number of Skipping ACK packet.

value r_a is calculated by using the Simple Moving Average (SMA) to increase the accuracy of the estimation. Noted that r_d is estimated by the mechanism of TCP/NCwLRLBE [9].

The source estimates the number of data packets being on-the-flight (termed *PoF*) in one sending windows. After that, we estimate the number of ACK packets may be received at the source ($total_{ack}$) based on r_d and r_a in the case that the sink return ACK packet for every received data packet. $total_{ack}$ is calculate by Eq. (8). The number of ACK can be ignored s' is half of $total_{ack}$. And the average value (s) of s' is calculated based on SMA. The detail of the overall algorithm is shown in the flowchart of Fig. 5.

$$total_{ack} = PoF \times (1 - r_d) \times (1 - r_a) \tag{8}$$

If the sink sends an ACK packet with duplicated ACK number, this ACK packet will be sent immediately. Otherwise, the ACK packet is scheduled to send after 200 ms. The schedule is created only in the case that the number of scheduled packets is less than s. Besides, since the sink can receive more than n combinations, but the decoding process requires only n combinations, the sink needs to acknowledge only n combination. However, all ACK packets, including that of redundancy packets, need to return to the source to estimate more accurately the channel conditions. If the ACK packet sending to the channel is for redundancy combination, it will not be forwarded to source's TCP layer. Therefore, the next ACK packet must be sent without scheduling. The whole process is shown in the flowchart of Fig. 6.

V. Simulation result

A. Simulation setup

The simulation is accomplished by Network Simulator 3 (ns-3) [18] most widely used discrete-event simulator for packet networks. We compare the long-term goodput performance of the regular TCPs (TCP SACK and TCP Westwood+),

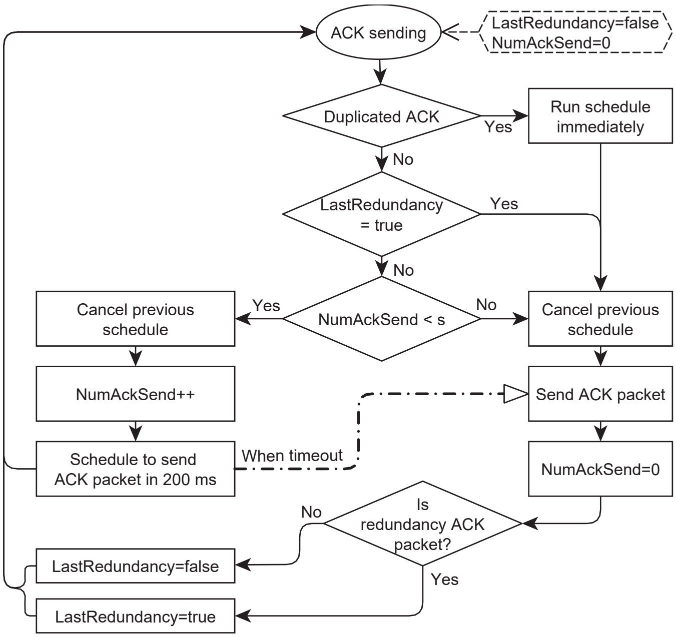

Figure 6. Flowchart: ACK Sending Scheduling.

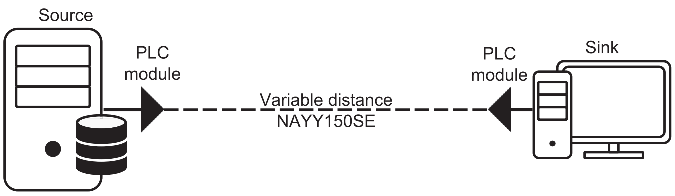

Figure 7. Simulation topology.

TABLE 2. Physical parameters.

Parameter	Value
Frequency Band	0.024414 MHz to 12.475586 MHz
Modulation	16-QAM
Encoded Rate	$^1/_2$
Number of IFFT[1] Point	1024
Number of sub-carriers	512
OFDM[2] duration	64.24 μs
Preamble duration	40.96 μs
PSD of signal at source	10^{-6} W/Hz
PSD of background noise	15^{-9} W/Hz
Impedance at source	100 ohm
Impedance at sink	40 ohm
Channel model	"Good"

[1] IFFT: Inverse Fast Fourier Transform
[2] OFDM: Orthogonal Frequency-Division Multiplexing

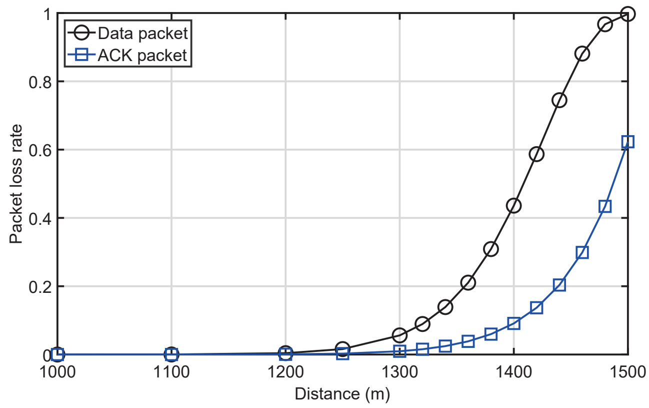

Figure 8. Packet loss rate of Data packet and ACK packet vs. distance of cable.

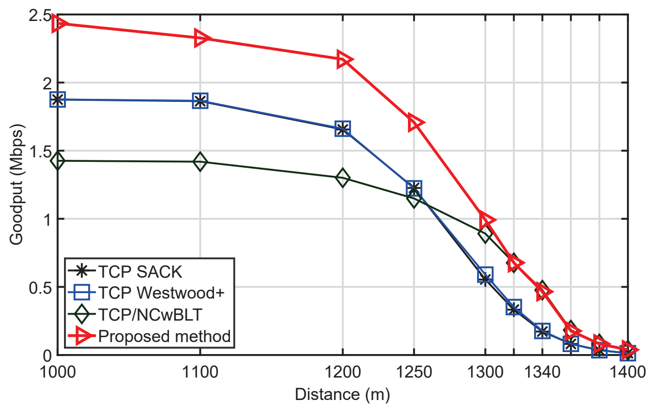

Figure 9. Goodput performance comparison.

a recent variant of TCP/NC [11], and TCP/NCwAS on a simple network topology consists of one source and one sink on either side of the electrical cable type NAYY150SE shown in Fig 7. The cable distance is variable from 1000 m to 1500 m. The PLC channel modeling setting is based on [15] and the physical parameter setting is modified based on the standard of IEEE Std 1901.1-2018 as shown in Table 2. The TCP payload size is 536 bytes. The minimum TCP TO is 1 s. 100-*Mbytes* data are sent on each simulation. And the simulations are run at 30 times to obtain the average goodput performance.

B. Distance vs Packet loss rate

The Fig 8 shows the relationship of the packet loss rate and the distance of the cable. Because the 536-bytes-data packet transports on 6 OFDM (Orthogonal Frequency-Division Multiplexing) symbols while the ACK packet is on only one OFDM symbol, the packet error rate of data packet equals $r_d{=}1{-}(1{-}r_a)^6$ where r_a is the packet loss of ACK packet calculated based on the formula Eq. (7).

C. Goodput performance

In this simulation, we observe the goodput performance when changing the cable distance from 1000 m corresponding to the packet loss rate of almost 0 to 1400 m corresponding to the packet loss rate of about 0.4. In the result shown in Fig 9, we can see the proposed method (TCP/NCwAS) outperforms the TCP variants in all distance, especially in low packet loss rate (less than 1250 m). When TCP variants with the default delayed-ACK setting of 2, the TCP/NCwAS can adapt the number of skipping ACK packets based on the channel conditions as shown in Fig. 10. Due to decreasing the number of returning ACK packet, the sink can limit to use the sharing link; thus, the source running TCP/NCwAS can send the data packet more frequent than the case running the standard TCPs.

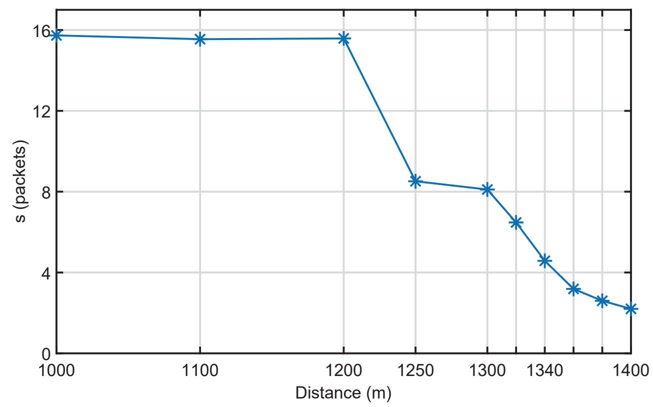

Figure 10. The number of Skipping ACK packets vs. distance.

When comparing to the recent variant of TCP/NC, the goodput performance of TCP/NCwAS is better. TCP/NC even has bad performance compare to the TCP variants when the distance less than 1250 m corresponding to less than 0.01 of packet loss rate. It is because TCP/NC return ACK packet for every received combination. From the distance of 1320 m, both versions of TCP/NC have the same goodput performance when the duplicated-ACK packet frequently occurs, resulting in the benefit of ACK-Skipping is decreased. However, both TCP/NC versions can get a higher goodput performance than the other TCP variants.

VI. Conclusion

In this paper, we have proposed TCP/NCwAS to improve the TCP/NC in half-duplex communication networks. It can determine the number of ACK packets that can be skipped based on the channel conditions. According to the determined skip interval, unnecessary ACK packets will not be sent unless a timeout happens. The proposal outperforms TCP SACK, TCP Westwood+, and the recent variant of TCP/NC (e.g., TCP/NCwBLT) in simple PLC network simulation. As future work, we will evaluate TCP/NCwAS in more complex PLC network simulation including, e.g., multiple terminals, bidirectional data exchange, and different types of noise. Furthermore, the adaptability in time-varying conditions should be investigated.

References

[1] C. P. Fu and S. C. Liew, "TCP veno: TCP enhancement for transmission over wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216–228, 2003.

[2] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. IEEE annual International conference on Mobile Computing and Networking*, Rome, Italy, Jul. 2001, pp. 287–297.

[3] K. Tsiknas and G. Stamatelos, "Comparative performance evaluation of TCP variants in WiMAX (and WLANs) network configurations," *Journal of Computer Networks and Communications*, vol. 2012, no. 9, 2012.

[4] C. Caini, R. Firrincieli, and D. Lacamera, "Comparative Performance Evaluation of TCP Variants on Satellite Environments," in *Proc. IEEE International Conference on Communications*, Dresden, Germany, Jun. 2009.

[5] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network coding meets TCP: Theory and implementation," *Proc. IEEE*, vol. 99, no. 3, pp. 490–512, Mar. 2011.

[6] J. Ridgewell and H. ElAarag, "Network-coded internet-friendly transport protocol," *Journal of Algorithms & Computational Technology*, vol. 11, no. 3, pp. 201–216, Mar. 2017.

[7] H. M. Ruiz, M. Kieffer, and B. Pesquet-Popescu, "TCP and Network Coding: Equilibrium and Dynamic Properties," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 1935–1947, Aug. 2016.

[8] T. V. Vu, N. Boukhatem, T. M. T. Nguyen, and G. Pujolle, "Dynamic coding for TCP transmission reliability in multi-hop wireless networks," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Sydney, Australia, Oct. 2014.

[9] N. V. Ha, K. Kumazoe, and M. Tsuru, "TCP network coding with adapting parameters for bursty and time-varying loss," *IEICE Transactions on Communications*, vol. E101-B, no. 2, pp. 476–488, Feb. 2018.

[10] N. V. Ha and M. Tsuru, "TCP/NC performance in bi-directional loss environments," in *Proc. International Conference on Electronics, Information, and Communication*, Auckland, New Zealand, Jan. 2019, pp. 203–206.

[11] N. V. Ha, T. T. T. Nguyen, and M. Tsuru, "TCP with Network Coding Enhanced in Bi-directional Loss Tolerance," *IEEE Communications Letters*, Dec. 2019, (Early Access).

[12] H. Balakrishnan and V. N. Padmanabhan, "How Network Asymmetry Affects TCP," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 60–67, Apr. 2001.

[13] N. V. Ha, L. V. Hau, T. T. T. Nguyen, and M. Tsuru, "The performance characteristics of TCP with Network Coding in Power Line Communication Network," in *Proc. International Conference on System Science and Engineering*, Quang Binh, Vietnam, Jul. 2019, pp. 267–272.

[14] *IEEE Standard for Medium Frequency (less than 12 MHz) Power Line Communications for Smart Grid Applications*, IEEE Std. 1901.1-2018, May 2018.

[15] F. Aalamifar, A. Schlogl, D. Harris, and L. Lampe, "Modelling Power Line Communication Using Network Simulator-3," in *Proc. IEEE Global Communications Conference*, Atlanta, GA, USA, Dec. 2013, pp. 2969–2974.

[16] K. H. Kim, H. B. Lee, Y. H. Lee, and S. C. Kim, "PHY abstraction methodology for the performance evaluation of PLC channels," in *International Symposium of Power Line Communication and Its Application*, Rio de Janeiro, Brazil, Mar. 2010, pp. 28–32.

[17] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing TCP's retransmission timer," *RFC 5681*, Jun. 2011.

[18] Network simulator 3. [Online]. Available: https://www.nsnam.org

Nguyen Viet Ha received the B.S. degree (2009), M.S. degree (2012) in Electronics and Telecommunications, from University of Science, Ho Chi Minh City, Vietnam and Ph.D. degree (2017) in Computer Science and System Engineering from Kyushu Institute of Technology, Japan. His research interests are transport layer protocols and network coding. He is a member of the IEEE and IEICE.

Le Van Hau received the B.S degree (2015) in Electronics and Telecommunications from Ho Chi Minh City University of Transport and M.S. degree (2019) in Electronics, Computer, and Telecommunications from University of Science, Ho Chi Minh City, Vietnam. His research interests are Network coding, Network performance assurance, and Network monitoring.

Masato Tsuru received the B.E. and M.E. degrees from Kyoto University, Japan in 1983 and 1985, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He has been a professor in the Department of Computer Science and Electronics, Kyushu Institute of Technology since 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the ACM, IEEE, IEICE, IPSJ, and JSSST.