



- (51) International Patent Classification:
H04L 12/807 (2013.01)
- (21) International Application Number:
PCT/MY2015/050119
- (22) International Filing Date:
9 October 2015 (09.10.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
PI2014703305 6 November 2014 (06.11.2014) MY
- (71) Applicant: **UNIVERSITI PUTRA MALAYSIA** [MY/MY]; Putra Science Park, Universiti Putra Malaysia, Serdang, Selangor, 43400 (MY).
- (72) Inventors: **OTHMAN, Mohamed**; Department Of Communication Technology And Network, Faculty Of Computer Science And Information Technology, Universiti Putra Malaysia, Serdang, Selangor, 43400 (MY). **ALJUBARI, Ammar**; Faculty Of Computer Science And Technology, Universiti Putra Malaysia, Serdang, Selangor, 43400 (MY).
- (74) Agent: **NAWAWI, Norunnuha**; Norunnuha Sdn Bhd, No17-2, Jalan Medan 2d, Persiaran Bangi, Bandar Baru Bangi, Selangor, 43650 (MY).
- (81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A METHOD FOR TCP CONGESTION IN MULTI-HOP WIRELESS NETWORK

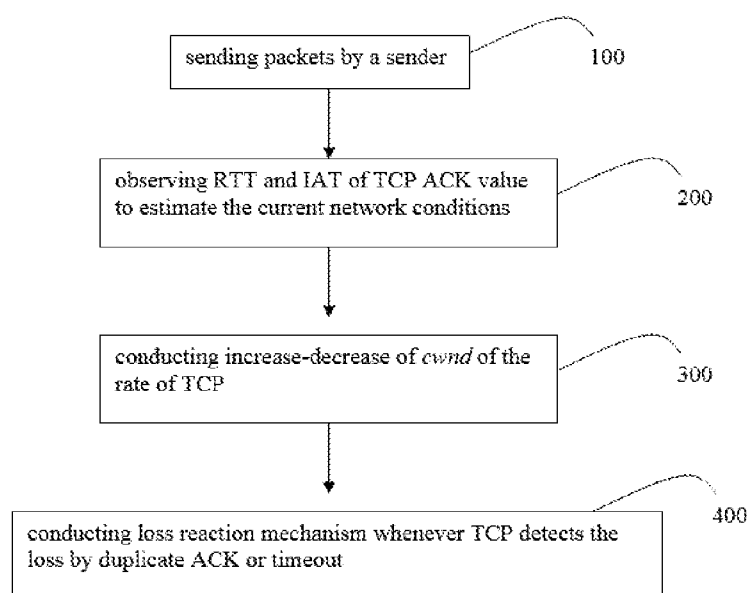


Fig.1

(57) Abstract: The present invention relates to a method for managing congestion in a wireless network. It deploys the step of observing RTT and IAT of TCP ACK value to estimate the current network conditions (200); conducting increase-decrease of cwnd of the rate of TCP (300); and conducting loss reaction mechanism whenever TCP detects the loss by duplicate ACK or timeout (400). The present invention efficiently estimates the network condition, distinguishes between losses, and adaptively controls the data packet transmission is required.

**Declarations under Rule 4.17:**

— as to the identity of the inventor (Rule 4.17(i))

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

— as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

— with international search report (Art. 21(3))

Description

Title of Invention : A METHOD FOR TCP CONGESTION IN MULTIHOP WIRELESS NETWORK

FIELD OF INVENTION

[0001] The present invention relates to a method for managing congestion in a wireless network protocol.

Background Art

[0002] The widespread deployment of multi-hop wireless networks (MWNs) that employ IEEE 802.11 as the underlying technology has attracted a great deal of research attention in both academia and industry. Transmission control protocol (TCP) is the most dominant reliable transport protocol that serves a large number of applications including web (HTTP), file transfer (FTP), email (SMTP), and remote access (Telnet) traffics. Currently, TCP carries more than 90% of today's Internet traffic and 80% of the total number of the flows in the Internet. Therefore, TCP becomes the natural choice for supporting reliable transmission over wireless networks. As multi-hop wireless technology further proliferates, it is expected that TCP will be used much more often over multi-hop wireless networks. Although TCP has shown an efficient performance in wired networks, its deployment in MWNs suffers from extreme performance degradation and poor bandwidth utilization. This is basically because the assumptions behind TCP design were initially inspired by the features of wired networks.

[0003] In fact, wireless networks are characterized by limited bandwidth, half duplex link, shared medium and random Bit Error Rate (BER). These characteristics render wireless networks to be substantially different from wired networks. Moreover, the resources available to wireless networks such as bandwidth, buffers and power are limited compared to wired networks. This problem is even more significant in MWNs as the absence of enough resources for the nodes not only affects their own performance, but also, the performance of other nodes due to forwarding property. Therefore, TCP protocol should be improved to optimise these resources efficiently.

[0004] It is well known that congestion control is one of the main mechanisms of TCP which controls the amount of packets sent into the network. Although this mechanism is tuned to perform well over traditional wired networks, the implicit assumption that any packet loss is due to congestion is not valid for multi-hop wireless networks. Unfortunately, wireless networks suffer from other types of losses that are not related to congestion, such as medium access contention complicated by the hidden and exposed terminal problems, shared path collision, route failures caused by node mobility, unpredictable channel errors, and limited wireless transmission range. When a packet is detected to be lost, TCP treats this loss as a sign of network congestion. As a result, it slows down its transmission rate or even experiences unnecessary timeouts. In addition, TCP is not properly adapted to the high variability of latency caused by wireless contention which often results in spurious timeouts.

[0005] Moreover, the congestion control mechanism of TCP cannot accurately adjust the sending rate to suit the transmission rates of wireless and this may cause either underutilized or overloaded bandwidth. In addition, the aggressive window increase policy of TCP itself is considered one of the critical reasons for poor TCP performance in multi-hop wireless networks. Due to the above factors, TCP needs to have some effective means to determine these events accurately so as to allow it to react appropriately.

[0006] Therefore there is a need for an invention within the framework which can overcome the above congestion problem.

Summary of Invention

[0007] According to an aspect of the present invention, the present invention provides a method for TCP congestion control in Multi-Hop Wireless Network comprising: sending packets by a sender (100); characterized in that observing RTT and IAT of TCP ACK value to estimate the current network conditions (200); conducting increase-decrease of cwnd of the rate of TCP (300); and conducting loss reaction mechanism whenever TCP detects the loss by duplicate ACK or timeout (400).

[0008] The above provision is advantageous as the present invention efficiently estimates the network condition, distinguishes between losses, and adaptively controls the data packet transmission is required.

Brief Description of Drawings

[0009] The present invention will be fully understood from the detailed description given herein below and the accompanying drawings as follows:

Fig.1

[0010] [Fig.1] is a flowchart of a method for TCP congestion control of the present invention.

Fig.2

[0011] [Fig.2] is a flowchart of the step of observing RTT and IAT of TCP ACK value to estimate the current network conditions (200) of the present invention.

Fig.3

[0012] [Fig.3] is a flowchart of the step of conducting increase-decrease of cwnd of the rate of TCP (300) of the present invention.

Fig.4

[0013] [Fig.4] is a flowchart of the step of conducting loss reaction mechanism whenever TCP detects the loss by duplicate ACK or timeout (400) of the present invention.

Description of Embodiments

[0014] As illustrated in FIG. 1, the present invention relates to a method for managing congestion for TCP in Multi-Hop Wireless Network comprising: sending packets by a sender (100); characterized in that observing RTT and IAT of TCP ACK value to estimate the current network conditions (200); conducting increase-decrease of cwnd of the rate of TCP (300); and conducting loss reaction mechanism whenever TCP detects the loss by duplicate ACK or timeout (400).

[0015] The step of observing RTT and IAT of TCP ACK value (200) as in FIG. 2 further comprising: determining current RTT once ACK returns to the sender (201); determining minimum RTT once ACK returns to the sender (203); determining IAT of ACK (205); and determining average time of receiving ACK in normal state (207).

[0016] The step of conducting increase-decrease of cwnd of the rate of TCP (300) further comprising: calculating RTT ratio on the receipt of an ACK (301); normalizing the RTT ratio on a linear scale of 0 and 1 (303); and mapping normalized RTT ratio to an increase-decrease action parameter or constant (305) as illustrated in FIG. 3.

[0017] The step of conducting loss reaction mechanism (400) further comprising: retransmitting corresponding segments and maintaining its cwnd (401); keeping its cwnd if IAT of duplicate ACK is less or equals the average time of receiving ACK (403); decreasing the cwnd if duplicate ACK is lesser than the average time of receiving ACK (405); waiting for a second timeout to make a decision in window reduction whenever the event of timeout happens (407); and reducing the cwnd (409). The above provisions are illustrated in FIG. 4.

[0018] The following detailed description of embodiments provides methods that optimize TCP performance in multi-hop wireless network. The present invention can be considered as a new TCP setup and will be called here in as TCP with adaptive congestion control (TCP-ACC). Since TCP-ACC does not seek any explicit feedback about the current network state, it maintains the round trip time (RTT) and the inter-arrival times (IAT) of TCP ACK value to estimate the current network conditions. Learning network state helps TCP-ACC to take proactive measures of the congestion and contention levels in the network. Moreover, TCP-ACC takes into account the delays that ACK packets are subjected to, due to the delayed ACK mechanism at the receiver side. Unlike most TCP variants, which increase and decrease their rate in several phases in a deterministic manner, TCP-ACC has only one phase that adaptively increases and decreases the rate of TCP using a mechanism called Adaptive Increase Adaptive Decrease (AIAD). Moreover, TCP-ACC detects the loss by time-out or duplicate ACK as in most TCP variants, but with a new mechanism of loss

reaction. An extensive description of TCP-ACC mechanisms is provided in the following subsections:

CAPTURING NETWORK CONDITION

[0019] To avoid the reliance on any explicit feedback about the network conditions from either the underlying layers or intermediate nodes on the path, TCP-ACC infers network conditions by observing both RTT and IAT values. The main components of this part include estimation of $RTT, minRTT$, IAT, in order to compute , , and mapping these values to parameters used to define actions to be performed.

RTT Samples Collection

[0020] Note that RTT is mainly used to compute the retransmission timeout value in the legacy TCP. However, TCP-ACC utilises the RTT values to learn the trends in the changing network conditions, that is, whether the network is free from congestion or getting congested. The key observation is that a long RTT generally indicates congestion. Hence, TCP-ACC uses RTT values as it allows accurate estimation of the network state which results in flexible and finer updates in the .

[0021] An RTT sample is measured as the time difference between the point where a packet is transmitted and the point where the associated ACK returns to the sender. If the incoming ACK belongs to the same transmitted packet, RTT value will be measured or it will be ignored otherwise. Since some TCP receivers may not generate ACK for every data packet received (delayed ACK), RTT value then will be measured as the average of number of packets that is covered by one ACK. This collection process operates fairly well under normal situations. However, a packet loss may inflate an RTT sample and consequently affect the collection accuracy. Thus, TCP-ACC ignore the RTT sample when a packet loss occurs. Equation (1) expresses the procedure of collecting RTT samples in TCP-ACC.

[0022] [Math. 1]

$$currentRTT = \frac{CurrentTime - SentTime}{Seqno - Lastacked} \quad (1)$$

[0023] The *CurrentTime* is the time of receiving new ACK, while *SentTime* is the time of sending the corresponding packets. The parameter *Seqno* represents the sequence number of the packet that the current ACK is acknowledged. Where as, the parameter *Lastacked* is the sequence number of the previous ACK. As several packets can be acknowledged by one ACK, the RTT is measured as an average of the total time taken by theses packets until their ACK is received over the number of these acknowledged packets.

Measuring Minimum RTT

[0024] For the first packet sent *minRTT* is equal to the RTT of that packet. Then for each ACK received, TCP-ACC updates *minRTT* by the *currentRTT* < *minRTT* if . Measuring *minRTT* is given by Equation (2).

[0025] [Math. 2]

$$minRTT = \begin{cases} CurrentTime - SentTime & \text{if first packet} \\ currentRTT & \text{if } currentRTT < minRTT \\ minRTT & \text{if } currentRTT \geq minRTT. \end{cases} \quad (2)$$

Measuring *ackIAT* and *safeIAT*

[0026] For each ACK received, TCP-ACC measure the inter arrival time of ACK *ackIAT* using Equation (3).

[0027] [Math. 3]

$$ackIAT = \frac{CurrentTime - LastackedTime}{Seqno - Lastacked} \quad (3)$$

[0028] This value is used to compute *safeIAT* which is the average time of receiving ACKs in the normal state. The *safeIAT* is used to alleviate IAT variability and is expressed in Equation (4).

[0029] [Math. 4]

$$safeIAT = \alpha * ackIAT + (1 - \alpha) * safeIAT \quad (4)$$

[0030] A typical value used for α is 0.125, which gives a low weight to the current and high weight to the historical average of IATs represented by *safeIAT* . Lower α avoid an *safeIAT* being skewed by any spikes in the measured IATs.

ADAPTIVE INCREASE ADAPTIVE DECREASE

[0031] As RTT values allow accurate estimation of the network state, it is possible to determine the number of bytes by which the *cwnd* should be increased or decreased by learning the congestion level in the network. The working mechanism of AIAD is described as follows.

Calculating *RTTRatio*

[0032] On receipt of an ACK, TCP-ACC obtains the current level of congestion and contention from *currentRTT* and *minRTT* values into *RTTRatio* parameter. As shown in Equation (5), the *RTTRatio* is normalized on a linear scale of 0 and 1.

[0033] [Math. 5]

$$RTTRatio = \frac{minRTT}{currentRTT} \quad (5)$$

[0034] In summary, when RTT value is close or equal to *minRTT*, *RTTRatio* gets a value close to 1. RTTs that resulted in such values for *RTTRatio* is called short RTT. In contrast, the RTT values that are significantly higher than are called as long RTTs. In this case, *RTTRatio* gets a small value close to 0. Long RTT is usually caused by congestion or contention in the network.

Mapping *RTTRatio* to η

[0035] As the network conditions can be learned from the normalized network response parameter *RTTRatio*, TCP-ACC defines Equation (6) to map *RTTRatio* to an increase/decrease action parameter called η . When *RTTRatio* is at the maximum value (*RTTRatio* = 1), η gets a value close to 1.

[0036] [Math. 6]

$$\eta = 2 * RTTRatio - 1 \quad (6)$$

[0037] When *RTTRatio* is close to 0, η will get a value close to -1. The value of η will be used to increase or decrease the *cwnd*. In summary, the parameter reflects the number of bytes that TCP should increase or decrease its *cwnd* to accommodate the network condition. η results in a flexible and finer updates in the congestion window.

Action Selection

[0038] The mechanism of increasing and decreasing *cwnd* in TCP-ACC has two important features.

[0039] Finer and flexible update in congestion window: In TCP-ACC, the amount of increase or decrease in *cwnd* is not fixed as in TCP. Because of this, TCP-ACC overcomes the problems caused by the aggressive and deterministic nature of TCP in updating the congestion window. When required, it can increase its *cwnd* by more than one MSS and it can even decrease its *cwnd* by few bytes or even keep it untouched.

[0040] Dynamic action selection: In TCP-ACC, the action selection (i.e., increasing or decreasing *cwnd*) is dynamic rather than deterministic. That is, when incipient congestion is detected, TCP-ACC can decrease the congestion window even on receiving ACKs. This directly helps TCP-ACC to take proactive measures to counter the congestion and reduce the packet loss due to congestion. Using η and β , TCP-ACC selects an action which is essentially an amount of increment or decrement in the congestion window as in Equation (7). TCP-ACC maintains β and η for each session (or flow) separately.

[0041] [Math. 7]

$$cwnd = \max(\beta * cwnd + \eta, 1) \quad (7)$$

[0042] Like other TCP variants, TCP-ACC prevents buffer overflow at the receiver by dynamically adjusting the number of packets which are ready to send (*win*) according to the minimum of advertised window (*awnd*) and *cwnd*. It also considered the unacknowledged packets which are still in flight as shown in Equation (8).

[0043] [Math. 8]

$$win = \min(cwnd, awnd) + lastacked - seqno \quad (8)$$

LOSS AND RETRANSMISSION

[0044] In multi-hop wireless networks, the loss can be attributed not only to congestion but also to wireless link errors, frequent route failures, fierce medium contention, mobility and high BER. When a packet is detected to be lost, TCP slows down the sending rate by decreasing its *cwnd* considering the loss is caused by congestion. In contrast, congestive losses in TCP-ACC occur rarely as it uses a proactive mechanism to reduce its *cwnd* to avoid congestion as early as possible. In addition, TCP-ACC defines two other mechanisms to react to the losses when it occurs either due to time-out or duplicate ACK. When a loss is indicated, TCP-ACC retransmits the corresponding loss segment and maintains its *cwnd* to avoid such losses using the following mechanisms.

Duplicate ACK

[0045] Duplicate ACK (*duback*) is triggered by receiving out of order packets. Each *duback* contains the sequence number of the last packet received in-order. Therefore, it is not beneficial to measure RTT in this case and thus TCP-ACC sets η to zero. Instead, TCP-ACC uses IAT of TCP ACKs to learn the network state. As the duplicate ACK is generated immediately after an out of order packet is received, the IAT of duplicate ACK (*dupackIAT*) can be measured using Equation (9).

[0046] [Math.9]

$$dupackIAT = CurrentTime - lastackedTime \quad (9)$$

[0047] In this case, TCP-ACC keeps its *cwnd* if *dupackIAT* is less or equal to the *safeIAT*, otherwise it will decrease the *cwnd* by computing β as in Equation (10).

[0048] [Math.10]

$$\beta = \begin{cases} 1 & dupackIAT \leq safeIAT \\ \frac{safeIAT}{dupackIAT} & dupackIAT > safeIAT. \end{cases} \quad (10)$$

Timeout

[0049] Time-out occurs if an ACK is not received within a certain time RTO. In MWNs, ACK can be lost due to other reason not related to congestion. In this case, TCP will experience timeout and retransmit not only the missing packet, but also all the packet in the same window and invoke slow start by reducing its *cwnd* to 1.

[0050] To avoid such cases, TCP-ACC uses its AIAD mechanism to avoid congestion and when the loss occurs, TCP-ACC considers it as a random loss. Thus, it does not blindly reduce its *cwnd* for the first timeout. As TCP-ACC is not able to use RTT or IAT due to the absence of ACK, it waits for a second timeout to make a decision on window reduction. TCP-ACC use Equation (11) to measure β .

[0051] [Math.11]

$$\beta = \frac{1}{noTimeouts} \quad (11)$$

[0052] The value of β will reduce the *cwnd* by a ratio of the number of consecutive time-outs (noTimeouts). However, this mechanism is used as a provisional procedure since the proactive mechanism of TCP-ACC avoids such timeout as early as possible.

TCP-ACC as a whole

[0053] In the normal state the value of η and β will be close to 1. In this case the *cwnd* will increase rapidly. If a congestion is sensed, the amount of that congestion will be reflected in the η parameter. The *cwnd* will be decreased based on the amount of the congestion unless the *cwnd* reaches its minimum value of 1. TCP-ACC also uses the inter arrival time of ACK to measure the *safeIAT* in the normal state. This value will be used to measure the network state at the occurrence of loss due to duplicate ACK. In this case TCP-ACC can distinguish between random and congestive losses. If a congestion is sensed, the *cwnd* will be reduced using the parameter of β .

[0054] In the loss state, the lost packet is predicted using the timeout or duplicate ACK mechanisms of TCP. TCP-ACC reacts to these events by setting the value of β according to the network state and loss type. If the network state is sensed uncongested, TCP-ACC keeps its $cwnd$ unchanged. Otherwise, it reduces the $cwnd$ by computing β using Equation (10) in the case of duplicate ACK and Equation (11) if the loss is due to timeout.

[0055] The invention has been described in a preferred form only and many variations may be made in the invention which will still be comprised within its spirit. The invention is not limited to the details cited above. The components herein described may be replaced by its technical equivalence and yet the invention can be performed. The structure thus conceived is susceptible of numerous modifications and variations, all the details may furthermore be replaced with elements having technical equivalence. In practice the materials and dimensions may be any according to the requirements, which will still be comprised within its true spirit.

Claims

- [Claim 1] A method for TCP congestion control in Multi-Hop Wireless Network comprising: sending packets by a sender (100);
characterized in that;
observing RTT and IAT of TCP ACK value to estimate the current network conditions (200);
conducting increase-decrease of *cwnd* of the rate of TCP (300); and
conducting loss reaction mechanism whenever TCP detects the loss by duplicate ACK or timeout (400).
- [Claim 2] A method for TCP congestion control in Multi-Hop Wireless Network as claimed in Claim 1, wherein the step of observing RTT and IAT of TCP ACK value (200) further comprising:
determining *current RTT* once ACK returns to the sender (201);
determining *minimum RTT* once ACK returns to the sender (203);
determining IAT of ACK (205); and
determining average time of receiving ACK in normal state (207).
- [Claim 3] A method for TCP congestion control in Multi-Hop Wireless Network as claimed in Claim 1, wherein the step of conducting increase-decrease of *cwnd* of the rate of TCP (300) further comprising:
calculating RTT ratio on the receipt of an ACK (301);
normalizing the RTT ratio on a linear scale of 0 and 1 (303); and
mapping normalized RTT ratio to an increase-decrease action parameter or constant (305).
- [Claim 4] A method for TCP congestion control in Multi-Hop Wireless Network as claimed in Claim 1, wherein the step of conducting loss reaction mechanism (400) further comprising:
retransmitting corresponding segments and maintaining its *cwnd* (401); keeping its *cwnd* if IAT of duplicate ACK is less or equals the average time of receiving ACK (403);

decreasing the cwnd if duplicate ACK is lesser than the average time of receiving ACK (405); waiting for a second timeout to make a decision in window reduction whenever the event of timeout happens (407); and reducing the cwnd (409).

[Fig. 1]

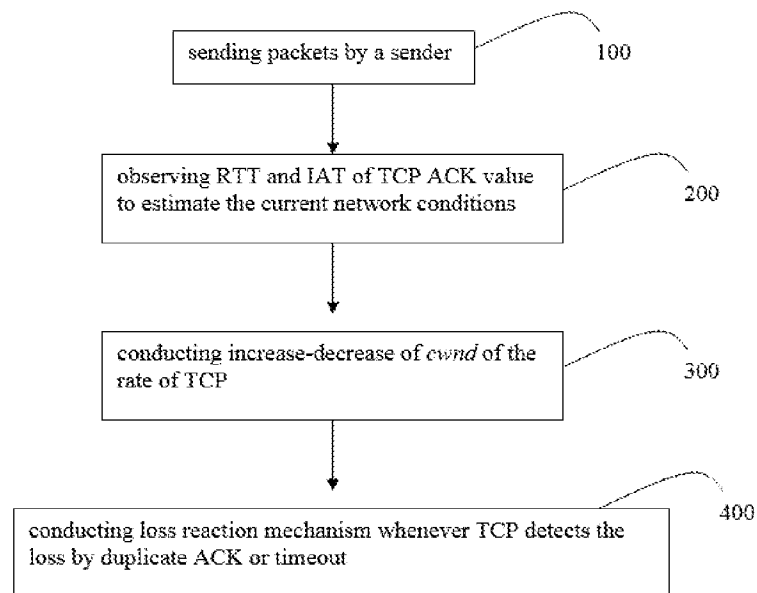


Fig.1

[Fig. 2]

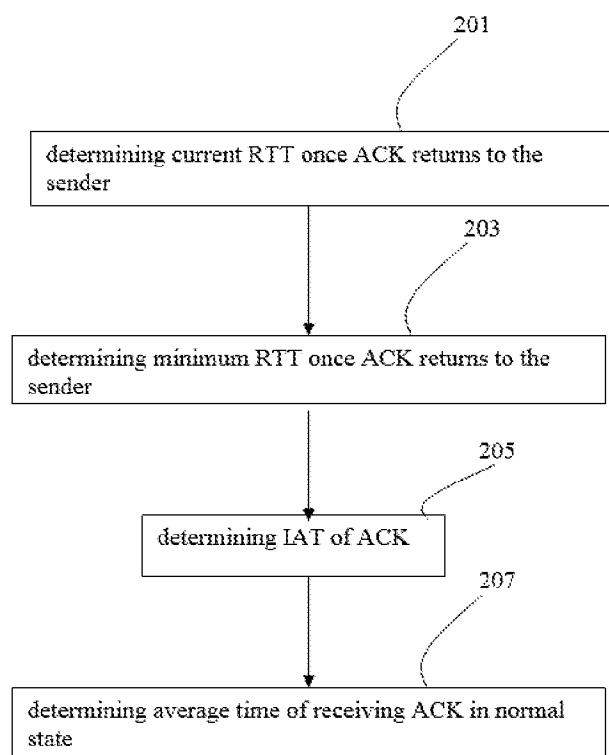


Fig.2

[Fig.3]

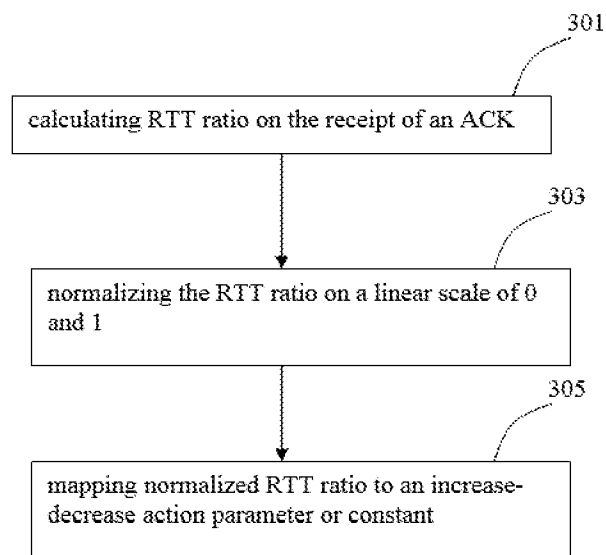


Fig.3

[Fig.4]

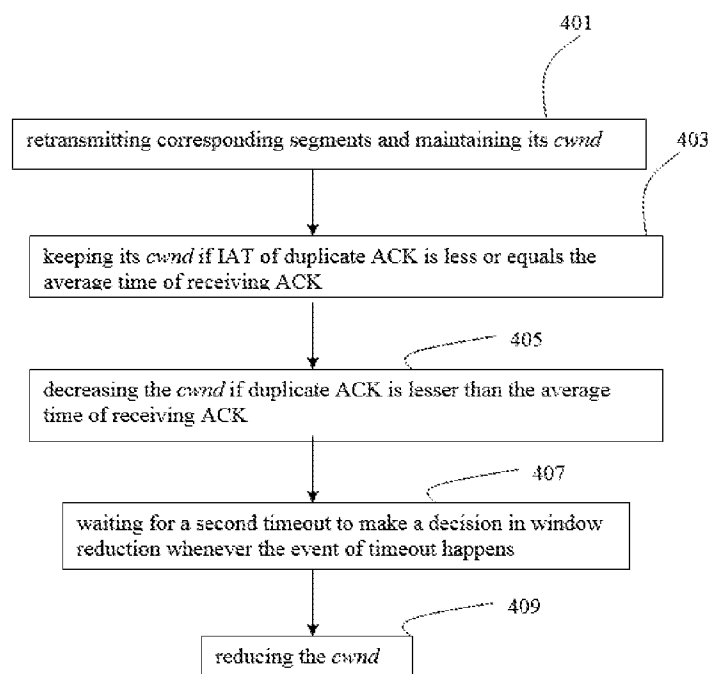


Fig.4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/MY2015/050119

A. CLASSIFICATION OF SUBJECT MATTER			
Int.Cl. H04L12/807 (2013.01) i			
According to International Patent Classification (IPC) or to both national classification and IPC			
B. FIELDS SEARCHED			
Minimum documentation searched (classification system followed by classification symbols)			
Int.Cl. H04L12/807			
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched			
Published examined utility model applications of Japan 1922-1996 Published unexamined utility model applications of Japan 1971-2016 Registered utility model specifications of Japan 1996-2016 Published registered utility model applications of Japan 1994-2016			
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)			
IEEE Xplore CiNii			
C. DOCUMENTS CONSIDERED TO BE RELEVANT			
Category*	Citation of document, with indication, where appropriate, of the relevant passages		Relevant to claim No.
A	ICHIMURA, Yuuki et al., Improving a Congestion Control Method extending TCP-Vegas over Wireless Multi-hop Ad-Hoc Networks, IEICE Technical Report, NS2010-207, 2011.02.24, Vol.110 No.448, pp.253-256		1-4
A	YAMAMOTO, Ryo et al., Flow Control Method Using Round Trip Time on Ad Hoc Networks, IEICE Technical Report, NS2008-124, 2008.12.11, Vol.108 No.359, pp.95-100		1-4
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.			
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family			
Date of the actual completion of the international search		Date of mailing of the international search report	
12.01.2016		19.01.2016	
Name and mailing address of the ISA/JP		Authorized officer	5X 3048
Japan Patent Office		ANDO Kazumichi	
3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan		Telephone No. +81-3-3581-1101 Ext. 3596	