Received: June 4, 2017

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

TCP LR-Newreno Congestion Control for IEEE 802.15.4-based Network

Andri Sembiring[1] Maman Abdurohman[1*] Fazmah Arif Yulianto[1]

[1]*School of Computing, Telkom University, Indonesia*
abdurohman@telkomuniversity.ac.id

Abstract: This paper proposes a new method of TCP congestion control for the IEEE 802.15.4 standard. This method, which evolved from TCP Newreno, is to be adopted on an IEEE 802.15.4 based network. The implementation of TCP Newreno on Wireless Sensor Network (WSN) is unable to classify losses of segments caused by congestion, wireless channel error or segment mishandling. An increase in TCP Newreno aggressive behavior on congestion avoidance phase has also led to frequent occurrence of congestion due to the low rate or limited bandwidth of IEEE 802.15.4. The proposed method is named LR-Newreno. LR indicates Low Rate - WPAN network specifications for WSN based on IEEE 802.15.4 standard. We propose enhancement on window growth function (addictive increase) and window decrease function (multiplicative decrease) in compliance with WSN characteristics. The new method has been simulated on network simulator-2 environment. The result shows that the TCP LR-Newreno provides better performance in term of several parameters such as throughput, data drop rate and energy consumption when compared to original TCP Newreno.

Keywords: LR-newreno, Transmission control protocol (TCP), Wireless sensor network (WSN), IEEE 802.15.4, WPAN.

1. Introduction

WSN is generally made out of a substantial amount of sensor devices equipped with a radio communication system, and used to observe environmental condition or object condition such as room temperature, level of air pollution, noise, magnitude of vibration, movement of objects, etc., without requiring immediate presence of human beings. WSN commonly incorporates some strong constraints regarding resources such as: memory, energy, computational processing speed and data or transmission rate.

Many applications or implementations of WSN require the availability of network protocols that can make sensor equipment or sensor nodes communicate directly with the external network. This can be done by applying TCP/IP on WSN as is done in the Internet; also, TCP can be used to perform remote management and re-programming of the sensor node [1]. There are several considerations for performing TCP implementation on wireless sensor network, such as:

a. TCP is the most reliable transport protocol used in IP-based networks.
b. TCP ensures reliability of data transmission from a sensor to a host external IP and vice versa.
c. With TCP, we can open SSH connection to log into wireless devices (sensors, actors, etc.) and execute commands.

TCP is a reliable transport protocol. When applied on wireless networks, however, it will cause some problems in terms of throughput and efficiency of energy consumption. Therefore, to increase its performance, it is deemed necessary to improve several functions and mechanisms contained in the TCP algorithm according to the characteristics of WSN.

Using congestion control over WSN is also very important, because the overload wireless network caused by simultaneous data transmission can increase the likelihood of data collisions. These

collisions affect data losses and thereby data re-transmission is required, which is certainly inefficient for WSN in term of energy consumption [2]. TCP is also unable to differentiate between a lost segment due to congestion and a lost segment due to bit errors. Whenever a segment is lost, i.e. when no acknowledgment is received for that particular segment and a timeout event is triggered, this loss is believed to be due to a congestion in the network. Consequently, the sending rate is reduced to avoid further segment losses [3].

In summary, we find that there are several issues which need to be solved before TCP becomes a viable protocol for use in a WSN, such us: header overhead, packets fragmentation and reassembly; the greatest hurdle which hinders TCP from being widely adopted in wireless sensor networks is TCP's flow and congestion control mechanism [4].

This paper focuses on investigating the TCP congestion control implementation in IEEE802.15.4 based WSN. We proposes enhancement for improving TCP performance in compliance with WSN characteristics by changing window growth function (addictive increase) and window decrease function (multiplicative decrease) of existing TCP congestion control mechanism. In other words, we propose another TCP variant which is used especially for IEEE802.15.4 based WSN.

This paper consists of five sections. Setion one explains about backgound and problem of this paper. Section II discusses the related works and literature review that relevant to this paper. Proposed method is presented in section III and section IV provides an analysis of the experiments outcome. Finally, Section V concludes and future work of this paper.

2. Related Work

There have been some work that already proposed solutions to the problems of TCP for implementation to a certain network or a specific type of network. It is indicated by the presence of some types of TCP known as TCP Variant, such as: Bic TCP/Cubic, TCP Vegas, Compound TCP, TCP Westwood, TCP Hybla, TCP HS etc. [5]. Most of the existing TCP variants were developed following the trends in the network technology which is showed by the existence of wider bandwidth. The main goal for developing TCP, therefore, is to be faster in utilizing or maximizing throughput of the available bandwidth. Yet, this is contrary to what happens in WSN, where available bandwidth is quite limited and the main objective of the WSN is the efficiency in network communication rather than getting larger throughput. This condition has made most of the existing TCP variants not suitable for use with WSN.

In addition to TCP, UDP is also fairly popular for use on the Internet. However, UDP also has several problems when implemented on WSN, such as [6]:

- User Datagram Protocol (UDP) has no congestion and flow control mechanisms, and therefore in the case of congestion, the possible amount of data lost can be much larger, which consequently requires more data re-transmission activities; this will be very inefficient for the implementation on WSN.
- There are no acknowledgment mechanisms in UDP, thereby no guarantee for the reliability of the data transmission.

According to [2], in determining or designing a transport protocol to be adopted in WSN, there are some important factors that must be considered. First, it must be able to reduce the dropping of data, and dropping of data means a waste of energy. Second, there must be a flow and congestion control mechanism, so that the reliability of data transmission is ensured. Third, the transport protocol should be able to maintain interoperability and fairness for a variety of sensor nodes available.

Both transport protocols, TCP and UDP, have problems when used in WSN [4][6], but according to the above consideration factors for WSN, we suggest that TCP is more appropriate to be used in WSN. Thus, in this research, we will focus on examining and trying to propose a solution for the implementation of TCP in WSN.

Several specific works that focus on solving the problem of adoption of TCP in WSN have also been proposed, some of which have given significant improvement and contribution. One of them is TCP header compression which is proposed to address header overhead problem of TCP adoption in WSN [3]. Another is TCP MSS tuning which contributes to the determination of the appropriate packet size of TCP for more efficient implementation in WSN [7]. There are several other works identified [2][8][9], but most of them are still focusing on the adjustment or modification of TCP protocol stack (header, packet size, code size, option, etc.), and we have not found any works proposing any new TCP congestion control variant designed according to WSN characteristics and especially used in WSN.

2.1 TCP congestion control

The TCP is a connection-oriented transport protocol that operates on Transport layer TCP/IP. It handles end-to-end data transmission [4]. Window

based control mechanism is used to control data flow. Window is data size of amount of unacknowledged data [10]. Tahoe TCP congestion control, in 1988, was introduced that consist of three congestion control schemes: congestion avoidance, fast retransmit and slow start [11]. Reno/Newreno TCP congestion control, in 1990, was introduced that used addition algorithm called fast recovery [11]. The enhancement of TCP congestion control is continuing in line with technological development.

2.2 TCP for wireless sensor network

Wireless Sensor Networks are becoming ubiquitous not only in the scientific world but also more and more in your home (e.g. monitoring the temperature of rooms, alerting of an upcoming rain shower, etc.). Thus the communication between sensor networks and other networks is getting more important too.

The de-facto networking standard protocol suite is TCP (with IP). That why TCP can and should be adapted to sensor networks. While it is possible to run TCP on sensor nodes [2] it is not feasible (in terms of energy consumption) to run pure TCP on them. In this thesis approaches to tailor TCP to the characteristics of a WSN are presented.

Other than TCP, UDP are also well-known transport protocol in Internet. But UDP also is not the good choice for WSN for some of the following reason [6]:

a. In UDP there is no mechanism for flow and congestion control. While UDP is used in WSN, there will cause lots of datagram drop when congestion happens.
b. There is no ACK mechanism in UDP, no any reliability mechanism.

Both transport protocols, TCP and UDP have problems if used in WSN [4] [6], but according to the above consideration factors for WSN, we suggest that TCP is more appropriate to be used in WSN. So in this thesis, we will just focus on examine and try to propose solution for implementation of TCP in WSN.

There are several issues which need to be solved, before TCP is a viable protocol combination to be used in a WSN. One issue is the header overhead [4]. TCP and IP headers combined have a minimal size of 40 bytes: 20 bytes TCP header plus 20 bytes IPv4 header, without any additional options. If we look at the maximum size of link layer frames a significant part of the message is being occupied by header data. The IEEE 802.15.4 standard [6] for example limits the maximum size of link layer frames to 127 bytes. That leaves a mere 87 bytes of TCP payload even without taking the link layer header into account.

The headers therefore occupy more than 30 % of the total maximum possible data which can be sent in one frame. The nodes scarce energy resources are thusly not utilized in an optimal way. Additionally it should be noted that larger payloads can be fragmented into many packets. Fragmentation and reassembly, however, are themselves energy consuming processes. This problem has been tried solved by [2] using TCP Header Compression and TCP MSS Tuning.

The greatest hurdle which hinders TCP/IP from being widely adopted in wireless sensor networks is TCP's flow and congestion control mechanism [4]. It is very important to implement congestion control in WSN. Too many data transmitted in the network will lead to collision. It will increase energy-inefficient [2]. TCP cannot differentiate between a lost segment due to congestion and a lost segment due to bit errors. Whenever a segment is lost, i.e. no acknowledgment is received for that particular segment and a timeout event is triggered, this loss is believed to be due to congestion in the network. Consequently the sending rate is reduced to avoid further segment losses [3].

While this might be a good strategy in wired networks it certainly is not appropriate for wireless sensor networks, where bit error rates are orders of magnitude higher (up to double digit percentage package error rates [4]).

Despite the fact that the loss occurred due to bit errors the sending rate is reduced nevertheless. This leads to a less than ideal throughput. In this study, we tried to solve this problem by doing improvement on TCP by creating a new TCP variant that is accordance with characteristic WSN.

3. The new method LR-newreno

The framework of the TCP Congestion Control mechanism is shown in Fig. 1. There are several main processes such as Retransmission Timeout, Congestion Avoidance, Fast Recovery, Slow Start, and Fast Retransmit [12].

Congestion Avoidance and Fast Recovery Phase is the function mostly used by TCP to adjust congestion window size during data transmission; it is, therefore, needed to improve the function to achieve a more efficient and better performance for specific applications like WSN.

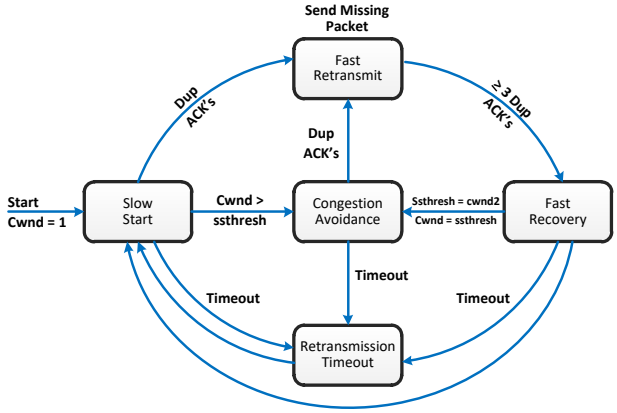


Figure.1 TCP Congestion Control Main Process [12]

3.1 TCP newreno

Existing congestion control mechanism of Newreno as illustrated in Fig. 2, also known as AIMD (Additive Increase Multiplicative Decrease), comprises four phases, namely slow start and congestion avoidance phase (part of AI), fast re-transmit and fast recovery phase (part of MD).

In the slow-start phase, congestion window (CW) will increase by 1 for each ack packet received, so the size of cwnd will be 2 times per RTT. It means that the cwnd increases exponentially (2n) per RTT, while in the congestion avoidance phase, CW will increase 1/CW per ack. Thus, the size of the CW will increase by 1 per RTT. This mechanism is very useful on networks with a wide bandwidth, because the achievement of the maximum bandwidth can be done quickly. Yet, for WSN with a limited data rate (250 Kbps), what happens is a rapid occurrence of congestion. Frequent occurrence of congestion will increase the resources requirement. Therefore, it is necessary to conduct an improvement which could minimize the congestion, but, at the same time, the bandwidth utilization can still be maximized.

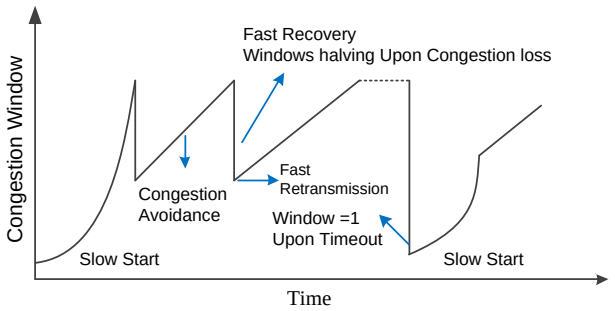


Figure.2 TCP newreno

TCP congestion control assumes that the packet losses occur as a result of congestion in the network, and responds to it with Multiplicative Decrease by decreasing the size of the cwnd to an half (cwnd = cwnd /2) or 50%. Whereas, in wireless networks, packet losses often take place due to the occurrence of the error channel (channel noise), which generally does not reach 50% and run in a very short time. Thus, this mechanism can drastically reduce TCP throughput performance in the WSN.

3.2 TCP vegas

To verify the results of the improvement, the proposed method is also compared with TCP Vegas. According to [13], TCP Vegas is more stable and better than some others TCP versions because of the congestion control before collision, but it has a drwaback on competitive ability and lack on fully take advantage of available bandwidth on transmitting packets.

TCP Vegas perform a very different approach respect to TCP New Reno in handling TCP windows growth function during data transmission. TCP Vegas is detect the congestion using the RTT fluctuation of the packets rather than using occurred losses packet as it does with TCP Newreno.

According to [14], Vegas-like protocol could improve the availability of the system for supporting real time application.

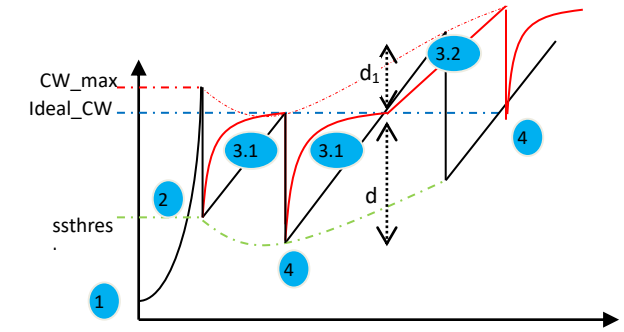
3.2 Design of proposed method

The New Method that is proposed in this paper as show in Fig. 3 that describes the concepts of proposed method (a) in detail and (b) in simple form.

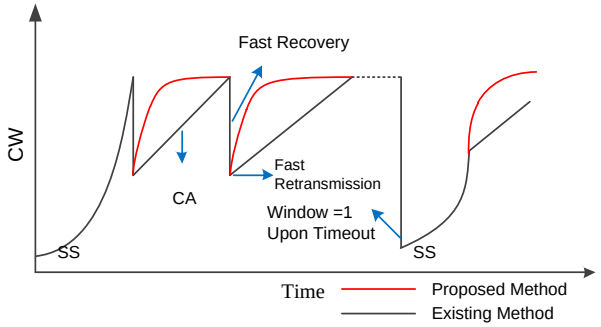
The LR-Newreno algorithm is designed as follows:

1) The main idea of this new method is how to identify and record the best CW achieved (CW_max) at each time before the occurrence of loss and make it a reference for the next phase.

2) Congestion Avoidance phase is divided into 2 sub phases. The first sub phase is when the CW increases rapidly approaching the last identified CW_max point and tries to maintain the position by slowing down the increasing speed of CW. The second sub phase is when the last CW max point can be passed smoothly, and CW will therefore increase moderately (faster with a constant increase) to utilize network bandwidth and identify new CW_max.



(a)



(b)

Figure.3 Proposed mehod: (a) details and (b) more simple

3) CW_max will be monitored and identified as "Ideal CW" if current CW has already left (d1 Range) the last CW_max point more than a half of the last distance range (d). (d1 > 1/2d).

The New proposed method has implemented as show in the algorithm pseudocode of each phase of TCP LR-Newreno. Every step or phase is indicated using number in circle.

Proposed Method Algorithm (pseudocode):

```
Phase 1:
CW =1
ssthresh = 20
Phase 2: SS (Exponential Growth)
If (rec Acks && CW < ssthresh)
CW = CW+1
Phase 3: CA (Additive Increase)
If (rec ACKs) {
   If (CW < CW_max) {  /* First sub phase */
        CW = CW + (CW_max-CW)/(α.CW_max)
} else { /* Second sub phase */
   CW = 1/(α.CW)
   Ideal_CW = CW_max
}}
Phase 4: Fast Retransmit/Fast Recovery
If (congestion) {
   If (rec 3 dupack or RTO) {
      CW_max = CW
      ssthresh = CW/2
      If (RTO) {
         CW = initial;
         Exit and go to SS;
      Else /* rec 3 dupack*/
         CW = ssthresh;
         If ((CW - Ideal_CW) > (β.ssthresh)) {
         CW = Ideal_CW
         }
      Exit and go to CA Phase;
      }
   }
}
```

ENHANCEMENT CODE

ENHANCEMENT CODE

Determination Value of α and β:

The value of α and β needs to be adjusted with the appropriate value to achieve a proper window growth function of the proposed method as illustrated in Fig. 3 (a) (red line), with the following steps:

1) The value of α, which is the decreasing coefficient of window growth function in the Congestion Avoidance phase, refers to the existing method. Because there is no coefficient of α, then the value of α can be assumed as 1; within this value, the window growth function of the existing method will not change.

2) The value α = 1, and then varies in a repetitive experiment while increasing the value by 1, which means higher than the value of α, will make the window growth function slower at the second sub phase of congestion avoidance, as visualized in Fig. 3 (a) (red line).

3) The value of β, which is a coefficient of ssthresh, the existing method without coefficient β, can be assumed as β = 1, which is also assumed as the highest value of ssthresh limit of the existing method. Thus, to get the best value of β, a repetitive experiment is conducted with varied values of β, 0 < β <1. After some repetitive experiments, the best value found was α = 4 and β = 1/2

4. Result and analysis

The study was accomplished by running simulations using tools called Network Simulator 2 (NS-2). The simulations were aimed to compare the performance of the existing methods and the proposed methods by conducting experiments in several scenarios with certain parameter values and performance metrics, as follows:

4.1 Simulation parameter

The simulation parameters used in this research are summarized as shown in Table I. It consists of some parameters and it's value such as traffic generator, TCP Flows, Wireless link and channel error.

4.2 Simulation scenario

According to Fig. 4, in the first scenario, the existing method and the proposed method will be tested in a simple network with only one TCP flow and without the presence of channel noise/error; the entire network resources can therefore be utilized and a maximum performance of each method be measured. Afterwards, the performance between both methods will be compared.

Simulations are run by activating tcp sink at host node as a receiver, and at the sensor node, 1 tcp agent activated as sender and using ftp application to generate data. For the first experiment, the tcp agent used at the sender is the existing method (Newreno); after that, the data are transmitted from sensor node 1 to host node, conducted over 200s, and then the results are measured. The same experiment is then repeated, but by replacing the TCP Agent with the proposed method (LR-Newreno) and also TCP Vegas, and then the results are measured and compared.

Table 1. Simulation parameters

Parameter	Value
Traffic Generator	FTP (File Transfer Procotol)
TCP Flows	1 (Scenario 1 & 2), 4 (Scenario 3 &4)
Wireless Link	802.15.4
Channel Error	1 – 10%
Routing Protocol	DSDV
Number of Sink	1
Mobility Support	No
Traffic Direction	Node --> Sink --> Host
Packet Size	100 Bytes
Queue Type	Drop Tail
Queue Length	50

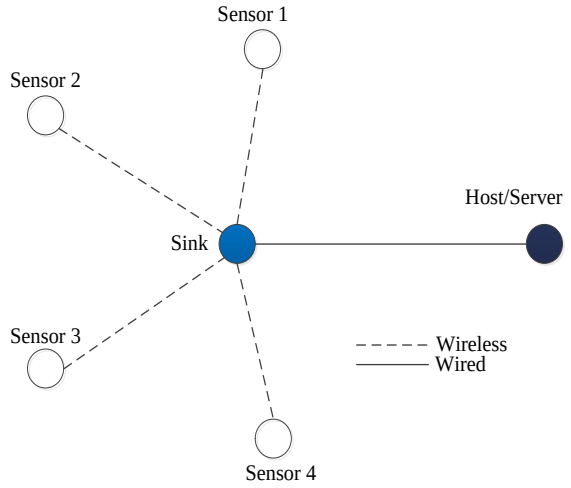


Figure.4 Scenario topology

The second scenario will run a test similar to the first one with a change of the parameter of channel error, randomly between 1 to 10%. This treatment aims to test and compare the response between the existing method and the proposed method as the effect of channel errors in the wireless networks, as commonly happen in an actual wireless network.

In the third scenario the existing method and the proposed method will be tested in a more complex network where the number of nodes propagated and tcp flow also multiplied up to 4 consecutive connection, in order to test and compare the performance between the existing methods and the proposed methods with a presence of bottlenecks and congestion due to shared network resources and without the existence of channel noise/error. Simulations are run by activating 4 tcp sinks at host node as a receiver, and at sensor node 1, 2, 3, and sensor node 4; tcp agent is activated as sender and uses ftp application to generate data. For the first experiment, the tcp agent used at sender is the existing method (newreno), and then it starts to transmit data from each sensor node to host consecutively; this is conducted over 100s, after which the results are measured. The same experiment is then repeated, but by replacing TCP Agent with the proposed method (LR-Newreno) and also TCP Vegas, and then the results are measured and compared.

The fourth scenario will do the same test as the third with changes in the parameter of channel error, randomly between 1 to 10%. This treatment aims to test and compare the response between the existing method and the proposed method as the effect of channel errors in the wireless networks, as usually happen in the actual wireless network.

4.3 Performance Metrics

1) Data Drop Ratio

is a percentage of lost data while on delivery from sender to receiver, also an indicator of congestion on the network.

$$Drop\_Rate = \frac{Number\_of\_Drop\_Packets}{Number\_of\_Received\_Packets} \times 100\ \% \quad (1)$$

2) Throughput

is an amount of data successfully transmitted from the source to the receiver at a certain time span. Throughput on this measure ignores all the overheads that arise on the network.

$$Throughput\_of\_a\_Node = \frac{Total\_Data\_Bits\_Received}{Simulation\_Runtime} \quad (2)$$

The throughput of the network:

$$Net_{Throughput} = \frac{Sum\_of\_Throughput\_of\_All\_Nodes\_Involved}{Number\_of\_Nodes} \quad (3)$$

3) Energy Consumtion

is a measure of the amount of energy spent by the node for data transmission activity. Measurements were made by giving the initial energy on each node, and will be reduced with a certain value for every data transmission and reception activity on the node.

$$Energy_{comsumption} = Initial\_energy - final\_energy \quad (4)$$

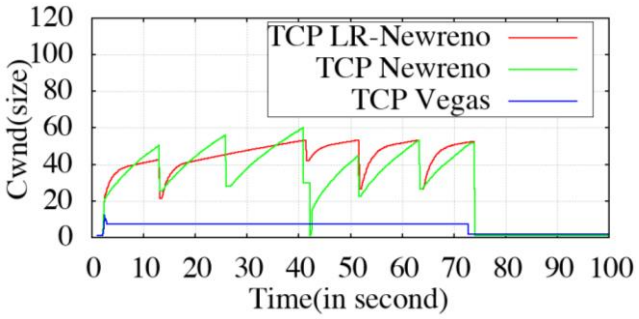
$$Average_{Energy_{Consumtion}} = \frac{Total\_Consumed\_Energy}{Number\_of\_Nodes} \quad (5)$$

A. Simulation Result and Analysis

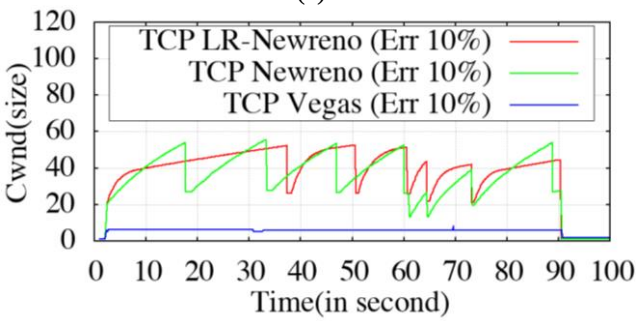
The existing method and the proposed method have been tested in all of the scenarios, and the results are then compared and analyzed regarding the response between the existing method and the proposed method as the effect of channel errors in the wireless networks, as commonly occur in the actual wireless network.

Simulation results as shown in the corresponding Fig. 5 (a) and (b) show the relationship between CW changes and simulation run time. The CW changes in the congestion avoidance and fast recovery phase clearly show a different reaction between the existing TCP Newreno and TCP LR-Newreno. Fig. 5 (b) shows that the presence of channel errors has resulted in CW to fluctuate more than that without channel errors in Fig. 5 (a).

Congestion avoidance phase of TCP Newreno (green line) indicates linear CW changes, while TCP LR-Newreno (red line) indicates arching CW changes. This means that at the beginning, the CW change will be faster and when approaching the previously known maximum CW, it becomes much slower. The aim is to maintain an ideal longer CW.



(a)



(b)

Figure.5. Congestion Window Comparison: (a) with and (b) without Channel Error

Fast recovery phase is the phase when the packet loss occurs with indications of reception of 3 duplicate acks at the receiver side and is visualized on the graph by the sudden changes of CW drops.

The comparison of TCP LR-Newreno and TCP Newreno are shown in Fig. 6. (a), (b) and (c). Fig. 6. (a) compare between Drop rate, (b) compare between throughput and (c) between energy consumption.

The main differences of fast recovery phase between TCP Newreno and LR-Newreno are shown at second 42, where TCP LR-Newreno just decreases the current CW to the previously identified ideal CW rather than doing windows halving (dividing the current CW by 2) as it does with TCP Newreno.

Sudden reduction of CW changes as visualization of packet loss occurrence in TCP LR-Newreno, which is less than that in TCP Newreno, shows that the modified TCP has a more efficient performance in term of lower drop rate and node energy consumption and also produce better throughput.

Figs. 6 (a), (b) and (c) show the proving of concept TCP LR-Newreno has better performance compare to previous protocol TCP Newreno.

TCP Vegas has better energy consumption compare to TCP Reno and LR-Newreno. Number of activity for sending and receiving routing protocol

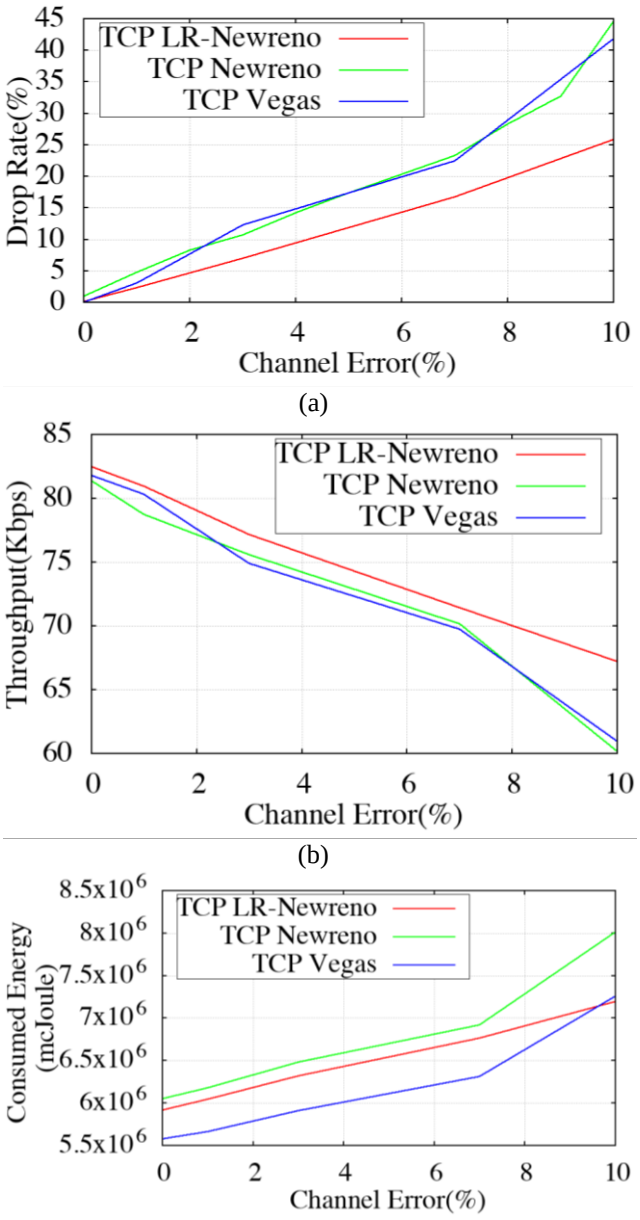


Figure.6 Comparison of TCP LR-Newreno, TCP Newreno and TCP Vegas on (a) drop rate, (b) throughput and (c) energy consumption

massages on the node is fewer than TCP-LR Newreno.

B. Summary of Findings

Enhancement of the existing algorithms shows us, through the simulation result of each scenario, that our suggestions to modify the existing TCP method in congestion avoidance and fast recovery phase improves its efficiency in term of drop rate and node consumed energy and also improves TCP performances on throughput. The simulation experiments conducted using star topology wireless sensor network with random channel error between 1 to 10%, both in simple network and more complex network, prove that our proposed method has outperformed the existing method almost in all aspects.

Table 3. shows that the problem of adopting TCP for WSN is not suitable for handling wireless channel error, but it is successfully improved based on our proposed hypothesis (scenario 2 & 4).

The result of experiments show the prove of concept of proposed method performance compare to existing method.

While the comparison of the proposed method with TCP Vegas, has not yielded satisfactory results. From Fig. 5 it can be seen that, TCP vegas has a very different congestion control approach. This figure shows TCP Vegas is more stable in the data transmission process, but on the other hand unable to increase Windows size agressiffly to increase bandwidth utilization available on the network.

Table 2. Summary of Simulation Result without Channel Noise/Error (LR-Newreno vs Newreno)

Scenario	TCP Scheme	Drop Rate (%)	Avg THR (Kbps)	Consumed Energy (mcJoule)
Scenario 1	TCP Newreno	1.49	81.43	4,304,211
	TCP LR-Newreno	0.31	82.71	4,178,861
Scenario 3	TCP Newreno	0.92	81.36	6,048,495
	TCP LR-Newreno	0.14	82,64	5,918,243

Table 3. Summary of Simulation Result with Channel Noise/Error (LR-Newreno vs Newreno)

Scenario	TCP Scheme	Drop Rate (%)	Avg THR (Kbps)	Consumed Energy (mcJoule)
Scenario 2	TCP Newreno	18.59	73.11	4,770,195
	TCP LR-Newreno	12.97	74.37	4,643,822
	Improvement	**5.62%**	**1.26 Kbps (↑1.8%)**	**126,373 mcJoule (Efficiency ↑2.65%)**
Scenario 4	TCP Newreno	20.21	71.17	6,897,275
	TCP LR-Newreno	12.94	74.20	6,579,965
	Improvement	**7.27%**	**3.20 Kbps (↑4.5%)**	**317,310 mcJoule (Efficiency ↑4.6%)**

Table 4. Summary of Simulation Result without Channel Noise/Error (LR-Newreno vs Vegas)

Scenario	TCP Scheme	Drop Rate (%)	Avg THR (Kbps)	Consumed Energy (mcJoule)
Scenario 1	TCP Vegas	0.01	82.29	3,961,653
	TCP LR-Newreno	0.31	82.71	4,178,861
Scenario 3	TCP Vegas	0.05	81.81	5,579,015
	TCP LR-Newreno	0.14	82.64	5,918,243

Table 5. Summary of Simulation Result with Channel Noise/Error (LR-Newreno vs Vegas)

Scenario	TCP Scheme	Drop Rate (%)	Avg THR (Kbps)	Consumed Energy (mcJoule)
Scenario 2	TCP Vegas	16.58	73.20	☑4,195,676
	TCP LR-Newreno	☑12.97	☑74.37	4,643,822
Scenario 4	TCP Vegas	15.39	73.68	☑6,125,993
	TCP LR-Newreno	☑12.94	☑74.20	6,579,965

The results as shown in the summary of performance in Table 5 on the noisy/error-prone network, the proposed method is more competitive to the Drop Rate and Throughput categories, but on the other side of TCP Vegas is more competitive in terms of energy consumption.

This result is a bit anomalous, because according to our hypothesis, that the decrease of the number of error (drop rate), will reduce the activity of data retransmission so that sender node will be have more efficient energy consumption.

After investigating the trace file of simulation results, we found that energy consumption for TCP Vegas is more efficient because the amount of activity for sending and receiving routing protocol messages on the node is fewer than TCP-LR Newreno and TCP Newreno, although drop rate of the TCP LR-Newreno is better.

This is confirmed because for the measurement of energy consumption in this research is applied by reducing the available energy of the node for each send/received activity, including for the needs as above. The method used in TCP Vegas is not the main focus of this research, and will be followed up in further research.

5. Conclusion

WSN environment is unique network condition. Congestion control in this system is a must to fit to its characteristic. This paper has proved The New TCP LR-Newreno congestion control that suitable for WSN environment. The result of all scenarios show that the new LR-Newreno method could improve efficiency in term of drop rate, node energy consumption and throughput. These simulations have been accomplished using star topology wireless sensor network with random channel error between 1% to 10%, both in simple network and complex network. The result shows that the TCP LR-Newreno provides better performance in term of several parameters such as throughput, data drop rate and energy consumption when compared to the existing method of TCP Newreno for WSN environment.

TCP protocol is still in use and technologies are evolving, there is a potential for other problems to arise in future. There is also an interest to improve the performance of networks. One part of our future work is to make further investigation compare to other TCP variant like Vegas, also investigate the performance of TCP algorithms in different network set-ups, different combination of routing protocol, another queue management and adjusting simulation parameter as actual as “real world” networks.

Acknowledgments

Authors thank to Telkom University for financial support under program of Research and Community Service Department (PPM) on International Research Scheme. We also thank to School of Computing Faculty for research lab. support.

References

[1] M. Anwander, G. Wagenknecht, and T. Braun, “Management of Wireless Sensor Net- works using TCP/IP,” In: *Proc. of IWSNE'08*, 1-8, Santorini Island, Greece, 2008.

[2] T. Braun, T. Voigt, and A. Dunkels, “TCP support for sensor networks”, In : *Proc. of Fourth Annual Conference on Wireless on Demand Network Systems and Services*, Oberguyrgl, pp. 162-169, 2007.

[3] A. Khurshid, M. H. Kabir, and M.A.T. Prodhan, “An improved TCP congestion control algorithm for wireless networks”, In: *Proceedings of 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, BC, pp. 382-387, 2011.

[4] O. Gasser, "TCP/IP communication in a WSN", In : *Proc. of Sensor Nodes–Operation, Network and Application (SN)* 75, 2011.

[5] N. Parvez, A. Mahanti, and C. Williamson, "An Analytic Throughput Model for TCP NewReno", *IEEE/ACM Transactions on Networking*, Vol. 18, No. 2, pp. 448-461, April 2010.

[6] C. Wang, K. Sohraby, Y. Hu, B. Li, and W. Tang, "Issues Of Transport Control Protocols For Wireless Sensor Networks", In: *Proc. 2005 International Conference on Communications, Circuits and Systems*, Hongkong, 2005.

[7] A. Ayadi, P. Maille, and D. Ros, "TCP over Low-Power and Lossy Networks: Tuning the Segment Size to Minimize Energy Consumption", In: *Proc. of the 4th IFIP International Conference on New Technologies, Mobility and Security*, Paris, pp. 1-5, 2011.

[8] A. Dunkels, "Full TCP/IP for 8-bit architectures", In: *Proc. of the 1st international conference on Mobile systems, applications and services (MobiSys '03). ACM*, New York, NY, USA, pp.85-98, 2003.

[9] L. Li, Y. Li, Q. Chen, and N. Nie, "PTCP: Phase-Divided TCP Congestion Control Scheme in Wireless Sensor Networks", In: *Proc. of Mobile Ad-Hoc and Sensor Networks, Springer Berlin Heidelberg*, pp. 281-290, 2007.

[10] S. Floyd, and K. Fall, "Promoting the use of end-to-end congestion control in the Internet", *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, pp. 458-472, 1999.

[11] J. Antila, "TCP Performance Simulations Using Ns2", 2002.

[12] B. M. Sabbar, "Generation and simulation of new transmission control protocol (TCP) agent over network simulator 2 (NS-2) platforms", *Scientific Research and Essays* 9.10, pp. 452-457, 2014.

[13] Y. Luo, M. Yin, H. Jiang, and S. Ma, "An improved congestion avoidance control model for TCP Vegas based on Ad Hoc networks", In: *Proc. of the 26th Chinese Control and Decision Conference (2014 CCDC)*, Changsha, pp. 2310-2314, 2014.

[14] M. Massaro, C. Palazzi, and A. Bujari, "Exploiting TCP Vegas algorithm to improve real-time multimedia applications", In: *Proc. of the 12th Annual IEEE Consumer Communications and Networking Conference*, pp.316 - 321, 2015.