Byteless CTS for passive load balancing in 802.11 networks

S Suherman1*, R Jasman1, M Pinem1 and Gunawan2

1 Electrical Engineering Department, Universitas Sumatera Utara, Medan, Indonesia
2 Politeknik Negeri Media Kreatif, Indonesia

*Email: suherman@usu.ac.id

Abstract. Load balancing in 802.11 networks is aimed at distributing traffics among access points (APs) to avoid overloaded access. Existing works may be classified into passive and active algorithms depending on the Ethernet changes in the user side. Since Ethernet 802.11 has been installed in millions of computers, changes in these user premises are not preferred. Software intervention on manufactured access points should also be kept as minimum as possible. Therefore, this work proposes a passive load balancing algorithm that transmitting byteless clear to send (CTS) signal in response to a request to send (RTS) sent by unwanted users when AP experiences overloaded. This method causes excessive users will passively be encouraged to find an alternative AP. There is no Ethernet change on the user side. Minor improvement is required in AP software to determine whether overload or not, and to decide which node(s) receiving byteless CTS. Evaluations show that average throughput to all connected users decreases as unwanted users failed to transmit. However, the average throughput to prioritized users increases significantly.

1. Introduction

Handover is not only applicable for the cellular network, but also for the 802.11 wireless fidelity (WiFi) network. It enables user migration from one access point (AP) to other APs [1]. User selection on AP is influenced by many reasons such as network eligibility, network surroundings, power sources or signal quality. An AP can be overloaded if a number of its users exceeds the network capability. In such a case, the load balancing algorithm is required so that users can be distributed among several APs [2].

Load balancing can be performed either passively or actively. Passive load balancing requires no modification within AP. In contrast, active load balancing requires interaction between users and APs. Load balancing methods use cell breathing proposed by Bahl [3] and Wang [4] are examples of passive load balancing. Bahl [3] proposed cell breathing by decreasing transmitting power when the network is overloaded. Meanwhile, Wang [4] restricted power reduction only for a beacon signal so that progressing communication is not distracted. Load balancers propose by Hojiev [5] and Shafi [5] also categorized as passive load balancing, but an additional server is added. The work on passive load balancing was also improved by Velayos et al [7] and Josua [8].

Active load balancing requires modification in both AP and users. Sectorized cell breathing in [9] is an example of the active load balancer. Modifications in both nodes are necessary to apply the method. It may be applicable if the standard is modified.

This paper proposes a simple passive load balancer to enable fast load balancing without significant modification in both nodes. The proposed method makes use of the clear to send (CTS) in carrier sense multiple access (CSMA) protocol.
(CSMA/CA) is the heart of the 802.11 networks which define the use of RTS/CTS. Request to Send (RTS) frame is sent out to obtain the permission of sending data. The CTS is a reply frame to grant permission. Clear to Send (CTS) signal is part of Ethernet software (CSMA/CA) which is a reply of the client request (request to send, RTS), informing a client when and how many bytes to send. CTS frame consists of four fields, which are frame control, duration, a receiver address (RA), and frame check sequence (FCS) as shown in Figure 1.

![Figure 1. CTS Frame.](image)

The proposed method is referred to as “byteless CTS” where the duration field within CTS octets is filled by zeros when a client or a user is “kick out” from the network to reduce network load. Client(s) or user(s) may be selected randomly or based on priority. This paper uses a random method. A user who receives byteless CTS will experience failure transmission all the time that may force the user to migrate to other AP. By doing this, load balancing will be achieved easily.

2. Research method
The proposed method and the existing method (cell breathing) are evaluated by simulations. Simulations are written in Java, employing some files extracted from many sources, such as Pamvotis [10]. Figure 2 shows the simulation steps. Two APs with 2 Mbps maximum capacity are employed to receive a connection of 8 packets per second with a packet length of 16000 bits generated by some nodes.

![Figure 2. Research steps.](image)
3. Evaluation results

Figure 3 shows the delay comparisons between two methods for a number of users higher than 16 nodes. The proposed method experiences a higher delay. Delay distribution shows that the proposed method dominates higher packet delay in each number of nodes. However, as packet delay is lower than 70 ms, which means that the quality is still much acceptable and delay characteristic exerts no problem.

![Graph showing delay comparisons between existing and proposed methods](image1)

(a) Average delay

![Graph showing throughput comparisons between existing and proposed methods](image2)

(b) Average throughput

Figure 3. Network performance.

Figure 3 proofs that the proposed method is able to increase network performances up 17.62 kbps or about 1.09%. Although the increment is relatively small, the lost connection decreases down to almost 50% from 12.22 nodes on average to only about 8.84 nodes as depicted in Figure 4. It means the overall network performance increases.
4. Conclusions
This paper has reported the proposed “byteless CTS” for the cell breathing method in the 802.11 networks. The proposed method is able to increase throughput and reducing lost connection. The experiment shows that the method increases throughput successfully about 1.09% and avoid wireless lost connection to almost 50%. Although its delay is higher, the figure is still acceptable. Future work may deal with the load balancing application in a real device.

Acknowledgment
This work was supported by Direktorat Riset dan Pengabdian Masyarakat, Kemenristek DIKTI Indonesia, under DRPM project Penelitian Tesis Magister 2018-2019.

References
[1] Krishnanjali and Patankar. 2014. Techniques for Load Balancing in Wireless LAN’s. Int. Conference on Communication and Signal Processing, p1831–36.
[2] H Long, H Y Shen, M Guo, F Tang. 2013. Dynamic Load-Balanced Routing in OpenFlow-Enabled Networks. AINA Int. Conf., 290–97.
[3] Bahl P, Mohammad T H, Kamal J, Sayyed V M, Lili Q, Amin S. 2007. Cell Breathing in Wireless LANs: Algorithms and Evaluation. IEEE Trans. on Mobile Computing 6 (2), 164–77.
[4] Wang S-L, Huang J-H, Cheng X-Z, Chen B. 2014. Coverage Adjustment for Load Balancing with an AP Service Availability Guarantee in WLANs. Wireless Networks 20 (3), 475–91.
[5] Hojiev Q, D-S Kim. 2015. Dynamic Load Balancing Algorithm Based on Users Immigration in Wireless LANs. Journal of Advances in Computer Networks 3 (2).
[6] Shafi U, Zeeshan M, Iqbal N, Kalsoom N, Mumtaz R. 2018. An Optimal Distributed Algorithm for Best AP Selection and Load Balancing in WiFi. 15th Int. Conf. on Smart Cities: Improving Quality of Life Using ICT & IoT, 65–69.
[7] Velayos H, Aleo V, Karlsson G. 2004. Load Balancing in Overlapping Wireless LAN Cells. IEEE Int. Conf. on Communications, 3833–36. https://doi.org/10.1109/ICC.2004.1313270.
[8] Napitupulu C J, Sihombing P, Suherman S. 2018. Optimizing the 802.11 hotspot performances by using load and resource balancing method. IOP Materials Science and Engineering 420(1).
[9] Suherman S, Sandi A, Ali H. 2019. Clustered cell breathing for load balancing in mobile and wireless networks, ELTICOM.
[10] Vassis D, Zafeiris V. Pamvotis–IEEE 802.11 WLAN simulator. http://www.pamvotis.org.