IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁙ Impact Factor 8.102 ⁙ Peer-reviewed / Refereed journal ⁙ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12739

Comprehensive Assessment of the Effectiveness of the Dynamic Adaptive ARQ (DA-ARQ) Methodology for Packet Analysis

V. Gokul[1], Dr. M. Shanmugapriya[2]

Research Scholar, Department of Computer Science, Park's College (Autonomous), Tirupur, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Science (UG), Kongu Arts and Science College (Autonomous),

Erode, Tamil Nadu, India[2]

Abstract: Traditional Karn's techniques, along with other techniques, could render the system unreliable because they provide a feedback mechanism to identify and correct errors. Increased error rates and decreased data integrity could come from this approach. Traditional Karn's method and other algorithms aren't likely to be able to manage an enormous number of users or adapt to shifting network circumstances, which could limit their potential to scale. This may reduce their utility in extensive networks. Inadequate security measures in the traditional Karn's algorithm, along with additional algorithms, could render them vulnerable to unauthorized access and data breaches. Since traditional Karn's method and other algorithms employ a constant retransmission rate, these techniques may not be ideal for changing network circumstances and have limited efficiency. As a consequence of this, it could result in greater delay, slower data transfer, and higher latency. The Dynamic Adaptive ARQ (DA-ARQ) method, which is more effective than the traditional Karn's method because it employs a dynamic approach to change the retransmission rate based on the network conditions, is the primary focus of this proposed survey paper. Overall, the DA-ARQ algorithm is more efficient, dependable, and secure than conventional Karn's method and other algorithms because of its dynamic and adaptive nature.

Keywords: Traditional Karn's Approach, Dynamic Adaptive ARQ (DA-ARQ), Retransmission Rate, Increased Error Rates

I. INTRODUCTION

A typical, easy-to-understand approach for packet retransmission in computer networks is the traditional Karn's algorithm. It is used when a packet needs to be retransmitted after being lost or damaged during transmission in order to maintain the integrity of its contents. Since it was created in the initial stages of computer networking, when network conditions were relatively stable and the number of network users was small, the traditional Karn's algorithm is regarded as an old technology. In these initial networks, which were distinguished by low capacity and high latency, packet loss and retransmission were issues that needed to be addressed. Sophisticated computer networks, however, have made the weaknesses of the traditional Karn's approach apparent. The high bandwidth, low latency, and numerous users that characterize modern networks can cause congestion and unpredictable network situations. Due to its reliance on a set retransmission rate that isn't always network condition-optimized, the traditional Karn's method may not be efficient in certain circumstances. Since other techniques for calculating packet timeout analysis rely on predefined timeout values that might not be ideal for network conditions, they might be regarded as obsolete. These specified timeouts can be predicated on network assumptions that aren't valid in practice, which would result in ineffective retransmissions and increased delay. For instance, fixed timeout values are dependent on the network's RTT or the quantity of retransmissions in the case of the Fixed Timeout (FTO) method and the Exponential Backoff (EB) algorithm, respectively. The dynamic nature of network circumstances, including variations in congestion levels or packet loss rates, isn't taken into consideration by these methods. Following are some conventional techniques for doing packet timeout analysis:

Karn's Algorithm: Depending on the predicted network round-trip time (RTT), this approach utilizes a preset timeout setting. The data packet is retransmitted if an acknowledgement (ACK) is not received until the timeout period ends, which is specified at a preset value.

Fixed Timeout (FTO) Algorithm: This technique makes use of a preset timeout value dependent on the network's RTT.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023
DOI: 10.17148/IJARCCE.2023.12739

The timeout duration is fixed and not altered in response to changes in the network.

Exponential Backoff (EB) Algorithm: During every resend, this technique employs a fixed timeout number that is progressively incremented. If an ACK fails to arrive before the timeout expires, the timeout amount is first fixed and then doubled for the subsequent retransmission.

Jacobson/Karels Algorithm: The timeout value is calculated by this approach using a smoothed estimate of the RTT and the variance in the RTT. The smoothed RTT estimate and a multiple of the RTT deviance are added to get the timeout value.

Although these traditional approaches have been employed in the past, today's dynamic network settings may not be conducive to their effectiveness. Modern techniques utilise adjustable timeout settings that are changed according to network circumstances, such as dynamic adjustable ARQ (DA-ARQ).More adaptable and dynamic algorithms, such as the Dynamic Adaptive ARQ (DA-ARQ) algorithm, have been designed to solve these constraints. These algorithms' capacity to modify the retransmission rate in response to communication circumstances can result in gains in efficiency, latency reduction, and dependability.

II. LITERATURE SURVEY

A well-known, lightweight protocol for controlling IoT networks is the Constrained Application Protocol [1] (CoAP). In constrained-node networks, the transport layer it runs on has a significant influence on how effectively it operates. With UDP serving as the underlying transport protocol, CoAP was first created as a lightweight web transfer mechanism for machine-to-machine communication. Recent research highlights the advantages of using CoAP over TCP to address the growing need for business infrastructure support whenever UDP-based traffic is constrained or firmware upgrades are sent. In this study, we conduct an experimental evaluation of CoAP across TCP and UDP in lossy networks. Both flow completion time and protocol overhead are included in our measures. The testing environment simulates the 3GPP-standard network known as Narrowband-IoT. Periodic and burst traffic, in addition to connections with extended and short lifespans, are examined. The results reveal that while CoAP over UDP works well for connections with a limited lifespan, its TCP equivalent is more appropriate for connections with lengthy lifespans and burst traffic. The varying flow completion times caused by the various packet loss rates have an impact on the performance. For effective deployments in various settings, the trade-off between superior performance and reduced resource usage must be taken into account. Our main conclusions may help CoAP management grow more effective.

The most commonly employed transport protocol [2] is TCP (Transmission Control Protocol), which employs a variety of techniques to ensure reliable data delivery. TCP is a dependable transport mechanism for the transport layer of the TCP/IP paradigm. Because of the enormous expansion that the internet has experienced lately, it is getting harder and harder to ensure that the services are available to as many people as possible. This is how extensive research has been done to enhance TCP's functionality generally and the control of congestion specifically. Several congestion management strategies have been put forth to boost TCP's functionality. However, it continues to have poor [2] performance. Determining the benefits and limitations of the most powerful algorithms in this sense (Tahoe, Reno, New Reno, Vegas, Sack, and Fack) in crowded environments will be the goal of the study we attempt to conduct in this article, which will be based on the analysis of some metrics, including packet drop, latency, and throughput. So, using 24 distinct situations for each method, we chose the NS2 simulator for the simulation. According to the study, TCP Tahoe, Reno, New Reno, and Sack are loss-based and favor packet loss in order to ensure low latency. Although TCP Vegas is dependent on delays, it is advised for applications that demand dependable packet transport. The findings of this study will serve as the foundation for subsequent research on the creation of a robust algorithm that combines the benefits of the algorithms that were studied and has the ability to fairly possess bandwidth with aggressive algorithms like TCP Reno in order to ensure effective congestion control.

The classification of network traffic [3] is an increasingly relevant issue in the field of computer science. Knowing what kinds of applications might flow through a network is a crucial challenge for Internet service providers (ISPs). The initial stage in analyzing and classifying the various types of applications running via the network is traffic classification. Internet service providers or network administrators can control the general performance of a network using this strategy. Traditional techniques for classifying internet traffic include those that use ports, payloads, and machine learning. The machine learning (ML) approach is now the most commonly used methodology. It has received excellent, accurate findings and is utilised by numerous researchers. Utilising a network traffic capture tool, a real-time internet data set is created.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⋇ Impact Factor 8.102 ⋇ Peer-reviewed / Refereed journal ⋇ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12739

The feature extraction tool is then used to extract features from the captured traffic, and four machine learning classifiers—the Support Vector Machine, the C4.5 decision tree, the Naive Bays, and the Bayes Net classifiers—are then utilised. According to experimental data, C4.5 classifications perform quite well in terms of accuracy when compared to another classifier.

In the research community, the issue of TCP incast in data centres has received a lot of attention. TCP incast is a catastrophic capacity collapse that happens when several senders submit TCP information to a single aggregator at the same time. Researchers [4] discovered that TCP timeouts are the main source of the incast issue based on a number of trials. Specifically, if at least one of the final three segments of a window is missing, timeouts brought on by inadequate duplicate acknowledgments are unavoidable. To increase the efficiency of TCP in data centre networks, these kinds of timeouts ought to be eliminated. Although there have been some initiatives to decrease timeouts, the issue has not yet been fully resolved, particularly in the case of timeouts caused by inadequate duplicate acknowledgments. In this paper [4], we introduce TCP-EFR, an effective TCP fast retransmission technique that can decrease TCP timeouts because of the absence of duplicate acknowledgments that are brought on by the loss of packets from a window's tail in data centre networks. Employing the congestion signal method of DCTCP based on instantaneous queue size, TCP-EFR modifies the TCP fast retransmission and recovery mechanism. Additionally, TCP-EFR regulates the transmitting rate to prevent switch buffer overflow and lower packet loss. TCP-EFR significantly reduces timeouts brought on by inadequate duplicate acknowledgments and outperforms DCTCP, ICTCP, and TCP in regards to efficiency, precision, and equilibrium under a wide range of networking conditions, according to an analysis of simulations using qualnet 4.5 in single and numerous congestion topologies.

In recent times [5], wireless network technology has been widely used to link remote user terminals with a core network. Whenever data is exchanged among terminal users and network managers, QoS is a crucial component that must be handled carefully. In WiMax, QoS is precisely specified at the MAC layer; however, the IEEE 802.16 network design does not clearly describe the bandwidth allocation scheduling mechanism that defines QoS. In addition, a novel bandwidth allocation scheduling technique is suggested for the IEEE 802.16 WiMax protocol in order to enhance Quality of Service (QoS). This work reviews and compares several current algorithms and sheds light on numerous challenges in constructing these algorithms.

The Internet [6] has evolved into a dynamic network wherein users may access resources via wired and wireless technology. On the Internet, there are currently additional types of bandwidths, delays, and errors. Since TCP is a common denominator for many services, it is possible to eliminate the requirement for local application of solutions by altering TCP. Our main goal [6] is to accelerate TCP's sluggish start phase, which attempts to determine the proper congestion window size. There will be an analysis of various Slow Start algorithms, and a new variation will be suggested.

The fundamental method [7] of data preprocessing is described initially in this study in order to increase the precision of data preparation in Web log mining. A session identification technique that utilizes dynamic timeout is then described after a thorough analysis of the conventional session identification algorithm. The initial timeout for each page is calculated at the start of the algorithm based on the statistical results combined with the importance level of the page.

While identifying sessions, the timeout is periodically changed, and sessions from users are identified using the dynam ic timeout analysis. The method [7] suggested can achieve superior performance on session identification, according to comparison experiments.

III. METHODOLOGY

3.1 Traditional Karn's Algorithm

Considering the traditional Karn's Algorithm was developed for early network circumstances with high latency and little bandwidth, it has come to be out of date. The network was dependable at the time, and there were not numerous individuals using it. However, in modern intricate and dynamic network contexts, it has become more apparent than ever before what the traditional Karn Algorithm's limits are. High bandwidth, low latency, and an enormous user base are characteristics of modern networks that can cause congestion and unpredictable network situations. Due to its reliance on a constant retransmission rate that can't be network condition-optimized, the traditional Karn's Algorithm might not be efficient in certain circumstances. Additionally, Traditional Karn's Algorithm ignores receiver input such as ACKs and NACKs, which might reveal important details about the present condition of the network.

DOI: 10.17148/IJARCCE.2023.12739

Modern methods, such as dynamic adaptive ARQ (DA-ARQ), can boost reliability, decrease latency, and improve efficiency by using feedback mechanisms to change the retransmission rate depending on network conditions.

In general, Traditional Karn's Algorithm was an effective approach at the time; however, in today's complex and dynamic network situations, it is no longer viable. Modern networks are complicated, and newer techniques that are more flexible and dynamic, like DA-ARQ, are better equipped to manage this complexity and can increase network performance. Due to its reliance on a predetermined timeout value that is dependent on the expected round-trip time (RTT) of the network, the traditional Karn's Algorithm is regarded as being out of date in terms of calculating packet timeout analysis. The RTT is considered to be reasonably stable and does not fluctuate much over time, depending on the algorithm. The RTT, however, can fluctuate dramatically in today's dynamic network environments because of variations in network circumstances, including congestion or packet loss. Due to the traditional Karn's Algorithm's failure to modify the timeout value in response to these modifications in the network environment, inefficient retransmissions and higher latency can occur.

3.2 Fixed Timeout (FTO) Algorithm

Since it uses a constant timeout amount that is not updated based on network conditions, the Fixed Timeout (FTO) Algorithm is regarded as outdated in terms of calculating packet timeout analysis. The technique fails to account for changes in network circumstances, including congestion or packet loss, as it maintains that the network conditions remain essentially steady. The technique does not account for changes in network circumstances, including congestion or packet loss, as it maintains that network conditions are largely steady.

A fixed timeout number could not constitute an ideal approach in all network settings in current dynamic network environments, where the network conditions might change dramatically over time. An excessively low timeout setting might lead to pointless retransmissions and increased delays. Retransmissions can be delayed, and throughput may be reduced if the timeout setting is set too high.

3.3 Exponential Backoff (EB) Algorithm

The exponential backoff (EB) algorithm remains current, although it has many disadvantages. The method is employed to control network congestion and decrease the frequency of collisions. For each subsequent attempt at retransmission, the time gap between retransmissions has gradually increased. Although certain network protocols continue to employ the Exponential Backoff (EB) algorithm, it has several drawbacks in contemporary network contexts. More advanced algorithms, including Dynamic Adaptive ARQ (DA-ARQ), which take input from the receiver into consideration, are frequently used to control congestion. Depending on the network conditions, these algorithms can modify the retransmission rate, leading to more reliable transmissions, less delay, and more effective retransmissions. Although the Exponential Backoff (EB) Algorithm may be used to control packet retransmissions in a network, current network settings are unable to lend itself to its application.

It is used to control network congestion and lower the frequency of collisions by progressively lengthening the time gap among retransmissions for each succeeding retransmission attempt. The Exponential Backoff (EB) Algorithm implies that network circumstances are mostly stable and fails to account for variations in network conditions, including congestion or packet loss, which is an additional limitation of the algorithm for packet timeout analysis. A fixed timeout number could not constitute the best option in all network settings in today's dynamic network environments, where the network conditions might change dramatically over time. It is used to control network congestion and lower the frequency of collisions by progressively lengthening the time gap between retransmissions for each consecutive retransmission attempt.

3.4 Jacobson/Karels Algorithm:

The Jacobson/Karels Algorithm's key disadvantage is that it uses a set timeout number that isn't modified depending on the state of the network. The technique does not account for changes in network circumstances, including congestion or packet loss, as it believes that the network conditions remain essentially steady. The TCP/IP networks, wherein the Jacobson/Karels Algorithm was developed, have advanced significantly since the algorithm's conception. Modern packet timeout evaluation techniques must take into account the unique requirements and difficulties of newer network protocols, including QUIC and HTTP/3. The Jacobson-Karels Algorithm is a straightforward, easy-to-implement algorithm. It is a viable option for some network protocols since it doesn't call for a lot of processing power or intricate computations. Even though the Jacobson-Karels Algorithm might not be the best option in all network situations, it could still prove helpful in specific situations. For networks with little usage or steady network circumstances, for instance, it could be appropriate as a set timeout amount can be utilised without significantly affecting performance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12739

3.5 Comparative Analysis

Algorithm	***Advantages***	***Disadvantages***
Karn's Algorithm	- Simple and easy to implement. - Works well in stable network conditions. - Suitable for networks with limited bandwidth or high congestion.	- Karn's Algorithm does not take into account changes in network conditions and assumes that the network delay is consistent. - Karn's Algorithm may not be optimal for dynamic networks with varying network conditions and high packet loss rates. - Can lead to inefficient retransmissions and increased latency.
Fixed Timeout (FTO) Algorithm	- Simple and easy to implement. - The FTO algorithm works well in stable network conditions where the packet loss rate is low and the network delay is consistent.	- The FTO algorithm does not take into account changes in network conditions and assumes that the network delay is consistent. - The FTO algorithm may not be optimal for dynamic networks with varying network conditions and high packet loss rates. - The FTO algorithm can lead to inefficient retransmissions if the timeout value is not set correctly.
Exponential Backoff (EB) Algorithm	- The EB algorithm is dynamic and adaptive, adjusting the retransmission rate based on feedback from the receiver and the network conditions. - Reduces the impact of packet loss on network performance. - Can lead to efficient retransmissions and reduced latency.	- May lead to increased network traffic due to increased retransmission rate. - Can be complicated to implement. - May not be optimal for networks with high traffic or congestion.
Jacobson /Karels Algorithm	- Simple and easy to implement. - Works well in stable network conditions. - Historical significance.	- May lead to increased network traffic - Can be complicated to implement - May not be optimal for networks with high traffic or congestion
DA-ARQ algorithm	- Algorithm works for low and high error rates - DA-ARQ can provide a higher level of reliability and data integrity - DA-ARQ adapts to changing network conditions, such as varying error rates and bandwidth usage, and adjusts its parameters accordingly.	- DA-ARQ is a more complex algorithm compared to some other packet timeout analysis algorithms. - DA-ARQ can reduce latency by minimizing the time it takes to retransmit lost packets, it can also increase latency if the retransmission rate is too high or if the network conditions are poor.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁑ Impact Factor 8.102 ⁑ Peer-reviewed / Refereed journal ⁑ Vol. 12, Issue 7, July 2023

DOI: 10.17148/IJARCCE.2023.12739

IV. PROPOSED METHODOLOGY

4.1 DA-ARQ Algorithm

As DA-ARQ is adaptive and dynamic, it can alter the rate of retransmission in response to feedback from the receiver and the state of the network. This could result in more reliable retransmissions with improved efficiency and decreased delay. The timeout value and retransmission rate are adjusted by DA-ARQ using feedback techniques like ACKs and NACKs. This enables it to consider the receiver's response and adjust the retransmission rate in accordance with the state of the network. By cutting down on the amount of time it requires to retransmit lost packets, DA-ARQ can lower latency. This can result in a rapid retransmission of dropped packets by altering the retransmission rate depending on the network circumstances. By lowering the number of lost packets and reducing the effect of packet loss on network performance, DA-ARQ can improve reliability. This is accomplished by modifying the retransmission rate for the network environment, which can result in more reliable and effective retransmissions. By adjusting the retransmission rate based on network circumstances, DA-ARQ can improve network performance. This can enhance dependability, decrease latency, and lessen packet loss, which will improve the whole network's performance. A packet timeout analysis method called DA-ARQ (Dynamic Adaptive Automatic Repeat Request) is used to optimize the rate at which lost or damaged packets are retransmitted in a network. Given the feedback from the receiver and the state of the network, it modifies the timeout value and retransmission rate.

A system for communication called DA-ARQ (Delay-Aware Automatic Repeat Request) enhances packet delivery across unreliable networks by combining Automatic Repeat Request (ARQ) with Forward Error Correction (FEC) methods. Since it considers the network's delay restrictions and modifies the rate at which packets are retransmitted based on the expected delay, DA-ARQ is superior to regular ARQ protocols in terms of analyzing packet delivery. Retransmitting missed packets until the target node properly receives them is how typical ARQ systems work. However, this strategy, particularly in high-loss conditions, might result in unnecessary retransmissions and increased network latency. On the other hand, DA-ARQ uses FEC to pre-emptively rectify packet mistakes and reduce the requirement for retransmissions. In order to optimize packet delivery while reducing latency, it additionally modifies the retransmission rate based on the predicted network delay. Since it combines ARQ and FEC methods and modifies the retransmission rate depending on the predicted network delay, DA-ARQ is more effective at analyzing packet delivery and maximizing packet delivery while minimizing delay.

4.2 DA-ARQ Algorithm Advantages

- The DA-ARQ algorithm provides redundant data in addition to the original data, enabling packet recovery without the need to wait for a retransmission request.
- By retransmitting only, the lost packets instead of the entire sequence, it may provide a greater level of dependability and data integrity.
- Using advanced techniques such as forward error correction (FEC) and interleaving, it is capable of handling more complicated situations like packet duplication and reordering.
- These approaches also adapt to changing network conditions, like varying error rates and bandwidth usage, and change their parameters accordingly.
- Mainly, it optimizes the use of available resources by dynamically adjusting the transmission rate and modulation scheme.
- This results in better efficiency of the system and reduced energy consumption.

V. CONCLUSION

The DA-ARQ technique employs a dynamic technique to change the retransmission rate based on network conditions, making it more effective than the traditional Karn's and other algorithms. This minimizes the demand for retransmissions and boosts the system's overall effectiveness. By modifying the retransmission rate in accordance with the network circumstances, the DA-ARQ algorithm lowers latency. This improves the system's overall performance and reduces the amount of time it takes to transfer data. Since the DA-ARQ method employs a feedback mechanism to identify and fix errors, it is more trustworthy than the traditional Karn's approach. This lowers the number of errors and improves the system's overall dependability. The DA-ARQ method can handle a greater number of users and react to changing network circumstances, making it more scalable than the traditional Karn's algorithm. As a result, large-scale networks may employ it. The DA-ARQ algorithm offers improved security since it employs encryption and decryption methods to safeguard data while it is being transmitted. This preserves the privacy of the data and prevents unauthorized access. Overall, DA-ARQ is a robust packet timeout analysis technique that can improve network performance by reducing latency, boosting reliability, and enabling more effective retransmissions.

DOI: 10.17148/IJARCCE.2023.12739

REFERENCES

[1] D. Pop, E. Kirdan and M. -O. Pahl, "Performance Comparison of UDP and TCP for Different CoAP Load Profiles," NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 2023, pp. 1-6, doi: 10.1109/NOMS56928.2023.10154441.

[2] Bazi, Kaoutar&Bouchaib, Nassereddine. (2019). Comparative study of TCP congestion control algorithms. International Journal of Advanced Trends in Computer Science and Engineering. 8. 3560-3564. 10.30534/ijatcse/2019/137862019.

[3] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn and F. Abdessamia, "Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 2016, pp. 2451-2455, doi: 10.1109/CompComm.2016.7925139.

[4] Sreekumari, Prasanthi& Jung, Jae-il & Lee, Meejeong. (2016). A simple and efficient approach for reducing TCP timeouts due to lack of duplicate acknowledgments in data center networks. Cluster Computing. 19. 10.1007/s10586-016-0555-z.

[5] Kaur, Avinash & Singh, Harvinder & Sharma, Parveen. (2014). Bandwidth Allocation Scheduling Algorithms for IEEE 802. 16 WiMax Protocol to Improve QoS: A Survey. International Journal of Computer Applications. 98. 16-22. 10.5120/17227-7550.

[6] Petrov, Ivan. (2013). Advanced Slow Start TCP Algorithm.

[7] Xinhua, He &Qiong, Wang. (2011). Dynamic timeout-based a session identification algorithm. 10.1109/ICEICE.2011.5777587.

[8] Roy, Rajesh & Das, Sudipto& Ghosh, Anup & Mukherjee, Amitava. (2007). Modified TCP congestion control algorithm for throughput enhancement in wired-cum-wireless networks.

[9] Byung-Jae Kwak, Nah-Oak Song and L. E. Miller, "Performance analysis of exponential backoff," in IEEE/ACM Transactions on Networking, vol. 13, no. 2, pp. 343-355, April 2005, doi: 10.1109/TNET.2005.845533.

[10] Byung-Jae Kwak, Nah-Oak Song and L. E. Miller, "Analysis of the stability and performance of exponential backoff," 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003., New Orleans, LA, USA, 2003, pp. 1754-1759 vol.3, doi: 10.1109/WCNC.2003.1200652.

[11] Y. Mohamed, N. Fisal and A. Mohd, "Performance of TCP on Mobile IP network during handoffs," Student Conference on Research and Development, Shah Alam, Malaysia, 2002, pp. 390-393, doi: 10.1109/SCORED.2002.1033140.

[12] Karn, Phil & Partridge, Craig. (2001). Improving Round-Trip Time Estimates in Reliable Transport Protocols. ACM Transactions on Computer Systems. 9. 10.1145/55483.55484.

[13] V. Jacobson. 1988. Congestion avoidance and control. In Symposium proceedings on Communications architectures and protocols (SIGCOMM '88). Association for Computing Machinery, New York, NY, USA, 314–329. https://doi.org/10.1145/52324.52356