Journal of Harbin Engineering University
ISSN:1006-7043

Vol 45 No. 11
November 2024

End-to-End Performance Analysis and Scalability of Tablet Devices in Wireless Environments

Mr. Deepak Jayanth P M[1], Dr. Rukkumani V[2*], Dr. Sivasankari S A[3], Dr. Srinivasan K[4]

[1]Principal Software Engineer, Social Innovation, Dell EMC, Bagmane World Technology Center, Mahadevapura, Bengaluru – 560048, India.

[2*]Associate Professor, Electronics and Instrumentation Engineering, Sri Ramakrishna Engineering College,Vattamalaipalayam, NGGO Colony Post, Coimbatore –641022, India.

[2*]Email:profvrukkumani@gmail.com

[3]Assistant professor (SG), Vignan's Foundation for Science, Technology and Research, Chebrolu (MD), VadlamudI (PO), Guntur (Dist) Andhrapradesh .PIN 522213, India.

[4]Professor and Head, Department Of Electronics And Instrumentation,Sri Ramakrishna Engineering College,Coimbatore-641 020, India.

Abstract: The proliferation of wireless devices such as tablets and smart gadgets underscores the need for continuous advancements in wireless LAN (Local Area Network) technologies. Despite routers being rated at high throughputs like 54Mbps or 100Mbps, the actual observed throughput often falls far below these benchmarks, typically ranging between 20 to 30 Mbps. This limitation becomes particularly apparent in scenarios such as wireless classrooms, where the number of connected users adversely affects throughput. This study aims to identify the bottlenecks hindering the connection of large numbers of wireless users (often exceeding 100) to a single access point. Additionally, it explores potential strategies to achieve this target, considering enhancements at the data link, transport, and application layers. Through a comprehensive analysis, this paper delves into strategies for improving scalability in wireless networking environments..

Keywords: Wireless LAN, Networking performance, Access point, Radio Resource Management (RRM), Interference Mitigation

Introduction

effectiveness of networks, services, and applications, as it guides critical decisions regarding hardware selection, technology implementation, user capacity estimation, and forecasting future system demands. Neglecting proper capacity planning and performance analysis can result in suboptimal functionality and increased maintenance costs for infrastructure. While wireless technologies have made significant strides in terms of speed, scalability remains a challenge, particularly when compared to wired networks. This is primarily due to the shared channel nature of wireless connections, where each connection competes for bandwidth. Presently, many wireless routers operate on the 2.4GHz band, which offers only three non-overlapping channels. Although the 5GHz band provides 23 non-overlapping channels, widespread adoption may be hindered by cost considerations. Therefore, maximizing the efficiency of existing technologies becomes imperative in achieving the desired capacity for accommodating a large number of wirelessly connected users [2,3].

Efficient handling of packets across all layers of the network stack is essential for realizing the goal of supporting a large number of wirelessly connected users. This involves optimizing protocols and algorithms at the data link, transport, and application layers to minimize contention and maximize throughput. By improving the efficiency of packet processing and transmission, networks can better accommodate increasing user demands without compromising performance. Moreover, leveraging advanced techniques such as Quality of Service (QoS) management and traffic prioritization can further enhance the overall scalability and user experience in wireless environments.

While the transition to 5GHz networks holds promise for alleviating scalability challenges by offering more available channels, the immediate focus should be on maximizing the efficiency of existing technologies. This entails deploying strategies to mitigate channel contention, optimize resource utilization, and enhance packet delivery mechanisms. By addressing these issues, organizations can effectively scale their wireless networks to support the growing number of connected devices and users, ultimately driving improved performance and user satisfaction.

2. Literature Survey

The research article titled "End-to-End Performance Analysis and Scalability of Tablets" presents a comprehensive exploration of tablet device performance and scalability. It investigates various facets including hardware specifications, software optimizations, and user experience considerations to provide a holistic view[1]. The paper meticulously examines the impact of hardware components such as processors, memory, and storage on tablet performance, shedding light on their contributions to scalability. Moreover, it delves into software optimization strategies implemented within tablet operating systems, aiming to improve responsiveness and resource management [7].

Through the utilization of benchmarking methodologies and performance metrics, the paper conducts a comparative analysis across different tablet models or operating systems, offering valuable insights into their relative strengths and weaknesses [11]. Furthermore, it identifies scalability challenges inherent in tasks like multitasking and resource-intensive operations, while also proposing potential solutions and forecasting future trends in tablet technology. Overall, this research contributes significantly to understanding the nuances of tablet performance and scalability, serving as a valuable resource for users and manufacturers alike in optimizing tablet usage and design strategies [4].

3. Materials and Methods:

Figures 1.1 and 1.2 illustrate the scenarios under consideration for which the analysis is applicable. In these figures, 'A' represents a wireless access point (AP) connected to a server machine denoted by 'S' via an Ethernet cable. Additionally, 'C' indicates wireless clients, typically exceeding 50 in number, which are wirelessly connected to the access point. These figures depict the network configurations and setups being analyzed in the study, providing visual representations of the wireless access point, server, and numerous wireless clients connected to the network [6].

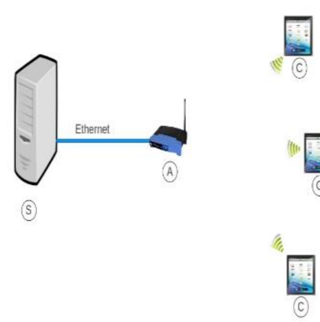

Figure 1.1: ***Scenario***

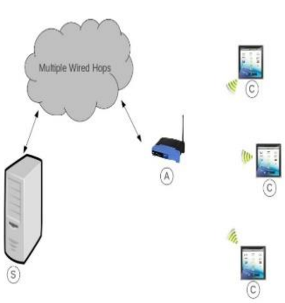

Figure 1.2: ***Scenario 2***

In the communication between a client and a server, potential bottlenecks can arise at various stages, each represented as a queue in the system. Initially, when a client sends a connection request, it arrives at the server's NIC (Network Interface Card) and is stored in a

buffer as an incoming request packet. This buffer may already contain similar request packets from other clients, leading to a bottleneck scenario if it becomes full [12,13]. Once the NIC's queue reaches its capacity, further requests cannot be processed, resulting in a potential bottleneck at this stage shown in fig.2.

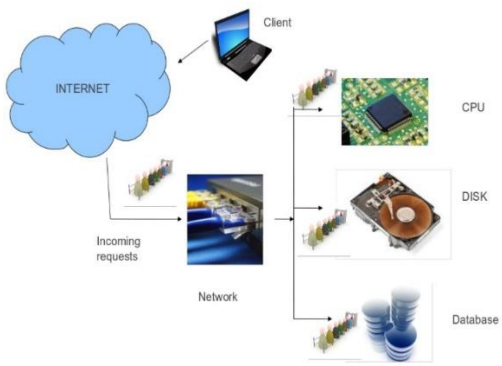

Fig 2:Potential sources of bottlenecks during a communication

If the packet successfully navigates through the NIC queue, it may encounter tasks that require access to resources such as the CPU, Database, or IO. Each of these resources has its own queue, mirroring the NIC queue but with different names. For instance, at the CPU level, bottlenecks may occur due to various scheduling methods for process/thread queues. Similarly, at the network level, the bottleneck arises from packets arriving at buffers in the network layer. Additionally, packets can be broken down and correlated with the buffer size of incoming frames at the data link layer, further contributing to potential bottlenecks [5].

Another perspective on bottleneck queues can be viewed from the application design level. For example, a protocol characterized by excessive messaging between peers, known as a "chatty protocol," can lead to inefficient bandwidth utilization due to overhead. In such cases, the percentage of bandwidth is wasted on the transmission of short messages, exacerbating bottleneck issues. Thus, addressing bottlenecks requires a holistic approach that considers various layers of the communication stack, from hardware resources to application design, to optimize performance and mitigate potential bottlenecks effectively [10].

4. Experiment Scenario

The experiment scenario does not include the assessment of bottlenecks caused by interference, as such effects cannot be accurately measured using wireless devices alone. To evaluate the impact of interference from neighboring Bluetooth devices, microwaves, or cordless phones on throughput, specialized equipment such as spectrum analyzers is necessary. These devices are essential for measuring and analyzing the spectrum, allowing researchers to quantify the extent of interference and its effect on wireless network performance accurately [8]. Therefore, the experiment focuses solely on factors that can be directly measured and analyzed using wireless devices, while acknowledging the importance of spectrum analyzers for comprehensive interference assessment in wireless environments.

- Three access points where installed on channel 1, channel 6 and channel 11 respectively.
- The whole capture runs for approximately 40mins.
- The wire shock capture was a merged one of two individual captures.
- During first capture, approximately 90 tablets where connected to 3 access points (30 users per A.P).
- The next capture, approximately 160 tablets where connected to 3 access points (50-60 users per A.P).
- The gap in the graphs approximately between 1100 to 1200 seconds indicates the stopping of wire shock between two captures.
- During first capture, 2 quiz where started and ended.

- During second capture 3 quizzes where started and ended.
- The capture was made using 3 USB wireless adapters working in monitor mode on 3 separate channels respectively.
- The tall peaks in the graphs represents quiz start or quiz end, where simultaneously all users try to connect.
- The graphs shown in the sections are from channel 1, for other channel graphs shown in fig.3

Quiz Status	Request Method	URL	Reply	Periodicity (second)
Before Quiz Begin	GET	QuizSSE.jsp	Quiz not started	Every 1 second
During Quiz Start	GET	Quiz.jsp	Quiz data (json)	Once
During Quiz Running	GET	QuizTimeUpdate.jsp	Seconds remaining	Every 1 second
During Quiz End	POST	QuizResult.jsp	Acknowledgement	Once
	GET	QuizResultSSE.jsp	Result XML	Once
	GET	QuizListener.jsp	Redirects to QuizSSE.jsp	Once
After Quiz End	GET	QuizSSE.jsp again	Quiz not started	Every 1 second

Fig 3:Quiz Client Server Communication model

4.1 Link layer Retransmission vs TCP layer Retransmission

The fig4 on page shows a typical TCP data connection from System 1(Server) to System 2 (Client) through an Access point. In the figure each transaction is marked by an arrow with a caption on top of it. The arrows also end with a number [14]. TheTCP connection shown in this figure is after connection establishment. The RTS/CTS mechanism are turned on automatically or by setting on the router. In the fig if L2(ack)[indicated by 5] does not reach the A.P, the A.P sends the packet no. 4 again to client.

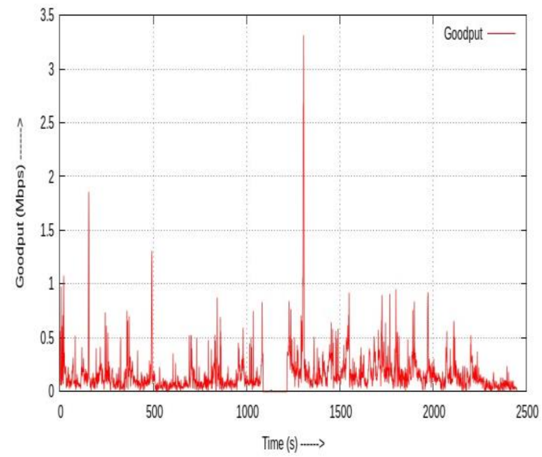

Fig 4: No of Distinct IP's communicating per second

This is called link level retransmission. In the same way if packet no. 9 is lost, packet no. 8 is send to access point again. This is also link level retransmission. If packet number 10 doesn't reach server then the server sends the data 1 again and follows the same procedure. This mechanism is called TCP retransmissions. The amount of time the device has to wait before TCP retransmission is calculated based on round trip time (RTT). The TCP layer retransmission is a well balanced mechanism which uses adaptive retransmission which calculates when to transmit a packet again, which is roughly twice as that of the time it waited for the first time retransmission.

In accordance with TCP adaptive retransmission specifications, a dynamically calculated timeout period governs each retransmission. This timeout duration is typically derived from the Round-Trip Time (RTT), a metric reflecting the time taken for a packet to travel from sender to receiver and back. As per protocol standards, if no acknowledgment is received within this dynamically calculated timeout period subsequent to sending a packet, the packet is retransmitted. Notably, the RTT, which forms the basis for calculating the timeout period, can fluctuate based on various factors including the number of network hops, prevailing network conditions, and other environmental variables. Therefore, the dynamic adjustment of timeout periods based on real-time RTT measurements ensures adaptive and responsive retransmission behavior in TCP communication, optimizing reliability and performance across diverse network scenarios.

The timeout value varies exponentially with every failed retransmission in case of TCP. In case of link layer retransmission, this exponential increase of the timeout value is not observed. This is well reflected in the graphs that the percentage change between the goodput and the datalink retransmission goes as high 500% as the number of wireless clients increase.

4.2 Goodput, TCP and Datalink overheads vs Time

This section shows how much of the bandwidth is wasted due to tcp retransmission, and link level retransmission. Three graphs are compared in this section. First is goodput, second is TCP adjustments and third is datalink adjustments. The following describes what the graphs denote. The fig here defines all TCP data excluding TCP retransmission and datalink retransmissions [15].Fig5 shows TCP data from Server to Client through A.P

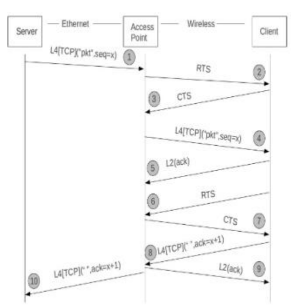

Fig 5:TCP data from Server to Client through A.P

4.3 Number of Network Transactions per second

The fig6 on page indicate how many transactions are running per second are observed. The transaction here quantifies any flow of communication which happen in the wireless channel. This overhead is constant across the time. The values of 200 to 400 indicates the continuous GET requests send by the clients to update the time on the tablets and the continuous poll for asking the start of quiz time. This overhead can be eliminated using web sockets, where the client need not poll the server continuously for data. It will automatically get the data from the server.

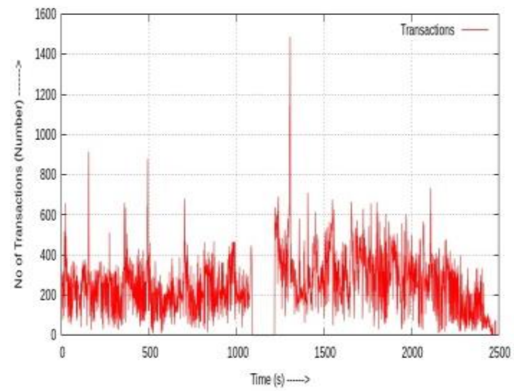

Fig 6:Potential sources of bottlenecks during a communication

4.4 Number of TCP Connections Established per second

In fig.7 on page refer the total no of TCP connection made throughout the capture process. As its noted from the graph there is always around 40 to 50 new connection established on the second capture. This is a overhead which can be minimized if keeping the connections open for a long time. This is possible with the help of long polling or better way using web sockets[9].

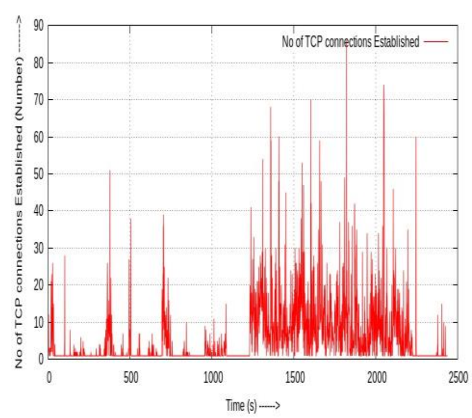

Fig 7:The number of TCP connection established

4.5 Analytical methods

Analytical methods for performance analysis adopt a bottom-up approach, beginning with modeling the functions of the hardware layer and progressing to higher layers such as Transport and Application. These methods provide a systematic means of analyzing performance, starting from the foundational hardware components and moving upwards. The analytical model discussed here focuses on quantifying the throughput achievable at the medium access level (MAC), which constitutes a sublayer of the Data Link layer. In essence, this model aims to provide an estimate of the throughput achievable specifically at the data link layer of wireless networks.

By employing analytical methods that begin with modeling hardware functionality and extend to higher layers, researchers can gain insights into the performance characteristics of various network components. Starting from the MAC layer, which plays a crucial role in managing access to the shared medium in wireless networks, this analytical model facilitates the estimation of throughput at the data link layer. Such analyses are essential for understanding the limitations and capabilities of wireless networks, informing the design and optimization of protocols, and ultimately improving overall network performance. Through these analytical approaches, researchers can systematically assess and optimize network performance, contributing to advancements in wireless communication technologies.

4.6 Software Methods

The software approach to performance analysis follows a top-down methodology, commencing with an analysis at the Application Level before delving into the intricacies of the hardware level. Unlike analytical methods, which may not capture every aspect comprehensively, software approaches leverage techniques such as profiling, logging, and simulation to approximate parameters or distributions. These methods prove invaluable when analytical modeling encounters challenges or when granular insights at the application level are imperative. While analytical models offer predictive insights into network capacity and user support, software methods step in when implementing analytical models becomes impractical.

Software methods shine in scenarios where modeling distributions of certain behaviors presents complexities, yet simulation or measurement remains feasible. By enabling the derivation of distributions that closely align with observed data, software methods enhance the accuracy and applicability of analytical models. These derived curves can then be integrated into analytical models to represent certain arrival or service rates, thereby enriching their predictive capabilities. Through the software approach, practitioners gain valuable insights into network performance, thereby facilitating informed decision-making and optimization efforts within intricate networking environments.

5. Experimental Results

5.1 Wireless Router Load Testing Tool

Wireless access points are tested in controlled environments with no presence of interference from blue tooth devices, cordless phones, microwaves and interference from other access points on the same channel. The throughput rated using this is not very useful in actual conditions. Because in actual sites where it is deployed, there will be interference from all the cases mentioned above. Hence the throughput measured vary from area to area. So there should be tool which measures some throughput qualities based on the particular site where it is deployed.

5.2 Cache and Compression

The other ways of minimizing data, is to minimize as much as possible in application layer. Caching and compression can be used whenever possible. A typical javascript jquery library comes around 250kB if uncompressed and 80kB if compressed. Application cache can be used on html5 compatible browsers. In order to use it, a manifest [14] file needs to be specified which says which all files need to be cached and the rest to not cache.

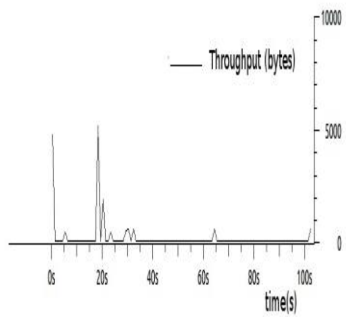

Fig 8:Cache and Compression technique

For example a typical web application, the java script libraries and other such libraries whose code wont change, can be in the cache and rest of network related authentication can be not cached.Fig.8 shows Cache and Compression technique

5.3 Traffic Shaping on 4users at 8mbps

The fig 9 on page is an excellent example where the traffic shaping worked perfectly as configured. Here the maximum rate of default queue which passes the traffic was set to8mbps. In between the experiment the traffic control was reset and again implemented. Here all four machines share the 8mbps equally of 2mbps each. When traffic control was off, they traffic of all four went to random speeds. When traffic control was back on, the four users transfer rate came back once again to the specified traffic rate specification of under 8mbps

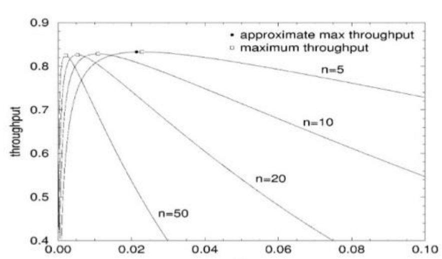

Fig 9:Traffic Shaping 4users at 2mbps

5.4 Traffic Shaping on 9users at 16mbps

The fig 10 shows the same set of rules as the above except the traffic was limited to 16mbps. Here the traffic after controlling where not smooth as the previous experiment. There are a lot of fluctuations. But still the maximum upper limit set by the traffic control was followed. The fluctuations can be attributed to the fact the classification of the packets itself may become a bottleneck. As the classification specification gets more and more into the packets, each packets at the queue needs to be examined to classify them. The traffic rules also needs to be simple as possible.

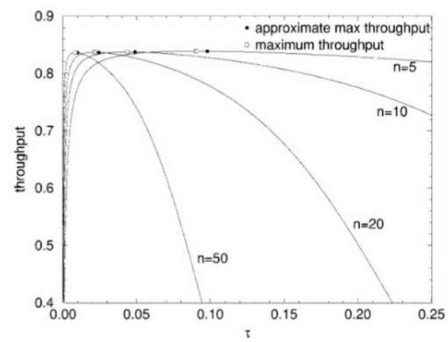

Fig 10:Traffic Shaping 9users at 8mbps

5.5 Proposed Solution for large number of users

The total available bandwidth varies based on the area where the access point is situated. The wireless load test tool will give an approximate upper bound to how many users, load the wireless access point can serve. This equals to the available bandwidth. After then a careful planning of per user bandwidth can be calculated. Traffic rules can be implemented based on this calculation, to guarantee bandwidth based on priorities. The traffic rules can be set such that the server machine has highest priority and higher available bandwidth. The clients will have comparatively lesser guaranteed bandwidth which they have to share with other clients.

The fig 10 on page describes the requirements of a scalable quiz model. With Web sockets, html5 cache this can be achieved. As seen from the table, the web sockets is held open always. The subsequent quiz can use the same connections to load data at just the overhead of 60 bytes of sending and 60 bytes for acknowledgement. Regular HTTP requests average header size is 300 to 400 bytes + additional 60 bytes of tcp header and the request HTTP header comes to 300 to 400 bytes + additional 60 bytes header. Total comes to 1000 bytes for every transaction. web sockets totally eliminates the http header overhead of 800 bytes every connection. The next problem solved is the TCP connection establishment overhead every time of (60 + 60 + 60 bytes) and connection termination overhead of (60 +60 +60 bytes).

The html5 cache saves images and java script libraries which need not be downloaded every time. In this setup as soon as the clients establish the connection the total transport layer overhead for that client is 0 bytes until the quiz starts. Since all the quiz data are server initiated, the server has entire control over how the data is send. It can send one by one or in groups thus decreasing the chance of collision by some amount. Collisions happen more when all clients at the same time wants to occupy the channel to send data. This doesn't mean collision is eliminated. At datalink level each machines sends keep alive signals now and then to the access point. But it significantly decreased the collision due to client machines querying http request and receiving http request at the same time. In this same setup, traffic shaping needs to be implemented in case of dropping any outside connections for file transfer, background data,etc are made from the devices.Table 1 shows Quiz Client Server Communication model.

Table 1: Quiz Client Server Communication model

Quiz Status	**Client Actions**	**Server Actions**	**Periodicity**	**No. of Bytes(Second)**
Before Quiz Begin	Get start html	Cache javascript libraries, images, css	Once	Len(jslibraries,css, images)+1000
Before Quiz Begin	Get Auth.html	Lad auth.js	Once	Len(auth.js)+1000
Before Quiz Begin	Open web socket	Keep connection open	Once	1000
Quiz 1 start	Ack	Send Quiz data	Once	Len(Quiz)+60+60
Quiz 1 Roaming				0
	Send answers	Ack	Once	Len(answers)+60+60
Quiz 1 End	Ack	Send Quiz 1 results(json)	Once	Len(results)+60+60
	Ack	Nextquiz-true	Once	Len(Quiz)+60+60
Quiz 2 Running	Ack	Send Quiz data	Once	0
	Send answers	Ack	Once	Len(answers)+60+60
Quiz 2 End	Ack	Send Quiz 2 results(json)	Once	Len(results)+60+60
	Ack	Nextquiz-false	Once	Len(Quiz)+60+60
After Quiz end	Close web socket	Ack	Once	60+60

6. Conclusions and Future Work

Scalability in wireless networks is usually achieved horizontally by deploying more access points. A single access point has its limits on the available bandwidth. The experiments and analysis suggest that, there is room for improvement in every levels of communication from datalink to application layer. The "wireless load test tool" and "laptop as access point" experiments suggest that a single device can play the role of client as well as access point. The amount of data transfer in both of the experiments are less than that of what normal access point and normal wireless client would transmit and still maintaining connectivity as it is supposed to. As the number of clients increases, the serving rate to each client should be decreased order to keep up the connections with all clients. Applications should be designed based on one or more of the experiments analysed, to definitely see improvements on the number of connected people.

Improving the wireless load testing tool to be robust to get more information based on its load to the Access point. Currently the graphs in the analysis shown can be obtained by giving a wire shock capture file as input to a shell script. It is made to automate completely. The shell script runs other sub scripts to finally produce all the graph images. One particular future work could make use of live graphs to represent the analysis. This can bemade as a plug in to wire shock. Building a Quiz application with complies with the improvements that can be done at all layers as suggested by this work.

Availability of supporting data: Data sharing does not apply to this article as no new data has been created or analyzed in this study.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Kim, J., Lee, H., and Choi, S." Scalability Enhancement Techniques for High-Density Wireless LANs." *In IEEE Communications Magazine,* 58.1 (2020): 112-118.

[2] Guérin, M., Nagaraja, K., and Woundy, R."Scaling Wi-Fi Networks for Large Crowded Venues. "*IEEE Internet Computing,* 24.5 (2020):70-77.

[3] Sen, S., Tatchikou, R., and Gerla, M." Power Control for Wireless LANs: A Measurement-Based Study." *IEEE/ACM Transactions on Networking,* 17.2 (2009):409-422.

[4] Zhang, X., Jin, H., and Zhang, X."Scalability of Wireless LANs Under Multi-Cell Environment."*IEEE Transactions on Mobile Computing,* 17.10 (2018):2324-2336.

[5] Chen, C., Hu, Y., and Wu, K." Scalability Enhancement in Wireless LANs Using Cooperative Transmission." *IEEE Transactions on Vehicular Technology,* 68.4 (2019):3893-3906.

[6] Kim, Y., Sharma, M., and Sridharan, A." Improving Scalability of Wireless LANs Through Dynamic Channel Assignment." *IEEE Transactions on Wireless Communications,* 16.5) (2017):3144-3156.

[7] Khademi, A., Mir, M., and Ghosh, A." Scalable WLAN Design for High-Density Environments. "*IEEE Communications Magazine* ,56.7 (2018):112-118.

[8] Song, L., Zhang, W., Wang, X."Enhancing Scalability of Wireless LANs with Multi-AP Cooperation." *IEEE Wireless Communications Letters,* 9.1 (2020):28-31.

[9] Liu, J., Liu, Y., and Yu, H." Scalable Wireless LANs with Beamforming Antennas." *IEEE Transactions on Wireless Communications,* 16.3 (2017):1874-1887.

[10] Shen, B., Wu, Y., and Li, J. " Scalable Wireless LAN Design for High-Density Deployments." *IEEE Transactions on Mobile Computing,* 18.9 (2019):2061-2075.

[11] Jin, S., Ma, L., and Qin, Z." Dynamic Load Balancing for Scalable Wireless LANs." *IEEE Transactions on Mobile Computing* ,17.6 (2018):1490-1504.

[12] Zhang, D., Li, S., and Wang. C." Enhancing Scalability of Wireless LANs with Multi-User MIMO." *IEEE Transactions on Communications* ,66.12 (2018):6168-6181.

[13] Zhang, Q., Zhang, W., and Li, S." Scalable WLAN Design with Coordinated Multi-Cell Transmission."*IEEE Transactions on Wireless Communications,* 19.2 (2020):1255-1269.

[14] Xu, L., Gao, Z., and Zhang, J." Enhancing Scalability of Wireless LANs with Device-to-Device Communication." *IEEE Transactions on Vehicular Technology,* 69.5 (2020):5134-5148.

[15] Zhou K., Yang Y., Chen Q."Scalable WLAN Design with Cognitive Radio Techniques." *IEEE Transactions*

on Cognitive Communications and Networking, 6.3 (2020):943-957.