IJIRIS: International Journal of Innovative Research in Information Security
Volume 11, Issue 10, November 2025
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2025.v1110.08

Efficient Scheduling Techniques for Performance Enhancement in Networks

Dr.Martin Joel Rathnam
Associate Professor/ Department of CSE(IC)
Sri Sairam College of Engineering, Bengaluru, Karnataka, India
martinrathinam.iot@sairamce.edu.in
https://orcid.org/0000-0001-9804-8894
Udipto Biswas,Lipun Swain,Lishanth Reddy C,Balaji Manivannan
Department of CSE(IC)
Sri Sairam College of Engineering, Bengaluru, India

Publication History
Manuscript Reference No: IJIRIS/RS/Vol.11/Issue10/NVISX10087
Research Article Open Access| Double-Blind Peer-Reviewed| Article ID: IJIRIS/RS/Vol.11/Issue10/NVISX10087 Received: 28, October 2025, Revised: 05, November 2025, Accepted: 12, November 2025, Published Online: 21, November 2025.
https://www.ijiris.com/volumes/Vol11/iss-10/08.NVISX10087.pdf
Citation: Dr.Martin,Udipto,Lipun,Lishanth,Balaji(2025),Efficient Scheduling Techniques for Performance Enhancement in Networks,IJIRIS: International Journal of Innovative Research in Information Security, Volume 11, Issue 10 of 2025 pages 676-684 **Doi:>** https://doi.org/10.26562/ijiris.2025.v1110.08
BibTeX Key: Dr.Martin@2025Efficient
IJIRIS papers should be cited as IJIRIS (International Journal of Innovative Research in Information Security, AM Publications, India 2025, ISSN 2349-7017, https://doi.org/10.26562/ijiris.2025.v1110.08 The journal's official abbreviation is IJIRIS. Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: In this work, a study on a novel Multiple- Input, Multiple-Output (MIMO) technique called Multiple Packet Transmission (MPT), with which the sender can send more than one packet to two distinct users simultaneously. Traditionally, in wireless networks, it is assumed that one device can send a data packet to only one receiving device at a given time. However, this restriction is no longer true if the sender has more than one antenna, by processing the data according to the channel state, the sender can send distinct packets to distinct users simultaneously. Problems related to MPT are seen and provided solutions. Here the problem of sending out buffered packets in minimum time is being formalized as finding a maximum matching algorithm. Since maximum matching algorithms are relatively complex and may not meet the speed of real-time applications, packet scheduling techniques from some algorithms like the Dijkstra's algorithm, the Linear Time ¾ Maximum Matching algorithm and Edmond's blossom algorithm are considered to enhance the downlink transmission, thereby improving the overall performance in wireless networks by using MPT. In this work the main concern is how to enhance the downlink performance in the wireless networks using the simultaneous behavior of multiple packet transmission. During the course of doing this paper, it becomes possible to learn Java functionalities by understanding the working of profiling java tools, features of wireless networks and implementing these algorithms.
Keywords: MPT-Multiple Packet Transmission, Packet transmission techniques, Downlink performance.

I. INTRODUCTION

Wireless access networks have been widely used in recent years when compared to the wired networks because wireless access networks are easier to install and use. Due to the tremendous practical interests, much research effort has been devoted to wireless access networks as great improvements have been achieved by adopting newer and faster signal processing techniques, for example, the data rate in 802.11 Wireless Local Area Network (WLAN) has increased from 1Mbps in the early version of 802.11b to 54 Mbps in the new 802.11a WLAN. It has been noted that in addition to increasing the point to point capacity, new signal processing techniques have made other novel transmission schemes possible, that greatly improve the performance in wireless networks. A wireless LAN is usually composed of an Access Point (AP), which is connected to the wired network and several users communicate with the AP through wireless channels. In wireless LANs, the most common type of traffic is the downlink traffic that is, the flow of packets from the AP to the users, when the users are browsing the Internet and downloading data. In today's wireless LANs, the AP can send one packet to one user at a time. However, if the AP has more than antenna and if MPT is used, then the AP can send two packets to two different users simultaneously whenever possible the throughput of the downlink increases. MPT is feasible for the downlink because it is not very difficult to equip the AP with two antennas in fact, many wireless routers today have two antennas. Another advantage of MPT that makes it very commercially appealing is that, although MPT needs a new hardware at the sender, it does not need any new hardware at the receiver. This means that to use MPT in a wireless LAN, the AP can be replaced and simple software protocols can be upgraded in the user devices without having to change their wireless cards therefore incurring minimum cost.

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2025.v1110.08

Go-Back-N ARQ protocol a specific instance of the automatic repeat request (ARQ) protocol is the existing system, in which the packet sending process continues to send a number of frames specified by a window size even without receiving an acknowledgement (ACK) packet from the receiver. It is a case a general sliding window protocol having the transmit window size of „N" and receive window size of "1" is implemented. In this protocol the receiver process keeps track of the sequence number of the next frame it expects to receive and sends that number with every ACK it sends. The receiver ignores any frame that does not have the exact sequence number it expects, whether that frame is a "past" duplicate of a frame it has already received, the ACK or whether that frame is a "future" frame waiting for an ACK. Once the sender has sent all of the frames in its window, it detects all frames, the first lost frame and will then go back to sequence number of the last ACK it received from the receiver process and fills its window starting with that frame and continues the process again. This algorithm was not efficient as it required more time for the procedure that only guarantees for an excellent reliability. Unlike waiting for an acknowledgement for each and every packet, the wireless connection would still be utilized as packets are being sent. However, this method results in sending frames at multiple times. If any frame was lost or damaged, or the ACK acknowledging them was lost or damaged, then that frame along with all the following frames in the window (even if they were received without error) will have to be re-sent again. Resulting in a low performance, that becomes a major issue in wireless network. The complexity of this protocol increases while choosing a Window size (N) for transmission. There are important issues that have to be kept in mind when choosing a value for "N", therefore restricting the efficiency of the wireless network.

1. The sender must not transmit too fast. "N" must and should be bounded by the receiver's ability to process packets.
2. "N" must be smaller than the number of sequence numbers (if they are numbered from zero to N) to verify transmission in cases of any packet (any data or ACK packet) being dropped.
3. Given the bounds presented in (1) and (2), "N" is to be the largest number possible. Taking into account all these restrictions present in the existing system in mind, a thirst for using a more effective and efficient scheduling technique for multiple packet transmission that successful dominates this existing system arises.

There are several algorithms that can be used in order to implement this paper but the Dijkstra's Algorithm finds its way deep into this work as it fits exactly to the likeliness of the existing system with some advanced algorithm and procedures**.** Dijkstra's algorithm solves the single source shortest path problem when all edges have non-negative weights. It is a greedy algorithm similar to Prim's algorithm. The algorithm starts at the source vertex "s", it then grows a tree "T", that ultimately spans all vertices reachable from 'S". Vertices are added to "T" in order of distance i.e., first "S", then the vertex closest to "S", then the next closest and so on.

II SYSTEM DESIGN

In order to implement the system there are several steps that must be followed like to create a design for the system, then analyzing the block diagram and arranging the flowchart in accordance with the algorithm. The DFD is called as a bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data and the output data is generated by the system as shown in figure 1.

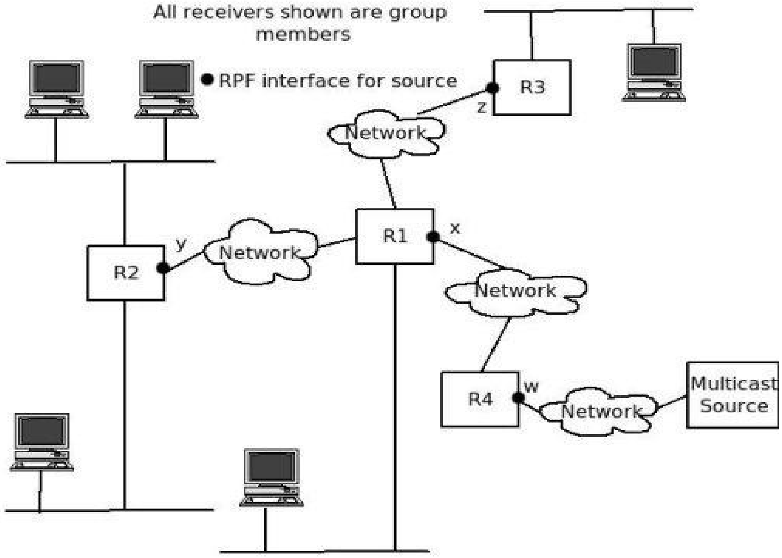

Fig 1: System Design

In this section of the work, an implementation on the design and performance of the wireless LAN after it was enhanced by MPT is discussed. The maximum arrival rate of the downlink is obtained first and then a study on the average packet delay by an analytical model and simulations is done. The performance of a wireless network depends on many factors, for example, the physical environment, the locations of the wireless nodes, such that the performance of one network could be different from that of another even when they are using the same devices.

In many cases, the performance of the same network may also be changing due to the occasional movements of the wireless nodes. This makes the performance evaluation in general, a difficult task. This algorithm solves the single source shortest-path problem when all the edges have non- negative weights. This algorithm starts with the source vertex "s", it grows a tree "T", that ultimately spans all vertices reachable from "S". Vertices are added to "T" in order of the distance. That is first "S", then the vertex closest to "S", then the next closest and so on. Dijkstra's algorithm runs in O(|E|lg|V|) time. Steps involved in the operation of the Dijkstra‟s algorithm indicate the working of this algorithm. The graph is made of vertices (nodes) and edges which link vertices together. Edges are directed and have an associated distance, sometimes called the weight or the cost. The distance between the vertex *u* and the vertex *v* is noted [*u*, *v*] and is always positive. The graph of nodes is shown in the figure 2.

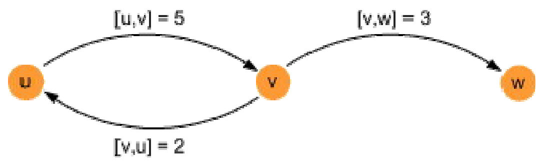

Fig 2 : Graph of nodes

Dijkstra's algorithm partitions vertices in two distinct sets, the set of „unsettled‟ vertices and the set of "settled" vertices. Initially all vertices are unsettled and the algorithm ends once all vertices are in the settled set. A vertex is considered settled and moved from the unsettled set to the settled set, once its shortest distance from the source has been found. The following data structures are used for this algorithm.

d Stores the best estimate of the shortest distance from the source to each vertex.

π Stores the predecessor of each vertex on the shortest path from the source.

S The set of settled vertices, the vertices whose shortest distances from the source have been found.

Consider an example shown in the figure 3, run Dijkstra's shortest path algorithm on the following graph, starting at the source vertex '*a*'.

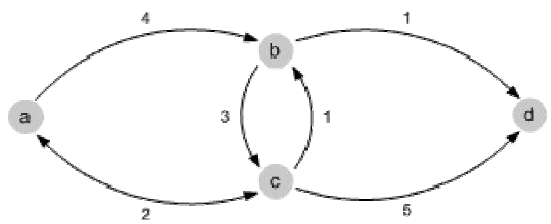

Fig 3: Graph of source vertex 'a'

Adding, a source vertex '*a*' to the set '*Q*'.
'*Q*' is not empty, extract its minimum '*a*' again add
'*a*' to '*S*', then relax its neighbours as shown in figure 4.

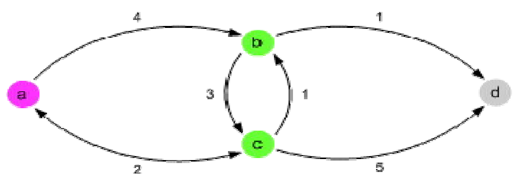

Fig 4: Relaxing the neighbours of 'a'

Neighbours along the vertices adjacent to '*a*', are '*b*' and '*c*' (in green above). First computing the best distance estimate from '*a*' to '*b*' by initializing '*d*(*b*)' to infinity.

d(b) = d(a) + [a , b] = 0 + 4 = 4............................. (4.1)

π(*b*) is set to '*a*' and adding '*b*' to '*Q*'. Similarly for '*c*', assign '*d*(*c*)' as '2' and '*π*(*c*)' to "*a*'. The second time around, '*Q*' contains '*b*' and '*c*', '*c*' is the vertex with the current shortest distance of '2' as seen in the figure 5. It is extracted from the queue and is added to '*S*' the set of settled nodes, then relax the neighbors of '*c*', which are '*b*', '*d*' and '*a*'.

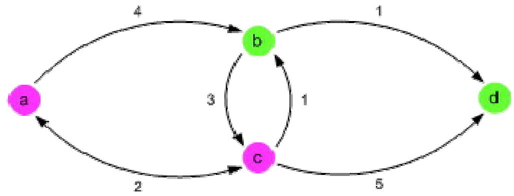

Fig 5: Relaxing neighbours of 'c'

Initially '*a*' is ignored because it is found in the settled set. But the first pass of the algorithm had concluded that the shortest path from '*a*' to '*b*' was direct. Looking at *c*'s neighbor '*b*',

d(b) = 4 > d(c) + [c , b] = 2 + 1 = 3...................(4.2)

After realizing, it was found that a shorter path going through „*c*' exists between '*a*' and '*b*'. So *d*(*b*) is updated to 3 and *π*(*b*) is updated to
c. *b* is added again to *Q*. The next adjacent vertex is *d*. *d*(*d*) is set to '7' and *π*(*d*)set to '*c*'. The unsettled vertex with the shortest distance is extracted from the queue, it is now '*b*'. Adding it to the settled set and relaxing its neighbors '*c*' and '*d*', skipping '*c*' that has already been settled. A shorter path is found for '*d*' as shown in the figure 6.

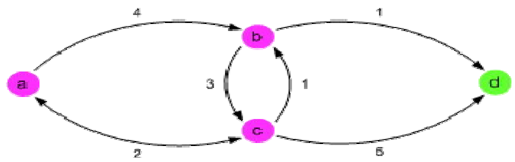

Figure 6: Shortest path between 'a' and 'b'

d(d) = 7 > d(b) + [b , d] = 3 + 1 = 4 .... (4.3)

Therefore update *d*(*d*) to 4 and *π*(*d*) to '*b*', then add '*d*' to the Q set. At this point the only vertex left in the unsettled set is d and all its neighbours are settled. The algorithm ends. The final results are displayed in red lines in figure 7.

- *π* - the shortest path, in predecessor fashion
- *d* - the shortest distance from the source for each vertex

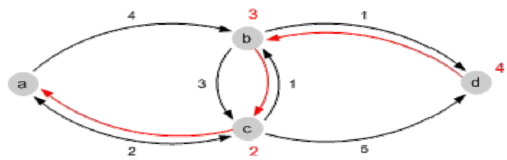

Figure 7 : Shortest distance found

Steps involved in the maximum matching operation of the Dijkstra"s algorithm are given.
Step 1: Given initial graph G = (V, E). All nodes have infinite cost except the source node s as shown in figure 8, which has 0 cost.
Step 2: First choose the node, which is closest to the source node s. Then initialize d[s] to 0. Add it to S, relax all nodes adjacent to source s. Then update the predecessor (see red arrow in figure 9) for all nodes updated.
Step 3: Choose the closest node x, relax all nodes adjacent to the node x, update predecessors for nodes u, v and y (again notice red arrows in figure 10)
Step 4: Now, node y is the closest node, so add it to S, then relax node v and adjust its predecessor (shown as red arrows in the figure 11).
Step 5: Now node u is the closest, choose this node and adjust its neighbor node v as shown in the figure 12.
Step 6: Finally, add node v. The predecessor list now defines the shortest path from each node to the source node s as shown in the figure 13.

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2025.v1110.08

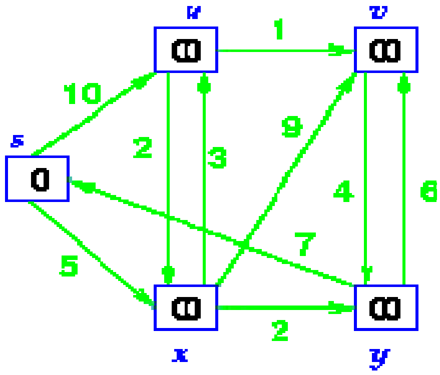

Fig 8: Node points

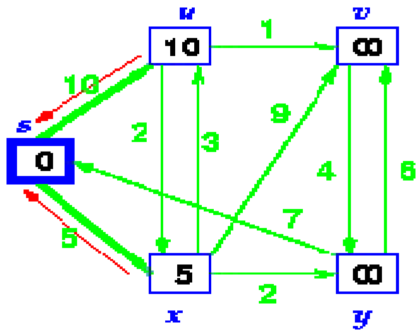

Fig 9: Predecessor updating

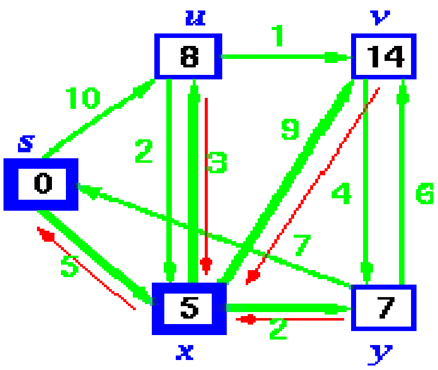

Fig 10: Predecessors for u, v and y

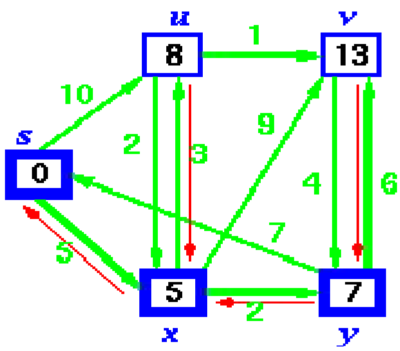

Fig 11: Predecessor adjustment

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2025.v1110.08

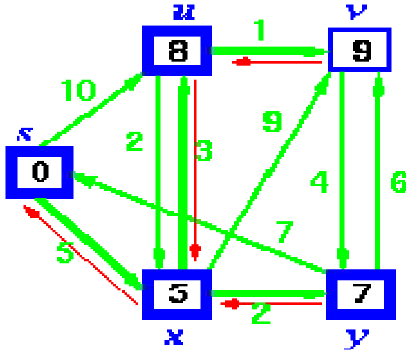

Fig 12: Adjusting node 'v'

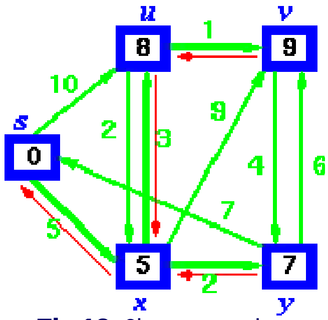

Fig 13: Shortest path

The snapshot shown in figure 17 displays the packet transmission initiated by pressing the send-new button. On pressing this button a new packet from the file selected get generated from the sender and proceeds towards the receiver. The snapshot shown in figure 18 displays the yellow packets indicate acknowledgements being sent from the receiver. Upon successful reception of each packet, the scheduling technique opted automatically adjusts and ensures that the acknowledgement reach the sender, so that errors can be reduced and the communication between sender and the receiver succeeds, as the sender comes to know that the packet is being correctly received by the receiver. The snapshot shown in figure 19 displays the successful completion of MPT. The entire process of sending and receiving packets finishes, clearly indicating that all the packets reach their ultimate destination.

III. EXPERIMENTAL RESULTS

The snapshot shown in figure 14 displays the selected file being successfully sent to multiple recipients simultaneously whenever the send button is pressed. In this snapshot the simultaneous multiple packet transmission can be visually seen on the screen of router 1. As soon as the send button is pressed, immediately at that same instant, the selected file can be seen passing through the nearest nodes called node 1, node 2 and then ultimately to the routers router 2 and router 3. Hence this snapshot provides the vital information about the file being successfully sent to the final destination selected by the user. The user had initially selected the destinations to be router 2 and router 3, therefore the file has been sent. The snapshot shown in figure 15 displays the file being successfully received by router 3, just immediately after the file is saved the address of where the file was saved appears inside a small box on the router 3 window giving assurance to the router 3 that the file received was successfully stored. The snapshot shown in figure 16 displays the file being successfully received by the router 2, just immediately after the file is saved the address of where the file was saved appears inside a

Fig 14: The selected file being successfully sent to multiple recipients simultaneously

IJIRIS: International Journal of Innovative Research in Information Security
Volume 11, Issue 10, November 2025
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2025.v1110.08

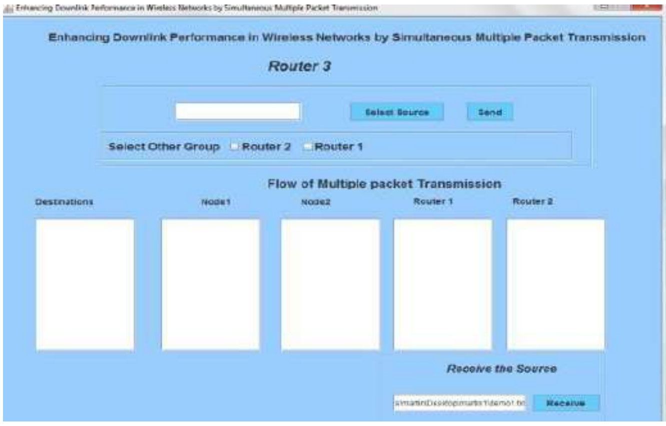

Fig 15: File being successfully received by ‘router 3’

Fig 16: File being successfully received by ‘router 2’

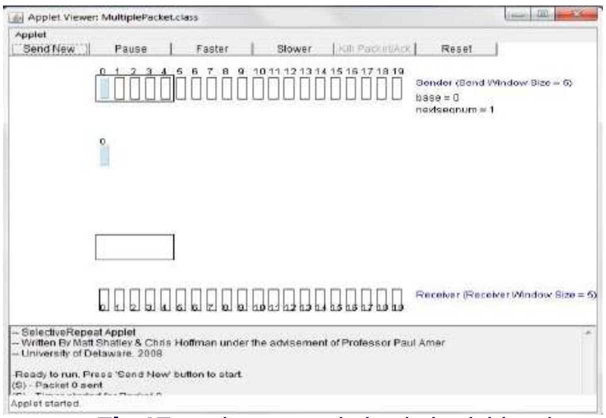

Fig 17: Packet transmission being initiated

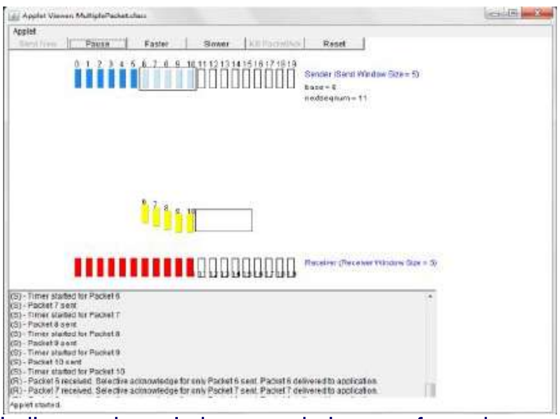

Fig 18: ‘Yellow packets’ indicate acknowledgements being sent from the receiver towards the sender

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2025.v1110.08

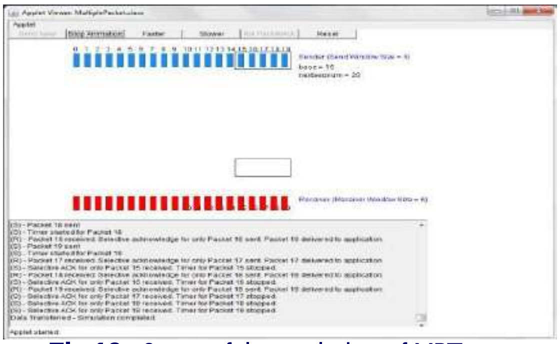

Fig 19: Successful completion of MPT

IV. FUTURE SCOPE

MPT was considered to improve the downlink performance of the wireless LANs. With MPT, the AP can send two compatible packets simultaneously to two distinct users. Formalizing the problem of finding a minimum time schedule as a matching problem and giving a practical linear time algorithm that finds a matching of at least 3/4 the size of a maximum matching algorithm an efficient cost effective technique is obtained. An approximate analytical model was used and simulations were seen in order to study the average packet delay and the results show that packet delay can be greatly reduced even with a very small compatibility probability. Java language being an open source language offered good help by providing very interesting as the results. Obtained results have to be constantly monitored such that no mistakes occur during the execution on a computer. Proper maintenance of the codes is being done. Finally, the simultaneous MPT was successfully seen with the help of snapshots and a good efficient algorithm was selected in order to increase the performance of wireless networks.

The advantages of using this work proves that by using efficient scheduling techniques for the sole purpose of enhancing downlink performance in wireless networks with the help simultaneous multiple packet transmission, is that, it yields for the following

- Ease of implementation.
- Sequential evaluation.
- Configurable accuracy.
- Prompt decision making.
- Performance enhancement.
- Optimal use of bandwidth.

In general something good for the betterment of the human society with effective utilization of available resources is achieved. Once this work gets implemented in real time, the first observation that would be very obvious is that the increase in speed of the downlink transfer due to which the internet users browsing and downloading files in the internet would be able to download bulk data required for them at a faster time interval. By increasing the total throughput of the internet system, without much delay in time, with faster access the working period of a potential user gets reduced in such a way that the user finds more time left in spare. Looking from the military point of view a quick access to the internet results in an entirely beneficial move, when it comes to finding enemy area location, places, climate forecast information and so on. Therefore the work done in this paper also finds itself useful in military communication. In medical applications, a good medical practitioner could quickly access the prescribing drug information pertaining to a particular disease from the database server of the hospital within a fraction of a second, without wasting his precious time on waiting for the downloading information but obtaining them soon, thereby getting back to attend to his patients at the earliest. Hence making the medical practitioner gets more time to attend on his patients. As the technology is always developing in a steady pace, chances are always certain, that a more advanced version arrives, that has more speed and more spectral bandwidth being allocated, a time may come that the servers throughout the world may be integrated in a sophisticated fashion, such that all these techniques may become in-built facilities, that work with a sophisticated methodology. A system that utilizes plenty of other systems, and simultaneous multiple packet transmissions may come to exist in the near future. The downlink performance can be still increased by making intermediate nodes self-tuning. An Explicit Rate (ER) feedback mechanism should be defined to design a controller for regulating the source rates. Data transfer rate is adjusted would be made at the source itself. Group node will be made able to stabilize the buffer occupancy automatically. Further research along this line of study would investigate TCP related issues and provide a solution.

V. CONCLUSION

Therefore in this paper, MPT is considered to improve the downlink performance of the wireless LANs. With MPT, the AP can send two compatible packets simultaneously to two distinct users. Java language being an open source language offered good help by providing very interesting results. Obtained results have to be constantly monitored such that no mistakes occur during the execution on a computer. Proper maintenance of the codes should be done.

Finally, the simultaneous multiple packet transmission can be successfully seen with the help of snapshots and a good efficient algorithm was selected in order to increase the performance of wireless networks.

REFERENCES

1. Zhenghao Zhang (Member IEEE), Yuanyuan Yang (Fellow IEEE) and Miao Zhao (Student Member IEEE), Enhancing Downlink Performance in Wireless Networks by Simultaneous Multiple Packet Transmission transactions on computers, vol. 58, no. 5, may 2019.
2. D. Tse and P. Viswanath, Fundamentals of Wireless Communication. Cambridge University Press, May 2015.
3. T.Lang,V. Naware, and P. Venkitasubramaniam, "Signal Processing in Random Access," IEEE Signal Processing Magazine, vol.21, no.5, September 2014.
4. G.Dimic, N.D. Sidiropoulos, and R. Zhang, "Medium Access Control-Physical Cross-Layer Design," IEEE Signal Processing Magazine, vol. 21, no.5, Sept.2014.
5. Q.Liu, S. Zhou, and G.B. Giannakis, "Cross- Layer Scheduling with Prescribed QoS Guarantees in Adaptive Wireless Networks," IEEE J. Selected Areas in Comm., vol. 23, no. 5, pp. 1056-1066,May 2020.