Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Evaluation of Packet Transfer Delivery Timeouts During Packet Transmission Using the DA-ARQ (Delay-Aware Automatic Repeat Request) Methodology

[1]V. Gokul, [2]M. Shanmugapriya
[1]Research Scholar, Department of Computer Science, Park's College (Autonomous), Tirupur, Tamil Nadu, India
gokulrj7@gmail.com
[2]Associate Professor, Department of Computer Science, Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India
priyasathyan@gmail.com

ARTICLE INFO

Received: 15 Dec 2024
Revised: 18 Feb 2025
Accepted: 26 Feb 2025

ABSTRACT

DA-ARQ is adaptive and dynamic; it can change the rate of retransmission in dependence on receiver feedback and network state. This might result in additional reliable retransmissions with increased efficiency and reduced latency. DA-ARQ adjusts its timeout duration and retransmission rate utilizing feedback mechanisms like ACKs and NACKs. This allows it to take into account the receiver's response and alter the retransmission rate based on the network's current status. DA-ARQ can reduce latency by reducing the time required to retransmit dropped packets. This may result in the quick retransmission of dropped packets by varying the retransmission rate based on network conditions. DA-ARQ can increase reliability by minimizing the number of lost packets and the impact of packet loss on network efficiency. This takes place by adjusting the network environment's retransmission rate, which may result in greater effectiveness and dependability of retransmissions. DA-ARQ can enhance network performance by altering the retransmission rate based on network conditions. This can increase reliability, reduce latency, and reduce packet loss, all of which will enhance overall network performance. The DA-ARQ (Dynamic Adaptive Automatic Repeat Request) packet timeout analysis method is employed to maximize the rate at which lost or damaged packets are transmitted again in a networked environment. It changes the timeout value and resend rate based on feedback received from the receiver and the status of the network. DA-ARQ (Delay-Aware Automatic Repeat Request) is a network communication technology that ensures reliable data transfer. It entails retransmitting lost or damaged packets to ensure they are received successfully. Machine learning techniques may be incorporated into DA-ARQ to improve networking packet analysis. To train the model, several methods, such as decision trees, random forests, or SVM, can be used. The algorithm discovers patterns and correlations between various variables in addition to the probability of successful packet delivery. By integrating DA-ARQ with machine learning, network packet transfer analysis may be enhanced, resulting in increased data transmission accuracy and effectiveness.

Keywords: DA-ARQ (Delay-Aware Automatic Repeat Request), Machine Learning, Random Forest, SVM, Packet Delivery

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

1. INTRODUCTION

The Traditional Karn's Algorithm eliminates receiver information, such as ACKs and NACKs, that may give critical facts about the network's current state. Advanced approaches, including dynamic adaptive ARQ (DA-ARQ), can enhance reliability, minimize latency, and increase efficiency by changing the retransmission rate based on network circumstances. The traditional Karn's Algorithm was a great technique at the time, but it is no longer sustainable in modern complex and dynamic network circumstances. Modern networks are complex, and contemporary, more adaptable and dynamic solutions, such as DA-ARQ, are better able to handle such complexity and can improve network performance. Ineffective retransmissions and greater latency might occur as a result of the classic Karn's Algorithm's failure to adapt its timeout value in response to changes in the network environment. The traditional Karn's Algorithm is considered obsolete to date in terms of determining packet timeout analysis because of its dependence on a predefined timeout value that is reliant on the network's predicted round-trip time (RTT). The assessment of packet transfer delivery timeouts during the transmission of packets using the DA-ARQ methodology entails examining the technique's performance in dealing with packet timeouts and retransmissions. Evaluate the data gathered to determine the efficacy of the DA-ARQ approach in dealing with packet timeouts throughout transmissions. Metrics including packet delivery ratio, average delivery time, and retransmission count should be calculated. DA-ARQ (Dynamic Adaptive Automatic Repeat Request) is a network communications approach that improves the dependability of data transfer over insecure channels. This combines the ARQ (Automatic Repeat request) protocol with dynamic adaptation methods.

Dynamic adaptation methods are used in DA-ARQ to alter the retransmission approach based on network circumstances. To make trained judgments, it continually evaluates channel quality, latency, and congestion. DA-ARQ adjusts its retransmission technique according to network circumstances. In order to recover from errors, it may alter the number of retransmissions tries, modify the timeout intervals, or apply forward error correction (FEC) methods. Dynamic adaptation approaches are used in DA-ARQ to alter the retransmission approach according to network situations and packet-level feedback. To make competent judgments, it continuously examines elements like packet loss rate, latency, and congestion. DA-ARQ changes its retransmission technique at the packet level according to observed network circumstances and responses from the receiver. To recover from failures, it may alter the number of retransmissions tries for individual packets, modify the timeout intervals per packet, or employ forward error correction (FEC) algorithms on specific packets. DA-ARQ additionally takes into account network congestion at the packet level. If congestion is observed, the retransmission rate for certain packets can be reduced, or congestion control techniques may be used to mitigate subsequent congestion. DA-ARQ operates at a smaller size, enabling the selective retransmission of certain packets that have been recognized as dropping or incorrect instead of retransmitting the whole data stream. The retransmission procedure is repeated at the level of the packet until either all packets are successfully transmitted to the receiver or a limit number of retransmissions for particular packets has been reached. DA-ARQ changes the retransmission technique according to observable network circumstances and packet-level feedback. In order to recover from failures, it may alter the number of retransmissions attempts for individual packets, modify the timeout intervals for each packet, or use forward error correction (FEC) algorithms on certain packets. DA-ARQ (Distributed Adaptive Automatic Repeat Request) is a wireless transmission paradigm that optimizes data transfer by avoiding mistakes and retransmissions. DA-ARQ uses error detection techniques such as CRC (Cyclic Redundancy Check) to determine if any errors occur throughout data transfer. DA-ARQ's retransmission technique is determined by the channel conditions. It modifies the number of retransmissions dynamically according to the perceived quality of the wireless channel. FEC techniques, in which redundant information is supplied to the transmitted data to facilitate error detection and correction at the receiver end, can also be used with DA-ARQ. It also serves to reduce the requirement for retransmissions.

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

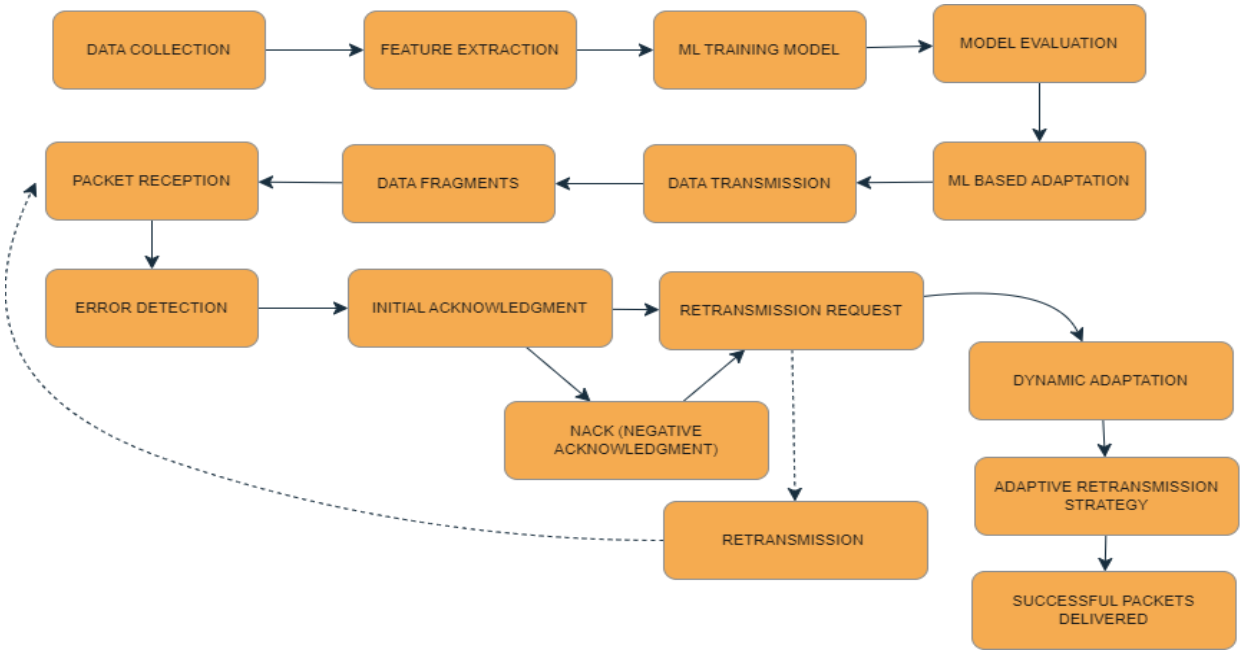

Figure 1: Proposed Architecture Diagram

2. LITERATURE REVIEW

Computer networks [1] have developed to pose serious threats since they are constantly targeted by various attacks. Emerging exploits and trends are being transmitted; these attacks target all open ports that are accessible on the network. For this, a number of technologies are available, including network mapping and vulnerability screening. In recent years, machine learning (ML) has been a popular method for enhancing the Intrusion Detection System's (IDS) ability to identify malicious network traffic. The quality of the dataset used to train the model is essential to how well ML models identify anomalies. In order to help IDS discover network traffic abnormalities, this study suggests a detection framework with an ML model. This detection technique makes use of a dataset that is made up of both malicious and legitimate traffic. The primary challenges in this research are the acquired characteristics needed to train the ML model regarding different assaults and differentiate between anomalous and regular data. The ISOT-CID network activity database is used in the ML model's training phase. We developed certain important column properties and gave our approval for them to be included in the training of the ML model. The traffic portion of the ISOT-CID dataset has two different types of characteristics: the first is taken via network traffic flow, while the other features are computed at predetermined intervals of time. We also demonstrated a brand-new column feature that was introduced to the source data and received approval for its improvement in detection quality. This functionality is dependent on the traffic flow's rambling packet payload duration.

For every component of traffic that is moving across a network, the proper identifier is recognized as network traffic classification. Network traffic categorization [2] has historically been accomplished using a variety of techniques, such as port-based, payload-based, behavior-based, and so forth. These techniques are shown to have specific constraints. Machine Learning (ML) techniques that rely on the statistical characteristics of the flow of traffic are currently receiving attention. When faced with massive amounts of traffic information that contains a lot of characteristics and instances, ML approaches fail to perform well. After submitting the data to machine learning classifiers, feature selection is used to eliminate irrelevant and redundant information. In this study, distinct perspectives—one involving feature selection while another which doesn't—are used to illustrate how network traffic may be classified using ML techniques. Runtime, accuracy, recall, precision, and F- score are only a few of the results measures taken into account. According to the experimental findings, a featureless classification

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

has a mean accuracy and runtime of 94.14% and 0.52 seconds, correspondingly. In contrast, the approach that uses feature selection has an accuracy rate of 95.61% with an average duration of 0.25 seconds. The rise in accuracy attained illustrates how crucial it is to use just pertinent and non-redundant features when employing ML approaches. To provide the highest level of accuracy, it was advised that feature selection should be integrated into the network categorization procedure. Numerous objectives of traffic [3] analysis include assessing the efficiency and security of network administration and activities. As a result, the analysis of network traffic has significance for enhancing the security and efficiency of networks. In this study, various machine learning methods for traffic analysis are discussed. New methods for detecting intrusions, analyzing virus activity, categorizing Internet traffic, and other security-related tasks are required due to increased network traffic with the growth of artificial intelligence. Networking issues may be effectively solved by machine learning (ML). This study presents an overview of various traffic analysis methodologies.

The analysis and forecasting of network traffic [4] has a wide range of applications and has recently garnered a sizable amount of research. To pinpoint numerous issues with current computer network applications, several experiments are carried out and then reported. Predicting network traffic is a proactive way to assure safe, dependable, and high-quality network communication. BBR [5] delivers data according to the bandwidth delay product (BDP), which assesses the restriction in bandwidth and round-trip propagation time consecutively. A flow with a lengthy RTT is unfairly given more bandwidth than a flow with a short RTT simply due to its dependency on the BDP. A brief RTT flow appears more vivid than a lengthy RTT flow in typical TCP, which runs completely counter to the nature of the BBR. We earlier suggested the bottleneck queue buildup suppression approach to address this issue, and NS-3 was used to validate it. However, we discovered that the prior approach was unstable and produced unanticipated outcomes throughout the testbed tests. As a result, we provide a better method in this study than the one that was previously suggested. ISPs [6] use network traffic categorization primarily to study the characteristics needed to construct the network, which has an impact on the efficiency of the network as a whole. There are several methods used to categorize network protocols, including port-based, payload-based, and machine learning-based approaches. Each has advantages and disadvantages of its own. In this article, we contrast the performance of two fundamental algorithms, Naive Bayes and K closest neighbor, when applied to a networking data set that was taken from a streaming video stream using the Wireshark program. Python's Sklearn package and the utility libraries Numpy and Pandas are used to create a machine learning algorithm. We find that the K-closest method provides a more precise forecast than Naive Bayes, Decision Tree, and Support Vector Machine techniques.

Numerous network operations [8], including QoS and network management, depend on accurate traffic classification. Machine learning techniques have potential in numerous capacities, but port- or payload-based categorization approaches are becoming harder and harder to use. By adjusting the variable selection probability in this study in accordance with the significance of the associated variable to categorize network traffic, we enhance the basic Random Forest model. Our test results demonstrate that the Improved Random Forest performs better in terms of categorization. Additionally, building the model takes a shorter amount of time. In this study [9], we investigate an innovative approach for end-to-end round-trip time (RTT) estimation based on the expert framework, a machine learning method. In our concept, a group of 'experts' each predicts a particular quantity. The RTT is computed using the weighted average of these predictions, with the weights changing depending on the discrepancy between the estimated and real RTT after each RTT measurement. We demonstrate through extensive simulations that the suggested machine-learning system responds to changes in the RTT relatively rapidly.

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3. METHODOLOGY

Packet Transfer Analysis

The core objective of packet transfer delivery timeouts is to guarantee that data packets are delivered on time. The efficacy of timeouts ought to be measured in regards to the proportion of effectively transmitted packets during the set timeout period. Timeout durations must be adjusted in such a way that they achieve a compromise between ensuring timely delivery and reducing needless delays. Assessing the delay imposed by timeouts aids in determining if the durations specified are acceptable for the particular communication system. Lower latency indicates quicker packet delivery. Timeouts cause lost or defective packets to be retransmitted. Assessing the efficiency of retransmissions is critical to comprehending how successfully timeouts aid in data recovery. Timeouts have to be able to respond to varying network circumstances. It is critical to assess how successfully timeouts react to varying degrees of network congestion, latency, and error rates. An adaptive timeout mechanism can enhance the delivery process dynamically, guaranteeing efficient data transfer. It is essential to assess the effect of timeouts on network consumption. Setting timeouts that are excessively brief can result in inefficient retransmissions, causing network congestion. Configuring timeouts excessively, on the other hand, may cause lost packet recovery to take longer. It is critical to balance network consumption with timely packet delivery. It is critical to assess the resilience of the timeout mechanism to verify that it can withstand unexpected events and changes in network circumstances.

Duplicate transmissions happen if the same packet is received at the destination several times. This may happen as a result of network congestion, packet retransmissions, or incorrectly configured network devices. If a number of packets fail to get to their destination, this is referred to as a loss of packets. This might happen as a result of network congestion, buffer overflowing, or routing issues. If any number of bits in a packet are damaged or reversed during transmission, bit errors occur. Bit mistakes can be induced by a number of sources, including noise, interference, and signal attenuation. Delayed packets arise whenever packets travel with considerable delays to their destination. Whenever there are issues with the implementation or comprehension of network protocols, protocol errors occur. The starting point in network packet analysis is to capture packets from the network using techniques that include packet sniffers or networking analysers. For additional examination, these tools record packets from the network interface or select network segments. Once recorded, the packets may be examined to obtain insight into network traffic patterns. Examining packet sizes, protocols utilized, source and destination addresses, and the number of ports is all components of this. Traffic analysis assists in identifying the types of network activity and detecting any anomalies or trends that may impact efficiency. Analyzing the protocols utilized by network packets is part of network packet analysis. This involves studying the headers of packets and payloads to learn how various protocols are implemented and interact with one another.

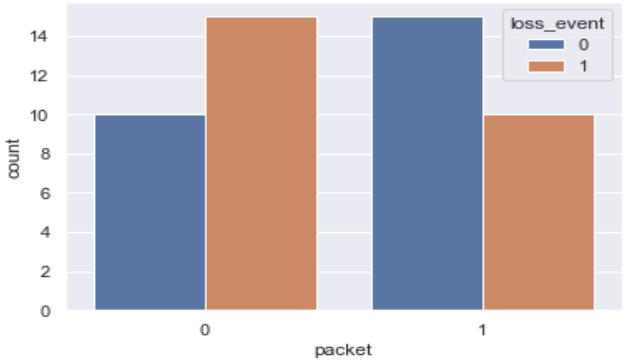

Figure 2: Random Packets with loss_events

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DA-ARQ Algorithm

The DA-ARQ technique (Decentralized Automatic Repeat Request) is a type of error management technique employed by systems for communication to provide reliable data transfer over unstable channels. It is developed primarily for networks that are decentralized with no centralized control or coordination. Within the DA-ARQ method, every node in the network handles error identification and recovery individually and autonomously. Through a communication channel, the sender node sends packets to the receiving node. Each packet is given a distinct sequence number. The DA-ARQ (Dynamic Adaptive Automatic Repeat Request) packet timeout evaluation technique is employed to maximize the rate at which damaged or lost packets are retransmitted in a networked system. It changes the timeout value and resend rate based on the feedback of the receiver and the status of the connection. By integrating Automatic Repeat Request (ARQ) with Forward Error Correction (FEC) technologies, the DA-ARQ communication network improves the transmission of packets across unstable networks. In regard to packet delivery analysis, DA-ARQ outperforms ordinary ARQ protocols because it incorporates network delay constraints and changes the rate when packets are transmitted again based on the projected delay. Typical ARQ systems function by transmitting lost messages until the intended node properly accepts them. It also changes the retransmission rate depending on the projected network delay to maximize packet delivery while minimizing latency.

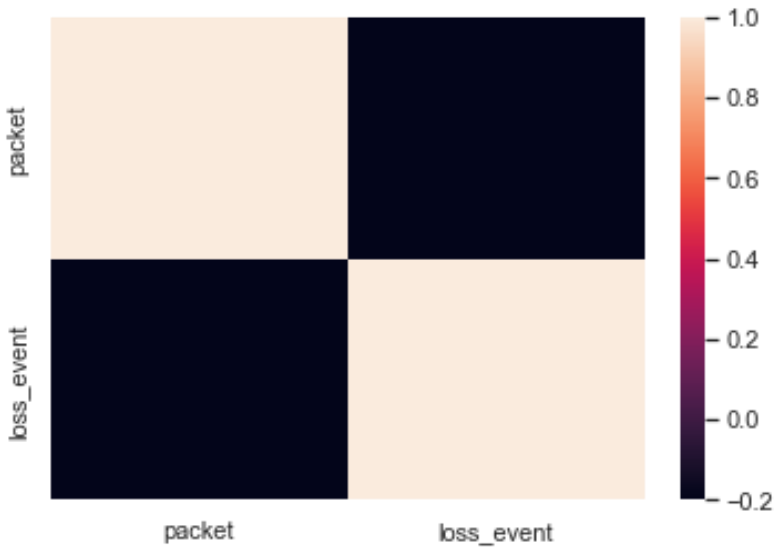

Figure 3: Correlation Matrix Structure

DA-ARQ is more effective at assessing packet delivery and increasing packet delivery while reducing delay because it integrates ARQ and FEC algorithms and alters the retransmission rate based on the projected network delay.

The DA-ARQ technique adds redundant data to the original data, allowing packet recovery without the need for a retransmission request. It may provide a higher level of reliability and data integrity by retransmitting only the packets that were lost rather than the entire sequence. It can handle more sophisticated circumstances like packet duplication and reordering by utilizing advanced techniques, including forward error correction (FEC) and interleaving. These techniques also adjust the parameters in response to changing network conditions, such as shifting error rates and bandwidth utilization. It primarily improves resource use by dynamically altering the transmission rate and modulation method. As a result, the system's efficiency improves and energy usage decreases. The DA-ARQ method uses a dynamic technique to adjust the retransmission rate according to network circumstances, which makes it more effective than traditional Karn's and other algorithms. This reduces the need for retransmissions

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

and improves the overall efficiency of the system. The DA-ARQ method reduces latency by adjusting the retransmission rate based on the network's circumstances. This enhances the overall efficiency of the system and shortens the time required to transport data. The DA-ARQ technique is more reliable than the standard Karn's approach because it uses a system of feedback to discover and correct mistakes. This reduces the number of errors and enhances the overall reliability of the system. The DA-ARQ approach can manage more users and respond to changing network conditions, thereby rendering it more efficient.

4. CONSTRUCTION

Mathematical Analysis of DA-ARQ Algorithm

DA-ARQ Algorithm describes the basic operation of protocol for packet transfer between a sender node and receiver node. By retransmitting packets that are not acknowledged within the timeout length, this technique assures reliable packet transport. The receiver transmits ACKs to verify the packet receipt and discards duplicate packets. The sender maintains track of retransmission counts and stops retransmitting when the maximum limit is reached. The procedure ends by distributing the analysis, which includes the total number of predicted packets and the number of packets that were received by the receiver.

Sender Node:

In the sender node approach, the process sets two variables to their default values: max_Retransmissions and timeout_Duration. These variables specify the maximum number of retransmissions permitted for each packet as well as the time the sender node must wait for an acknowledgement (ACK) from the receiver. The method enters a loop for every packet included in the total_PacketsCount. The method sets the retransmission_Count to 0 and the isTransmitted flag to false during the loop. The algorithm subsequently enters another loop, which continues till either the retransmission count surpasses the max_Retransmissions value or the packet is successfully transferred. The sender node transmits the packet to the receiver node and begins a timer for the transmitted packet inside the inner loop. The sender node then awaits an acknowledgement from the receiving node. If the ACK arrives within the timeout duration, the isTransmitted flag will be set to true, indicating that the transmission was successful. If there is a timeout before obtaining the ACK, the retransmission count is increased, and the message is reissued. The resend packet's timer is resumed. The inner loop is repeated until either the packet has been transmitted effectively or a certain number of retransmissions are achieved.

Step 1: The sender initializes the maximum number of retransmissions allowed for each packet (max_Retransmissions) and the duration for which it waits for an acknowledgement. (timeout_Duration).

- *Initialize variables: max_Retransmissions, timeout_Duration*

Step 2: The sender iterates through each packet to be transmitted.

- *For packet_Number in range(total_PacketsCount):*
- If *retransmission_Count = 0* - The sender initializes the retransmission count for the current packet to 0
- *isTransmitted = false* condition - The sender sets a flag to false, indicating that the current packet has not been successfully transmitted yet.

Step 3: The sender attempts to transmit the current packet multiple times until either it reaches the maximum number of retransmissions or the packet is successfully transmitted.

- *While retransmission_Count < max_Retransmissions and not isTransmitted:*

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- *Send packet to receiver node*
- *Start timer for the sent packet*

Step 4: Wait for ACK - If an acknowledgment is received within the specified timeout duration, the sender marks the packet as successfully transmitted by setting the isTransmitted flag to true.

- *If ACK is received within timeout_Duration & isTransmitted = true*

If timeout occurs:

- Increment retransmission_Count

- Resend the packet

- Restart timer for the resent packet

If the timeout duration elapses before receiving an acknowledgment, the sender increments the retransmission count and retransmits the packet. The timer is restarted for the resent packet.

Receiver Node:

In the receiver node approach, it sets two variables to their starting values: expected_SeqNum and total_PacketsCount. The expected_SeqNum is the number that represents the sequence of the next packet to be received, and the total_PacketsCount reflects the overall number of packets to be received. The method enters a loop that is repeated until the expected sequence number equals the total packet count. The receiver node receives a packet from the sender node inside the loop. If the anticipated sequence number is present in the received packet, the data is transmitted to the higher layer, an ACK with the expected sequence number is transmitted back to the sender node, and the expected sequence number is increased. If the sequence number of the received packet is more than the expected_SeqNum, the receiver node sends an ACK for the last successfully received packet to notify the sender node of the missing packets. If the packet that was received is a duplicate (it has the same sequence number as another packet), the receiver node discards it and provides an ACK for the previous validly received packet. The loop will continue until all of the predicted packets have been received. By retransmitting packets until the packets are acknowledged successfully or the maximum number of retransmissions has been reached, the DA-ARQ algorithm assures reliable packet transport. The receiver node acknowledges packet reception and correctly handles duplicated or missing packets.

Step 1: The receiver initializes the expected sequence number (expected_SeqNum) to 0 and the total number of packets it expects to receive (total_PacketsCount).

Initialize variables: expected_SeqNum, total_PacketsCount

Step 2:

The receiver continues to receive packets until it has received all of the intended packets.

While expected_SeqNum < total_PacketsCount

Receive packet from sender node

- If packet has expected_SeqNum:

- Deliver data to upper layer

- Send ACK to sender with expected_SeqNum

- Increment expected_SeqNum

- If packet has sequence number greater than expected_SeqNum:

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Send ACK for the last correctly received packet

- If packet is a duplicate:

- Discard the packet

- Send ACK for the last correctly received packet

Step 3: Result outcome indicates whether the packet got transferred at the receiver side.

- *Print "Packet transfer analysis:"*

- Print "Total packets expected: " + total_PacketsCount

- Print "Total packets received: " + expected_SeqNum

If the received packet contains the expected_SeqNum, the receiver passes the data to the top layer, sends an acknowledgement (ACK) to the sender with the anticipated sequence number, and increments the expected sequence number.

If the received packet contains a sequence number that is larger than the expected sequence number, the receiver sends an ACK for the last successfully received message to request retransmission of the missing packet.

If the packet is identical: If the received packet is a duplicate (has a sequence number that the receiver has already received), the receiver discards it and provides an ACK for the last validly received packet.

5. EXPERIMENTAL ANALYSIS

RTT Packet Transfer using DA-ARQ Algorithm

The round-trip time (RTT) in DA-ARQ packet transfer signifies the time required for the packet to be transmitted from the sender's node, reach the receiver node, and be acknowledged (ACK) sent again to the sender node. The RTT represents the time it takes for a packet to be processed, transmitted from the sender node, arrive at the receiver node, and then be acknowledged (ACK) by the sender node.

Timeout Duration (To): The sender node specifies a timeout period, indicating the maximum amount of time it can wait for an ACK before concluding that the packet was dropped during transmission. If the sender node does not receive an ACK within this delay period, a retransmission is initiated. It is the amount of time the sender node waits for an ACK before believing the packet was lost throughout transmission and starting a resend.

Processing Time (Tp): It denotes the amount of time it took the receiver to process the received packet and create the ACK message.

Propagation Time (Tp): It indicates the time it takes for a packet and an acknowledgement to go across the network among the sender of the message and the recipient. This time is determined by the distance among sender and receiver along with the speed of propagation through the medium of communication.

Transmission Time (Tt): It is the amount of time it takes for a packet to be transferred from the sender to the recipient. It involves the time required physically transmitting the packet over the network.

$$RTT = Tt + Tp + Tp$$

The timeout duration (To) can be determined depending on the predicted RTT or by taking into account network circumstances. The DA-ARQ method dynamically modifies the timeout length based on observed RTT fluctuations to maximize RTT and decrease retransmissions. The mathematical analysis gives a foundation for understanding the variables impacting RTT and how it might be adjusted, but

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

the particular values for Tt, Tp, and To rely on the network's characteristics underlying the DA-ARQ algorithm implementation.

Machine Learning with DA-ARQ for Packet Transfer

Machine Learning (ML) could potentially be applied with the DA-ARQ method to improve network packet transfer performance. ML may be used to enhance the DA-ARQ algorithm in a variety of ways, including improving retransmission techniques, anticipating network conditions, and identifying and recovering from packet errors. To enhance the retransmission approach, ML systems can gain insight from previous information. To estimate the ideal number of retransmissions or the appropriate timeout length for each packet, ML models may examine network circumstances, packet loss patterns, and other relevant aspects. This can help reduce latency and eliminate unintentional retransmissions. Utilizing historical data and real-time observations, ML models may be trained to anticipate network circumstances, including congestion or connection quality. These projections may be used to dynamically alter DA-ARQ algorithm variables, including the maximum number of retransmissions or the timeout length, to adapt to evolving network circumstances and enhance overall efficiency. Machine learning algorithms can learn to recognize trends in packet loss or corruption and forecast what packets will be impacted. This data may be utilized to prioritize packet retransmission or to implement error correction algorithms to retrieve lost or damaged data. ML models may assess network parameters and dynamically optimize the size of packets.

```
# Defining DA-ARQ Algorithm for Packet Transfer

def da_arq(packet):

    # Perform packet analysis and return prediction

    prediction = rf_classifier.predict([[packet]])

    return prediction
```

To improve the retransmission approach, ML algorithms may examine RTT fluctuations and trends. For instance, if the ML model identifies an unanticipated increase in RTT, it can initiate proactive packet retransmission to minimize potential congestion or network concerns. The DA-ARQ method can improve its performance and minimize packet loss by adjusting its retransmission behavior according to RTT changes. Based on real-time RTT data, ML models are able to dynamically alter the timeout length. ML algorithms may optimize the timeout value for every packet of transmission by investigating the link between RTT and network circumstances. This adaptive adjustment guarantees that the DA-ARQ algorithm responds to shifting network circumstances, which reduces wasteful retransmissions and improves overall efficiency. ML approaches may be used to optimize retransmission strategies, timeout duration, and packet size in the DA-ARQ algorithm. In the absence of packet loss, ML may nevertheless increase performance by dynamically changing these settings depending on real-time network conditions. The ML model may learn network patterns and trends, resulting in improved resource usage and perhaps lower latency. It is now feasible to proactively change transmission rates or retransmission behavior according to these forecasts by introducing ML within the DA-ARQ algorithm.

 # DA-ARQ Algorithm with Machine Learning for Packet Analysis

- *Add the required library (NumPy).*
- *Set the num_packets, loss_probability, and rtt_threshold parameters.*
- *Data generation using random packets: Using random values 0 and 1, create an array state of packets with size num_packets.*
- *Creating an array state of loss_events packets of size num_packets and generating random loss events by utilizing random values 0s and 1s*

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- *Setting window size and initializing "window_start" and "window_end" as well as the ARQ variables*
- *'acked_packets' stores acknowledgement packets, while 'predictions' is a list of empty entries.*
- *Next, initiate classifiers by creating instances.*
- *The process iterates until the 'window_end' reaches the 'num_packets'. It uses array slicing to choose the packets inside the current window for each iteration.*
- *Utilizing np.array(), train the classifier using the input data that has been transformed into a 2D array.transform(-1, 1).*
- *To guarantee that packets are received, determine with precision the status of the current window.*
- *The window is recognized if the 'window_accuracy' value is greater than 0.5. Alternatively, the predictions are added as 0s to "predictions" and the unacknowledged packets are added to "acked_packets."*
- *The mean of the loss events within the window is multiplied by the 'rtt_threshold' to determine the round-trip time (RTT) for the current window.*

The perceptions of "num_packets," "loss_probability," and "rtt_threshold" indicate the overall number of packets, the probability of a packet loss event, and the round-trip time threshold, respectively. Random values of 0s and 1s are used to populate the 'packets' array, which has a size of 'num_packets'. Using random values of 0s and 1s, an array identified as "loss_events" of size "num_packets" is constructed, whereby the likelihood of a 1 is determined by the "loss_probability." This shows when occurrences involving packet loss occur. Variables in the ARQ are set to zero. The value of "window_size" is set to 10, which denotes the quantity of packets for each window. 'window_start' and 'window_end' have been modified to 0 and 'window_size', correspondingly. The empty lists 'acked_packets' and 'predictions' will be used to record the acknowledged packets and the accompanying predictions. The main loop continues to execute until window_end' approaches num_packets. It uses array slicing to choose the packets inside the current window for each iteration. The classifiers ('rf_classifier', 'lr_classifier', and svm_classifier) are trained using the acknowledged packets and the predictions that go along with them. The input data is transformed to a 2D array using 'np.array().reshape(-1, 1)' when the 'fit()' function is invoked. The accuracy of the packets inside the current window is then predicted using the classifiers. Forecasts depend on input data that has undergone the same reshaping as training data. The mean of the loss events within the window is multiplied by the 'rtt_threshold' to determine the round-trip time (RTT) for the current window. The classifiers are trained and used to forecast every classification's accuracy.

Comparative Analysis

It entails the sender transmitting packets to the receiver and the recipient responding with acknowledgments (ACKs), signaling effective receipt of the packets. If the sender of the message does not get an ACK within a defined timeout interval, the packet is retransmitted. Random Forest has been utilized for predicting DA-ARQ performance in a variety of ways, including retransmission accomplishment, appropriate timeout length, and packet loss detection. It can deal with numerical and categorical variables and offers feature relevance rankings to aid with comprehending the underlying reasons impacting DA-ARQ performance. SVM finds the best hyperplane for separating distinct classes or predicting continuous values. SVM could potentially be employed in the context of DA-ARQ prediction to identify packets that are effectively transported or lost depending on network circumstances, packet size, or retransmission history. The suitable ML approach is determined by a number of parameters, including the unique problem, dataset size, feature complexity, interpretability requirements, and accessible computing power.

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

 # Calculate accuracy of each classifier

rf_accuracy = accuracy_score (predictions,
rf_classifier.predict(np.array(acked_packets).reshape(-1, 1)))

lr_accuracy = accuracy_score (predictions,
lr_classifier.predict(np.array(acked_packets).reshape(-1, 1)))

svm_accuracy = accuracy_score (predictions,
svm_classifier.predict(np.array(acked_packets).reshape(-1, 1)))

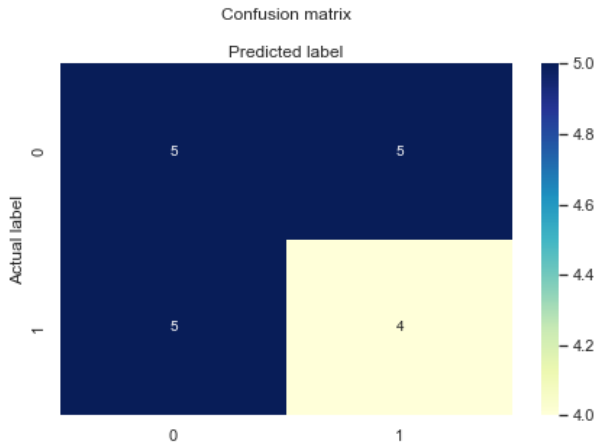

Figure 4: Confusion Matrix for ML Classifier

By using advanced pattern recognition and anomaly detection techniques, machine learning algorithms possess the potential to increase the accuracy of packet analysis. Algorithms are able to extract information from enormous data sets and discover complicated patterns that standard approaches might overlook. These may result in more precise detection of packet errors, network overload, or malicious activity. Incorporating machine learning with the DA-ARQ method contributes to the broader system's complexity. Building and evaluating machine learning models involves knowledge of data analysis, feature engineering, and model selection. Machine learning could assist in detecting complicated patterns and abnormalities, whereas the DA-ARQ algorithm can cope with packet retransmissions and assure consistent delivery. Overall, using machine learning can improve the capabilities of the DA-ARQ algorithm and deliver more sophisticated results.

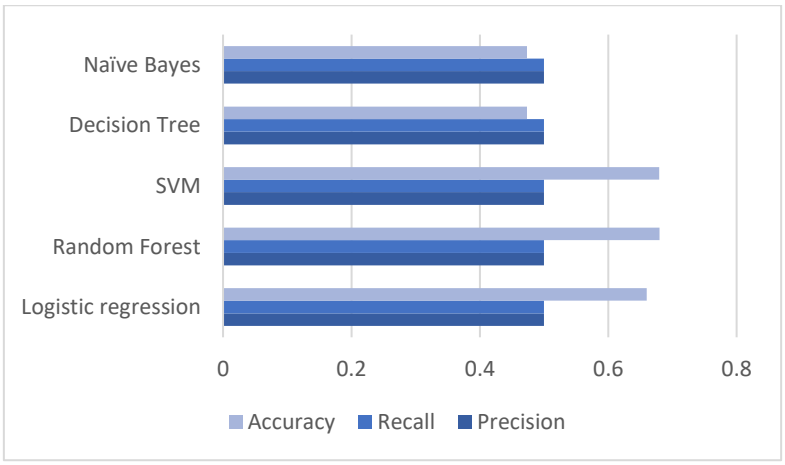

Figure 5: Analysis of DA-ARQ with ML Approach

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6. CONCLUSION

The DA-ARQ technique uses acknowledgement (ACK) messages and retransmissions to verify that all packets are received at the receiver. The DA-ARQ method guarantees all packets are received at the receiver node by leveraging these processes. The receiver sends ACK messages to indicate its receipt of each packet, letting the sender maintain track of effective transfers. If the sender does not get an ACK within the delay period, the packet is retransmitted to assure delivery. To notify the sender of any incomplete packets, the receiver processes duplicate packets and delivers ACKs for the last successfully received packet. DA-ARQ (Decentralized Automatic Repeat Request) is a network protocol that ensures reliable packet delivery between a sender and a receiver. DA-ARQ also provides flow control methods to prevent overloading the receiver or the network with several packets. These methods guarantee that the transmitter adjusts its transmission rate in accordance with the receiver. DA-ARQ supports reliable packet transfer in networks by identifying and rectifying errors, retransmitting packets that were dropped, and altering the transmission rate dependent on the receiver's capabilities. It contributes to the accuracy and reliability of data transmission in a variety of network applications, including file transfers, messaging via email, online surfing, and media streaming.

7. REFERENCES

[1] Alshammari, A., Aldribi, A. Apply machine learning techniques to detect malicious network traffic in cloud computing. J Big Data 8, 90 (2021). https://doi.org/10.1186/s40537-021-00475-1
[2] Jonathan, Oluranti & Misra, Sanjay & Osamor, Victor. (2021). Comparative Analysis of Machine Learning techniques for Network Traffic Classification. IOP Conference Series: Earth and Environmental Science. 655. 012025. 10.1088/1755-1315/655/1/012025.
[3] Alqudah, Nour & Yaseen, Qussai. (2020). Machine Learning for Traffic Analysis: A Review. Procedia Computer Science. 170. 911-916. 10.1016/j.procs.2020.03.111.
[4] Aldhyani, Theyazn. (2020). A review of network traffic analysis and prediction techniques.
[5] G. -H. Kim, Y. -J. Song, I. Mahmud and Y. -Z. Cho, "Enhanced BBR Congestion Control Algorithm for Improving RTT Fairness," 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2019, pp. 358-360, doi: 10.1109/ICUFN.2019.8806064.
[6] S. Patel, A. Gupta, Nikhil, S. Kumari, M. Singh and V. Sharma, "Network Traffic Classification Analysis Using Machine Learning Algorithms," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 1182-1187, doi: 10.1109/ICACCCN.2018.8748290.
[7] 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.
[8] Wang, Chao & Xu, Tongge & Qin, Xi. (2015). Network Traffic Classification with Improved Random Forest. 78-81. 10.1109/CIS.2015.27.
[9] Nunes, Bruno & Veenstra, Kerry & Ballenthin, William & Lukin, Stephanie & Obraczka, Katia. (2014). A machine learning framework for TCP round-trip time estimation. EURASIP Journal on Wireless Communications and Networking. 2014. 47. 10.1186/1687-1499-2014-47.
[10] B. A. A. Nunes, K. Veenstra, W. Ballenthin, S. Lukin and K. Obraczka, "A Machine Learning Approach to End-to-End RTT Estimation and its Application to TCP," 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), Lahaina, HI, USA, 2011, pp. 1-6, doi: 10.1109/ICCCN.2011.6006098.
[11] P. S and S. -H. Chung, "An Efficient Algorithm for the Performance of TCP over Multi-hop Wireless Mesh Networks," 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 2010, pp. 816-821, doi: 10.1109/ITNG.2010.129.
[12] J. Yang, S.. -H. S. Huang and M. D. Wan, "A clustering-partitioning algorithm to find TCP packet round-trip time for intrusion detection," 20th International Conference on Advanced Information

Research Article

Networking and Applications - Volume 1 (AINA'06), Vienna, Austria, 2006, pp. 6 pp.-236, doi: 10.1109/AINA.2006.13.

[13] Jin Zhao and Fuyan Zhang, "A scalable RTT estimation algorithm for multicast congestion control," 2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914), Chengdu, 2004, pp. 571-574 Vol.1, doi: 10.1109/ICCCAS.2004.1346197.

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

[15] O. Elloumi, H. Afifi and M. Hamdi, "Improving congestion avoidance algorithms for asymmetric networks," Proceedings of ICC'97 - International Conference on Communications, Montreal, QC, Canada, 1997, pp. 1417-1421 vol.3, doi: 10.1109/ICC.1997.595022.