International Journal of Research in Computer Science Engineering and Technology
ISSN:3049-1665 (Online) IJRCET-2024 Paper, Volume 1 Issue 1, pp:10-19, www.ijrcet.in

An Efficient Packet Delivery Approach for Optimized Wireless Communication Systems

[1] **Gotham Jessie, Software Developer, jessiegotham@neoshaantechnologies.com**

[2] **Kongara Gowthamraju, Developer, gowthamrajukongara@neoshaantechnologies.com**

Abstract-

Wireless communication networks have some big problems when it comes to sending packets. These issues include interference from signals, crowded networks, moving devices, and changing network conditions. Older methods, like flooding, help send packets reliably but can waste energy, cause more traffic, and result in sending the same packet multiple times. To solve these issues, we propose a new model that mixes Reinforcement Learning (RL) with the Flooding method to improve how packets are sent. This new way helps ensure packets get delivered reliably while cutting down on sending duplicate packets and lowering the strain on the network. When one packet gets to its destination, any extra copies are thrown away, which makes better use of the network and helps it work better, especially in large or busy systems.The RL part of this model learns the best ways to send messages by adjusting to what's happening in the network, which means less wasted effort and better use of resources. This method not only boosts how reliably packets are delivered but also helps reduce congestion and save energy, making it great for networks with limited resources, like wireless sensors and mobile networks.By combining RL's ability to adapt quickly with the solid reliability of the Flooding method, we create a good solution for sending packets. This method is especially useful in situations like disaster recovery, Internet of Things (IoT) applications, and communication for self-driving cars, where it's important to share information efficiently and reliably.

Index Terms: Wireless Communication Networks, Packet delivery, Congestion, Network Congestion, Reinforcement Learning (RL).

1. Introduction

In wireless communication, packet delivery is how we send data packets from one place to another without wires. Data gets split into smaller parts called packets, and these packets travel separately through the network. Each packet has important information, like where it's going, its order, and checks to see if there are mistakes. However, delivering packets in wireless networks is tricky. Things like signal interference, too much traffic, movement, and changing conditions can complicate things. While the packets move through the network, they might experience delays, get lost, or become damaged. This means we need methods like fixing errors, sending packets again, and following rules to help with the delivery. Tools like TCP (Transmission Control Protocol) or newer technologies like MIMO (Multiple Input, Multiple Output) and OFDM (Orthogonal Frequency-Division Multiplexing) are often used to make packet delivery better and help reduce data loss in changing wireless setups.

There are many problems that can affect how well wireless communication works. A big issue is signal interference. This happens when things like buildings, trees, or other devices mess with the signal, causing packets to get lost or delayed. The wireless channels can also change due to weather or movement, which affects the signal quality. This can slow down data transfer or lead to more mistakes. Packet loss is another common issue. Sometimes packets get dropped if too many people are using the network at once, or if the signal isn't strong enough. This is especially an issue when devices move around and switch connections, causing delays during the switch. On top of that, wireless networks face security problems like eavesdropping and unauthorized access, which can put the safety and privacy of the data at risk. To deal with these problems, we use advanced techniques like adjusting how we send signals, correcting errors, and sending packets again. Still, since wireless communication is always changing, it can be hard to keep packet delivery reliable and efficient.

In the Flooding method, when a node gets a data packet, it shares it with all its neighbors, who then do the same with their neighbors until the packet reaches its final spot. Flooding makes sure that the packet spreads all over the network, even when the network structure changes quickly, like in mobile networks. While flooding guarantees that the packet will arrive, it can lead to problems like too much traffic, using too much energy, and having extra copies of packets, which can slow down the network. To help with these issues, improvements like tracking sent packets to avoid duplicates and setting time limits on packets

are often used. Even though flooding is simple, it works well when a reliable, low-cost option is needed, but it struggles to keep up in bigger networks because of these drawbacks.

2. Related Work

The flooding method used in wireless communication helps to send packets, but it has some big problems that can hurt network performance. One major issue is network congestion. Every node sends the packet to all its neighbors, which means the same packet can be sent out many times. This can create a lot of extra traffic, especially in large networks, causing delays, collisions, and lower performance. Another concern is energy use, especially for battery-operated devices like sensors. Constantly sending packets uses a lot of energy, which can make these devices run out of power faster. Additionally, flooding can create duplicate packets since nodes might send the same packet more than once, adding unnecessary load to the network. Flooding also struggles with larger networks. As the network gets bigger, the amount of broadcasting increases a lot, making it less efficient. Plus, flooding does not choose the best paths for sending packets; it just sends them out without considering things like link quality or distance, leading to less effective routes and making the network less efficient.To help with these issues, methods like duplicate suppression and time-to-live (TTL) values are often used, along with better routing protocols. However, these improvements do not completely fix the problems that come with flooding.

Another problem is that flooding can waste resources. Because each node sends packets to all its neighbors, it creates a lot of duplicate messages. This can clog the network and overwhelm the nodes with too much information, particularly in large networks. Also, without control over how messages spread, routing loops may occur, adding more load to the network. Flooding also doesn't scale well in large networks, as the extra work increases a lot as you add more nodes. Finally, it doesn’t adjust to changing network conditions, meaning it can’t optimize how it works based on real-time situations. This makes flooding a poor choice for networks that need to be efficient and scalable. A hop counter is one way to reduce some of the problems with flooding. It limits how many hops a packet can make before it is discarded. This counter is part of the packet’s header and goes down by one every time a node forwards the packet. When it reaches zero, the packet stops moving through the network. While the hop counter helps lessen certain issues of flooding, it doesn’t fix everything. It can help with network congestion and resource waste by stopping packets from endlessly circulating, which cuts down

on repeated transmissions and uses bandwidth more efficiently. It also helps avoid routing loops since it limits how far packets can travel and reduces the chance of them going in circles.

However, the hop counter does not completely eliminate redundant traffic. Even with a limit, different nodes can still send out the same packet multiple times, especially in crowded areas. Moreover, the hop counter doesn't change based on real-time network conditions like congestion or node failures. Scalability is still a problem, especially in large or very active networks, where the hop count might not be enough to prevent too much traffic.

3. Proposed System

We have come up with a new way to solve the flooding algorithm problem. Our method sends out three packets for each message: one original and two copies. This way, we find a good mix between having extra copies and being efficient. With just a few duplicates, the message is more likely to get to where it needs to go, even if some packets are lost or the network is not very reliable. At the same time, we avoid creating too many copies that can slow things down. The original packet moves through the network, while the copies take different paths. This gives us backups without flooding the network with too much traffic. Our method helps to keep things running smoothly and cuts down on unnecessary data compared to old flooding techniques, while still being dependable in big or changing networks. By limiting how many copies we send, we make the network work better, reduce the number of extras, and enable it to handle different amounts of traffic and layouts more effectively.

3.1 System Implementation

In our method, when one of the three packets (the original or a copy) arrives at its destination, we quickly throw away the other two. This way, only one version of the message is used there, which stops extra copies from being processed. It keeps the network running smoothly by preventing delays that can happen when too many copies are sent. By getting rid of the extra packets right after one gets to where it needs to go, we keep the network efficient and avoid too much traffic from identical messages. This method finds a good mix between being dependable and saving resources. It offers backup to make sure the message gets through while also being careful with network use. As a result, we have a better way to spread messages that cuts down on extra traffic and keeps the network from getting overloaded, which is especially useful in large or changing networks where managing resources is important. Our new way of solving the flooding problem helps network efficiency by creating three packets for each

message—one original and two copies. This makes it more likely for messages to get through while reducing the extra copies that usually come with traditional flooding. By removing the copies once one arrives at its destination, we cut down on unnecessary traffic, ease congestion, and save on network resources. This balance between backup and efficiency makes our method very flexible, able to adjust to different network situations, and great for large, changing environments. In the end, this system improves how flooding works, providing a smarter and more resource-saving way to send messages.

3.1.1 Methods

Reinforcement Learning (RL) is a branch of machine learning focused on training agents to make sequences of decisions by interacting with an environment [7][8]. In RL, the agent learns a policy to maximize cumulative rewards by exploring and exploiting different actions. It operates in a framework involving states, actions, and rewards, where the agent observes the current state, takes an action, and receives feedback in the form of a reward and a new state. Over time, the agent uses this feedback to improve its decision-making. RL algorithms, such as Q-Learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO), can tackle complex tasks like game playing, robotics, and autonomous systems. The key challenge in RL lies in balancing exploration (trying new actions to discover their effects) and exploitation (choosing the best-known actions to maximize rewards).

Combining Reinforcement Learning (RL) with the Flooding algorithm can lead to improved results by leveraging the strengths of both approaches. While the Flooding algorithm ensures robust and reliable information dissemination across a network, RL can optimize this process by learning more efficient broadcasting strategies. For instance, an RL agent could learn which nodes are most critical for information propagation, reducing redundant transmissions and minimizing network congestion. This hybrid approach could dynamically adapt to changing network conditions, such as node failures or varying traffic loads, by learning optimal policies for message dissemination. By integrating the adaptability of RL with the simplicity and reliability of Flooding, the system can achieve a balance between efficiency and robustness, particularly in large-scale, dynamic, or resource-constrained networks.

4. Performance Analysis

Reinforcement Learning (RL) with the Flooding algorithm can create a powerful synergy, enhancing efficiency, adaptability, and scalability in communication networks. The Flooding

algorithm ensures robust information dissemination, guaranteeing that messages reach all nodes, even in dynamic or unreliable network environments. However, it often suffers from inefficiencies like message redundancy, excessive bandwidth usage, and increased latency due to unnecessary retransmissions. RL can address these inefficiencies by learning optimal dissemination strategies tailored to specific network conditions.

For example, an RL agent could monitor network states, such as node connectivity, traffic patterns, and energy levels, to dynamically decide which nodes should propagate a message. This adaptive behavior can significantly reduce redundant transmissions while maintaining high delivery reliability. Additionally, RL can optimize the trade-off between energy consumption and coverage, ensuring resource-efficient operation in scenarios like wireless sensor networks or mobile ad hoc networks. By incorporating feedback mechanisms, the RL component can continuously improve its decision-making as the network evolves, making it resilient to disruptions such as node failures or congestion.

This hybrid approach can be applied in various contexts, including disaster recovery networks, Internet of Things (IoT) systems, and autonomous vehicle communication, where efficient and reliable information dissemination is crucial. By combining the robustness of Flooding with the intelligence and adaptability of RL, the system can achieve enhanced performance, reduced resource consumption, and greater resilience in diverse and complex environments.

5. Conclusion

In conclusion, integrating Reinforcement Learning (RL) with the Flooding algorithm offers a compelling solution to enhance the efficiency, adaptability, and robustness of information dissemination in communication networks. While the Flooding algorithm ensures reliability through its comprehensive propagation mechanism, RL introduces intelligence and optimization by learning context-specific strategies to minimize redundancy and resource usage. This hybrid approach leverages the strengths of both methods, enabling dynamic adaptation to changing network conditions and improved scalability in complex environments. The result is a more efficient and resilient system, well-suited for applications ranging from IoT and wireless sensor networks to disaster recovery and autonomous communication systems.

References

[1] **Tse, D., & Viswanath, P. (2005).** *Fundamentals of Wireless Communication.* Cambridge University Press.

[2] **Varga, A., & Hornig, R. (2008).** *The OMNeT++ discrete event simulation system.* In Proceedings of the European Simulation and Modelling Conference (pp. 319–324).

[3] **Boukerche, A., & Li, Y. (2007).** *A survey of routing protocols in mobile ad hoc networks.* In *Mobile Ad Hoc Networking: The Cutting Edge Directions* (pp. 93–128). Wiley-Interscience.

[4] **Zhao, H., & Cao, G. (2009).** *Efficient flooding algorithms for mobile ad hoc networks.* In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM 2009)* (pp. 1665–1673).

[5] **Srinivasan, K., & Srikant, R. (2003).** *On the performance of flooding in wireless networks.* IEEE/ACM Transactions on Networking, 11(5), 743–756.

[6] **Akyildiz, I. F., & Wang, X. (2005).** *A survey on wireless mesh networks.* Computer Networks, 47(4), 445–474.

[7] **Watkins, C. J. C. H., & Dayan, P. (1992).** *Q-learning.* Machine Learning, 8(3–4), 279–292.

[8] **Mnih, V., et al. (2015).** *Human-level control through deep reinforcement learning.* Nature, 518(7540), 529–533.

[9] **Tubaishat, M., & Mizan, M. (2012).** *A survey on wireless sensor network protocols for efficient energy management.* Journal of Communications, 7(3), 139–145.

[10] **Deng, J., & Li, Z. (2010).** *Routing in mobile ad hoc networks: A survey.* IEEE Communications Surveys & Tutorials, 12(3), 333–347.

[11] **Georgiou, A., & Papageorgiou, A. (2014).** *A survey of optimization techniques for wireless networks with reinforcement learning.* Wireless Networks, 20(4), 927–942.

[12] **Xie, L., & Liu, H. (2011).** *Flooding in wireless networks: A survey and research directions.* In *Proceedings of the 2011 IEEE International Conference on Communications (ICC 2011)* (pp. 1–6).

[13] **Zhang, X., & Yang, X. (2016).** *Improving the efficiency of flooding protocols in mobile ad hoc networks using reinforcement learning.* In *Proceedings of the 7th International Conference on Wireless Communications and Signal Processing (WCSP 2016)* (pp. 543–548).

[14] **Cao, G., & Chen, W. (2007).** *Efficient flooding-based routing in wireless sensor networks.* In *Proceedings of the 9th International Symposium on Autonomous Decentralized Systems* (pp. 243–249).

[15] **Li, X., & He, S. (2018).** *Adaptive packet forwarding in mobile ad hoc networks using machine learning.* Journal of Mobile Networks and Applications, 23(2), 234–245.