

Article

# An Innovative Priority Queueing Strategy for Mitigating Traffic Congestion in Complex Networks

Ganhua Wu

Business School, University of Shanghai for Science and Technology, Shanghai 200093, China; godfreywu@189.cn

**Abstract:** Optimizing transportation in both natural and engineered systems, particularly within complex network environments, has become a pivotal area of research. Traditional methods for mitigating congestion primarily focus on routing strategies that utilize first-in-first-out (FIFO) queueing disciplines to determine the processing order of packets in buffer queues. However, these approaches often fail to explore the benefits of incorporating priority mechanisms directly within the routing decision-making processes, leaving significant room for improvement in congestion management. This study introduces an innovative generalized priority queueing (GPQ) strategy, specifically designed as an enhancement to existing FIFO-based routing methods. It is important to note that GPQ is not a new queue scheduling algorithm (e.g., deficit round robin (DRR) or weighted fair queueing (WFQ)), which typically manage multiple queues in broader queue management scenarios. Instead, GPQ integrates a dynamic priority-based mechanism into the routing layer, allowing the routing function to adaptively prioritize packets within a single buffer queue based on network conditions and packet attributes. By focusing on the routing strategy itself, GPQ improves the process of selecting packets for forwarding, thereby optimizing congestion management across the network. The effectiveness of the GPQ strategy is evaluated through extensive simulations on single-layer, two-layer, and dynamic networks. The results demonstrate significant improvements in key performance metrics, such as network throughput and average packet delay, when compared to traditional FIFO-based routing methods. These findings underscore the versatility and robustness of the GPQ strategy, emphasizing its capability to enhance network efficiency across diverse topologies and configurations. By addressing the inherent limitations of FIFO-based routing strategies and proposing a generalized yet scalable enhancement, this study makes a notable contribution to network optimization. The GPQ strategy provides a practical and adaptable solution for improving transportation efficiency in complex networks, bridging the gap between conventional routing techniques and emerging demands for dynamic congestion management.



Academic Editors: Davide Valenti and Sergio Gómez

Received: 10 December 2024

Revised: 19 January 2025

Accepted: 30 January 2025

Published: 2 February 2025

**Citation:** Wu, G. An Innovative Priority Queueing Strategy for Mitigating Traffic Congestion in Complex Networks. *Mathematics* **2025**, *13*, 495. <https://doi.org/10.3390/math13030495>

**Copyright:** © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** generalized priority queueing strategy; routing strategy; traffic congestion; complex networks

**MSC:** 60K30; 82C70

## 1. Introduction

As traffic congestion continues to escalate with rapid economic and social development, network traffic has become a critical area of focus in recent research [1–10]. Network traffic can be conceptualized as a spreading process in a network, in which performance

is intricately influenced by the underlying network topology, resource allocation, and employed routing strategies [11].

Given the inherent challenges in modifying existing infrastructure—often suboptimal in nature—and the complexity of large-scale resource reallocation after network establishment, routing strategies emerge as a promising alternative for improving network performance. Modifications to routing strategies are less costly, easier to implement, and more feasible than changes to network topology or resource distribution. For example, adjustments can be made through software updates in computer network routing algorithms.

Consequently, designing efficient routing strategies has become a central focus of research, yielding a substantial body of work [12–19]. However, many of these strategies rely on the FIFO queueing discipline, often neglecting the potential impact of alternative queueing strategies on network performance.

Queueing strategies play a pivotal role in networking, with applications ranging from traffic management to optimizing performance. For instance, queue scheduling algorithms such as priority queueing (PQ), DRR, and WFQ are commonly employed to manage multiple queues, ensuring fairness, quality of service (QoS), and efficient bandwidth utilization [20,21].

In routing strategies, the rules governing packet dequeuing typically include FIFO, LIFO (last-in-first-out), random selection, and priority service. These rules determine how packets are processed from the router's buffer for transmission.

It is essential to distinguish between the concept of priority service in routing strategies and its role in queue scheduling.

Routing strategies determine which output port a packet is distributed to. In this context, priority service refers to selecting and dequeuing the highest-priority packets from a single queue within the router's buffer. This method prioritizes individual packets based on specific routing policies, ensuring that critical traffic is handled promptly.

Queue scheduling, such as the PQ (Priority Queueing) strategy, focuses on transmitting packets from the current output port to a network link. Unlike routing strategies, queue scheduling manages multiple queues, each assigned a distinct priority level. Packets are dequeued from these queues according to their priority, and within each queue, packets are processed in FIFO order. This approach ensures that higher-priority traffic is serviced first while maintaining order among packets of the same priority level.

Among these strategies, non-FIFO queueing rules, such as LIFO and priority service, have been a focus of recent studies. These investigations explore how alternative queueing strategies within routing algorithms can impact network performance, highlighting the potential for increased efficiency and reduced congestion compared to traditional FIFO-based methods. By optimizing packet dequeuing rules, researchers aim to enhance overall network behavior and service quality.

For example, Tadić et al. investigated web-graph models using the LIFO discipline [22–26]. Wang et al. applied the LIFO queueing policy to network congestion scenarios with limited queue caches. Kim et al. introduced a priority routing strategy based on pre-established packet priorities, noting improvements in congestion states but deterioration under free-flow conditions [27]. Tang and Zhou proposed a strategy in which nodes select the shortest path among unoccupied edges based on an effective distance metric by combining waiting time and path length, which significantly enhances network capacity [28]. Du et al. introduced a shortest-remaining-path-first queueing strategy, prioritising packets by their distance to their destination. This strategy led to improvements in several transportation efficiency metrics despite unchanged network capacity [29]. Zhang et al. presented a dynamic information-based queueing strategy, and they observed notable improvements in traffic indices, such as average travel time and waiting time rates, although network capacity remained unaffected [30].

Wu et al. proposed a shortest-distance-first queueing strategy that notably improved the network throughput and packet arrival rates [31].

Priority service rules exhibit varying degrees of performance enhancement; however, their efficacy is often contingent on specific routing policies. This raises the question of whether a universally applicable priority service rule exists that can consistently improve network transmission performance across all FIFO-based routing strategies. In this study, we propose a GPQ strategy, designed to enhance the performance of all FIFO-based routing approaches.

The remainder of this paper is organized as follows: In Section 2, we present the materials and methods used in this study, followed by simulation results and analysis in Section 3. Finally, the paper concludes with a summary of the findings in Section 4.

## 2. Materials and Methods

### 2.1. The Network Models

#### A. Single-layer network model

Commonly used network models include the SW (small-world), ER (Erdős–Rényi), and BA (Barabási–Albert) models. The SW and ER networks are generated by randomly rewiring edges with a probability  $p$  based on a regular network. In contrast, the BA model involves edges being preferentially connected to nodes with higher degrees.

Many real-world complex networks exhibit a scale-free nature characterized by a power law degree distribution  $P(k) \sim k^{-r}$ . To study the heterogeneous structure of such networks, the BA model proposed by Barabási and Albert provides a framework for generating scale-free networks [32]. The BA model is described as follows:

- (1) Initialization. Beginning with  $m_0$  nodes, each node is fully connected to all other nodes in the initial network.
- (2) Node addition. A new node is introduced to the network at each time step. This new node connects to  $m (\leq m_0)$  existing nodes.
- (3) Attachment mechanism. The probability  $p_i$  that a new node connects to an existing node  $i$  with degree  $k_i$  is given as follows:

$$p_i = \frac{k_i}{\sum_j k_j}, \quad (1)$$

where the summation is the degree of all existing nodes. This model effectively captures the evolution of scale-free networks through its preferential attachment mechanism, in which nodes with higher degrees are more likely to receive new connections.

#### B. Two-layer network model

In real-world scenarios, many networks, including information, transport, and power networks, possess a multi-layer structure in which layers are interconnected through shared links or nodes [33,34]. The two-layer network model is composed of two distinct layers: the physical network (Layer P) and the logical network (Layer L), which contain  $N_P$  and  $N_L$  nodes, respectively. For operational simplicity, we assume that the number of nodes in the logical network is equal to the number in the physical network and that there is a one-to-one correspondence between nodes in the two layers. This correspondence is established randomly, as illustrated in Figure 1.

In Figure 1, it is assumed that both the logical and physical layers use the shortest path routing strategy. In the logical layer, the routing path from node 2 to node 5 is  $RL_{2,5} = \{2, 6, 5\}$ , and the corresponding logical edges are  $L_{2,6}$  and  $L_{6,5}$ . This logical layer path maps the physical layer paths as  $\{2, 3, 6\}$  and  $\{6, 1, 5\}$ . Consequently, the entire path for packet transmission through the physical layer, from source to destination, is

$RP_{2,5} = \{2, 3, 6, 1, 5\}$ . The actual packet transmission in the physical layer is constrained by both the logical and physical layer routes. In this study, it is assumed that the two subnetworks are built by using a BA model of the same size.

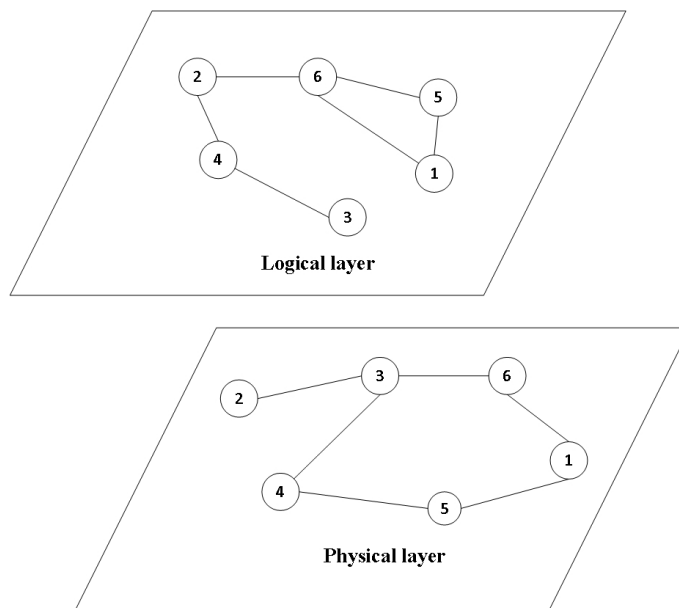


Figure 1. Illustration of the two-layer network model.

C. Dynamic network model

Complex networks, such as social, biological, and technological networks, often exhibit dynamic behavior, with nodes and edges evolving over time. In such dynamic networks, static network models may become insufficient due to real-time topology changes. Thus, new models are required to address these characteristics [35].

Yang et al. proposed a dynamic network model in which  $N$  agents, numbered from 1 to  $N$ , move within a square area of size  $L \times L$  with periodic boundary conditions [36]. Initially, agents are randomly distributed. Each agent’s movement direction is updated randomly at each time step  $\Delta t$ , and speed  $v$  is kept constant for simplicity. All agents have a uniform communication radius  $\alpha$ , and two agents can communicate if the distance between them is less than  $\alpha$ .

2.2. Traffic Model

The network model provides the foundational framework for dynamic traffic management, whereas the traffic model describes the evolving dynamics of traffic flow within this network framework.

A. Traffic model for a single-layer network

The traffic model for a single-layer network is defined as follows: Each node in the network functions both as a host and a router capable of generating and forwarding packets. At each time step,  $R$  packets are generated, with each packet having randomly assigned source and destination nodes. Each node can transmit up to  $C$  packets to its immediate neighbors. The transmission path of each packet is determined by the network routing strategy. When multiple paths are available, a single path is selected randomly. Packets are removed from the network after reaching their destination [37].

The network traffic capacity is characterized by the maximum packet generation rate  $R_c$  at which a phase transition from free-flow to congestion occurs. This transition is quantified by the order parameter, as follows [38]:

$$\eta(R) = \lim_{t \rightarrow \infty} \frac{C \langle \Delta W \rangle}{R \Delta t}, \tag{2}$$

where  $\langle \Delta W \rangle = W(t + \Delta t) - W(t)$  and  $\langle \dots \rangle$  represent the time average over windows of width  $\Delta t$ . Here,  $W(t)$  denotes the number of packets in the network at time  $t$ . For small values of  $R$ , where  $\eta = 0$ , the packet generation rate is balanced by the packet removal rate, thereby preventing congestion. However, as  $R$  increases, more packets fail to be delivered in a timely manner, leading to accumulation at central nodes and, ultimately, traffic congestion.

**B. Traffic model for a two-layer network**

The traffic model for a two-layer network extends the principles of the single-layer network model by incorporating additional complexities. Packets are generated at the logical layer, and source and destination nodes are selected randomly within this layer. Each edge in the logical layer corresponds to a designated path in the physical layer, which handles the actual transmission of packets [39]. For a detailed discussion and illustrative examples, refer to the subsection on the two-layer network model.

**C. Traffic model for a dynamic network**

In dynamic networks, the model must account for the continual evolution of nodes and links, which requires sophisticated approaches that are beyond those used in static network models. This paper introduces a traffic model specifically designed for dynamic networks [36] that is built on the foundational single-layer network model.

At each time step,  $R$  packets are generated, and each packet is assigned a source and destination node at random. The transmission path of each packet is determined by the network routing strategy. Importantly, packets can only be transmitted between nodes when the communication distance between the nodes is less than a specified threshold,  $\alpha$ . After reaching their designated destination, packets are removed from the network.

*2.3. Routing Strategy*

This paper explores enhancements to the FIFO routing strategy by examining several routing approaches: shortest path routing [40], global dynamic routing [41], and greedy routing based on dynamic networks [31]. Each strategy is introduced and analysed in the following sections.

**A. Shortest path routing**

The shortest path routing strategy selects the path with the minimum total length for packet transmission. For a given packet  $m$  with multiple possible paths between the source and destination, the routing strategy aims to minimize the sum of the path weights. Formally, the shortest path  $P(i \rightarrow d_m)$  from node  $i$  to destination node  $d_m$  for packet  $m$  is denoted as follows:

$$P(i \rightarrow d_m) = \min \sum_{j=0}^l (w_j), \tag{3}$$

where  $w_j$  denotes the weight of path segment  $j$ , and  $l$  is the total number of segments in the path.

**B. Global dynamic routing**

The global dynamic routing strategy selects paths based on the sum of the node queue lengths, aiming to minimize congestion. For a packet  $m$  travelling from a source to a destination, this approach determines the path that minimizes the cumulative queue lengths at intermediate nodes. Thus, path  $P(i \rightarrow d_m)$  is given as follows:

$$P(i \rightarrow d_m) = \min \sum_{j=0}^l [1 + n(x_j)], \tag{4}$$

where  $n(x_j)$  represents the queue length of node  $x_j$ , and  $l$  denotes the path length.

C. Greedy routing strategy based on dynamic network

In this strategy, packets are directed to the neighboring node that is closest to the destination. Specifically, for a given packet  $m$  at time  $t$ , the routing strategy selects the neighbor  $j \in J$  that minimizes the Euclidean distance to the packet’s destination. The routing path  $P(i \rightarrow d_m, t)$  is defined as follows:

$$P(i \rightarrow d_m, t) = \min_{j \in J} \sqrt{[x_j(t) - x_d^m(t)]^2 + [y_j(t) - y_d^m(t)]^2}, \tag{5}$$

where  $J$  denotes the set of neighboring nodes of agent  $i$ ,  $x_j(t)$  and  $y_j(t)$  are the coordinates of the neighbor agent  $j$  at time  $t$ , and  $x_d^m(t)$  and  $y_d^m(t)$  are the coordinates of the packet destination at time  $t$ . This strategy directs packets to the nearest neighbor relative to the destination at time  $t$ . Because of the dynamic nature of node movements, the chosen neighbor may not remain optimal in subsequent time steps; therefore, this strategy is inherently locally optimal. Thus, it is referred to as a greedy routing strategy based on dynamic networks.

2.4. The Proposed Queueing Strategy

We propose an innovative GPQ strategy to enhance FIFO-based routing approaches. Consider a network node  $i$  with  $M$  packets queued for delivery. Each packet  $m \in M$  has a designated destination node  $d_m$  and a set of possible delivery paths, denoted as  $All\_Paths(i \rightarrow d_m)$ . The delivery path for a packet is determined by a routing function  $F$ , as follows:

$$P(i \rightarrow d_m) = F(All\_Paths(i \rightarrow d_m)), \tag{6}$$

where  $F$  selects the optimal path for each packet based on the routing strategy. If multiple paths are equivalent, one is selected randomly. The GPQ strategy introduces a priority mechanism to the queueing process. Specifically, it prioritizes packets for transmission by leveraging the same routing function  $F$  used to determine the packet’s delivery path. The priority service policy is expressed as

$$PQ_k(i \rightarrow d_k) = F(P(i \rightarrow d_m) | m \in M), \tag{7}$$

where  $PQ_k$  represents the priority value of packet  $k$ , and the packet with the highest priority is selected for transmission. In cases where multiple packets have the same priority, the packet with the longest waiting time is prioritized for transmission.

The objective of the GPQ strategy is to dynamically adjust the packet transmission order based on real-time network conditions. By incorporating the routing comparison function  $F$ , the strategy prioritizes packets to reduce network congestion, improve overall transmission efficiency, and ensure fairness by preventing excessive delays for low-priority packets. When the comparison function  $F$  is set to the shortest path and the shortest buffer queue, respectively, our strategy corresponds to the shortest-remaining-path-first queueing strategy proposed by Du et al. [29] and the dynamic information-based queueing strategy proposed by Zhang et al. [30] This demonstrates the versatility of our strategy.

The detailed process of sending  $C$  packets per unit time from each node is as shown in Algorithm 1.

**Algorithm 1:** GPQ Strategy for Packet Transmission

---

```

Input:
Sending capacity  $C$ ;
Routing strategy and comparison function  $F$ ;
Queue size  $M$ ;
Maximum waiting time  $maxT$ .
Output:
 $C$  packets sent according to the adaptive priority.
// sending capacity is no less than queue size
if  $C \geq M$  then
    Send all packets in the queue;
    Exit;
end
else
    // FIFO-based routing strategy
    if FIFO then
        for  $i \leftarrow 1$  to  $C$  do
            Remove the first packet from the queue;
            Determine sending path using routing strategy;
            Send the packet;
        end
    end
    // GPQ Strategy
    else
        Initialize  $Q \leftarrow 0$  // Number of packets sent using GPQ;
        for  $i \leftarrow 1$  to  $M$  do
            Retrieve waiting time and sending path for packet  $m_i$ ;
            if  $waiting\_time(m_i) > maxT$  then
                Send packet  $m_i$  immediately;
                Increment  $Q$ ;
                if  $Q = C$  then
                    Exit;
                end
            end
            else if  $destination(m_i)$  is a neighbor then
                Send packet  $m_i$  immediately;
                Increment  $Q$ ;
                if  $Q = C$  then
                    Exit;
                end
            end
            else
                Record the priority value of  $m_i$ ;
            end
        end
         $J = C - Q$ ; // Calculate remaining packets to send;
        if  $J > 0$  then
            Select  $J$  packets with the highest priority  $F(m_i)$ ;
            if priorities are the same then
                Choose the packet with the longer waiting time first;
            end
            Send the selected packets;
        end
    end
end

```

---

The core innovation of this strategy lies in aligning the queueing policy with the routing function. By using the same  $F$  function for both path selection and priority determination, the GPQ strategy ensures consistency and compatibility with the underlying routing policy.

### 2.5. Comparison of Time and Space Complexity for FIFO-Based Routing Strategy and GPQ Strategy

To evaluate the efficiency of the proposed GPQ strategy, we compare its time and space complexity with the traditional FIFO-based routing strategy. The comparison is organized as follows.

#### 2.5.1. Time Complexity

FIFO-based routing strategy: In this strategy, packets are processed sequentially. For each packet, the following operations are performed:

1. Remove packet from the queue: This operation takes constant time,  $O(1)$ .
2. Determine the sending path: The routing strategy determines the path, with a complexity of  $O(R)$ .
3. Send the packet: This operation also takes constant time,  $O(1)$ .

For  $C$  packets sent per unit time, the total time complexity is

$$O(C \times R)$$

GPQ Strategy: The GPQ strategy involves additional steps for priority-based processing:

1. Initialize counter  $Q$ : This takes constant time,  $O(1)$ .
2. Traverse all  $M$  packets in the queue: For each packet, perform the following:
  - Retrieve waiting time and routing path:  $O(1)$ .
  - Call the routing strategy to calculate the path:  $O(R)$ .
  - Evaluate the conditions for immediate sending:  $O(1)$ .
  - Record packet priority:  $O(1)$ .

Total complexity for  $M$  packets:  $O(M \times R)$ .

3. Select and send  $J$  packets with the highest priority:
  - Sort  $M$  packets (e.g., using merge sort):  $O(M \log M)$ .
  - Select top  $J$  packets (usually  $J \leq M$ ):  $O(J)$ .

The total time complexity of the GPQ strategy is

$$O(M \times R + M \log M)$$

If  $M \times R$  dominates  $M \log M$ , the complexity simplifies to  $O(M \times R)$ .

#### 2.5.2. Space Complexity

FIFO-based routing strategy: The space complexity in the FIFO-based routing strategy includes the following:

1. Queue storage: Space for  $M$  packets,  $O(M)$ .
2. Temporary space for routing: Space required for path calculations,  $O(R)$ .

Total space complexity:

$$O(M + R)$$

GPQ Strategy: The GPQ strategy requires additional space for priority handling:

1. Queue storage: Space for  $M$  packets,  $O(M)$ .
2. Priority storage: Space to store priority values for  $M$  packets,  $O(M)$ .
3. Temporary space for routing: Similar to FIFO,  $O(R)$ .

Total space complexity:

$$O(M + R)$$



### 2.5.3. Summary

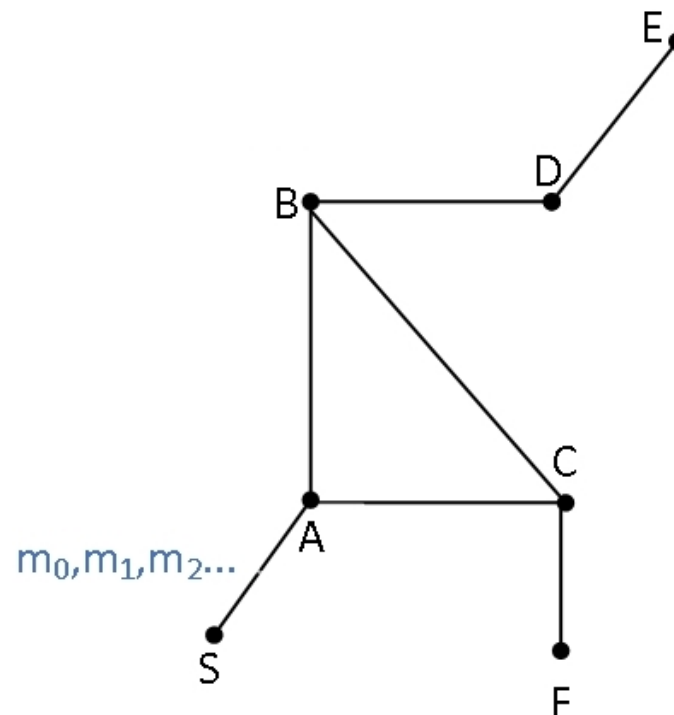
As shown in Table 1, a comparison of the time and space complexity between FIFO-based routing and GPQ strategies is presented. While the FIFO-based routing strategy is simple and efficient in terms of time complexity, it lacks the flexibility required for dynamic network conditions. On the other hand, the GPQ strategy, though more adaptable, introduces additional computational overhead due to its sorting and prioritization processes. However, this increase in time complexity is justified by its superior congestion management and overall network performance, particularly in environments with sufficient computational resources. The computational overhead of the GPQ strategy is further evaluated through detailed experimental simulations.

**Table 1.** Comparison of time and space complexity for FIFO-based routing and GPQ strategies.

Aspect	FIFO-Based Routing Strategy	GPQ Strategy
Time Complexity	$O(C \times R)$	$O(M \times R + M \log M)$
Space Complexity	$O(M + R)$	$O(M + R)$

### 2.6. GPQ Application Analysis

The key feature of the GPQ algorithm is dynamically adjusting the packet sending order based on the routing comparison function  $F$ . This approach is particularly significant in real-world scenarios such as vehicular traffic networks and air transport networks. For instance, in a vehicular traffic network, as shown in Figure 2, the vehicles  $m$  represent the flow of traffic, and the traffic intersections serve as nodes. At each time step, the capacity of each intersection is limited. When traffic volume is high, vehicles must wait at the intersection. Suppose vehicle  $m_0$  is heading to node E, and vehicle  $m_2$  is heading to node F. The routing algorithm is a global dynamic routing strategy, where vehicles choose the least congested paths.



**Figure 2.** Example diagram of priority algorithm.

The route for  $m_0$  is  $A \rightarrow B \rightarrow D \rightarrow E$ , while the route for  $m_2$  is  $A \rightarrow C \rightarrow F$ . Even if the  $B \rightarrow D$  path is heavily congested, under the FIFO strategy,  $m_0$  will still be dispatched first onto the  $A \rightarrow B$  path. After reaching the  $A \rightarrow B$  path,  $m_0$  must continue waiting before proceeding to its destination.

In contrast, under the GPQ strategy, if the  $A \rightarrow F$  path is less congested and allows  $m_2$  to reach its destination more quickly,  $m_2$  will be prioritized for transmission along the  $A \rightarrow C$  path. Once  $m_2$  reaches its destination, the overall network traffic will decrease accordingly. Applying this logic to other nodes in the network will significantly reduce overall congestion compared to the FIFO algorithm, thereby alleviating network traffic bottlenecks more effectively.

Although this algorithm may seem unfair to  $m_0$ , as it delays  $m_0$  in favor of  $m_2$ , the overall network congestion will decrease substantially, which in turn will reduce congestion on the path taken by  $m_0$ . This allows  $m_0$  to reach its destination more smoothly. Additionally, to ensure that  $m_0$  does not wait indefinitely due to path congestion, the algorithm will prioritize sending  $m_0$  if its waiting time exceeds a threshold, thus mitigating unfairness to individual nodes.

By incorporating this adaptive priority mechanism, the GPQ strategy significantly improves network transmission efficiency compared to traditional FIFO methods while maintaining broad applicability across various routing paradigms and network configurations.

### 2.7. Routing Strategy vs. Queue Scheduling Algorithms

To clarify the distinction between priority service in routing strategies and queue scheduling algorithms such as PQ, DRR, and WFQ, we analyze the following example illustrated in Figure 3.

1. **Determining the Output Path:**  
Data flows from node  $S$  to node  $A$  (nodes can be computers, routers, etc.). Upon receiving packets, node  $A$  first determines the output path for these packets. Node  $A$  may have multiple output paths, such as  $A1$  and  $A2$  shown in Figure 3. This decision is made by the routing strategy, which examines the destination addresses of the packets and considers the network's congestion status to select the appropriate path. For example, the routing strategy determines whether to send the packets to node  $B$  through port  $A1$  or to node  $C$  through port  $A2$ .
2. **Buffering Excess Traffic:**  
If a significant amount of traffic flows from  $S$  to  $A$ , the routing strategy at node  $A$  may be unable to process all packets promptly. These unprocessed packets are temporarily stored in a buffer queue  $Q$  (e.g.,  $m_1, m_2, \dots$ ).
3. **Routing Strategy for Packet Selection:**  
Traditional routing strategies often adopt a FIFO rule, where packets in queue  $Q$  are processed in the order they arrive. However, the proposed strategy employs a routing comparison function  $F$  to select and prioritize specific packets for transmission, ensuring more efficient handling of critical traffic.
4. **Sending Packets to Output Ports:**  
Once the routing strategy determines the transmission path, packets are sent to the output ports ( $A1$  or  $A2$ ).
5. **Queue Scheduling Algorithms:**  
If there is heavy traffic at the output ports, packets must queue and wait for processing by the scheduling algorithm.  
Each port (e.g.,  $A1$  or  $A2$ ) often manages multiple priority queues. Packets are assigned priority levels, typically based on their QoS requirements, and placed into corresponding queues. Scheduling algorithms such as PQ, DRR, and WFQ are used

to determine the order in which packets are dequeued from these queues. These algorithms ensure fairness, QoS, and efficient bandwidth utilization. However, within each individual queue, packets are usually processed in FIFO order by the scheduling algorithm.

This discussion focuses on improvements to routing strategies and does not address queue scheduling algorithms, which operate at the output ports to manage multiple queues. By distinguishing these layers, the interaction between routing decisions and queue scheduling becomes clear, illustrating how traffic is efficiently handled across the network.

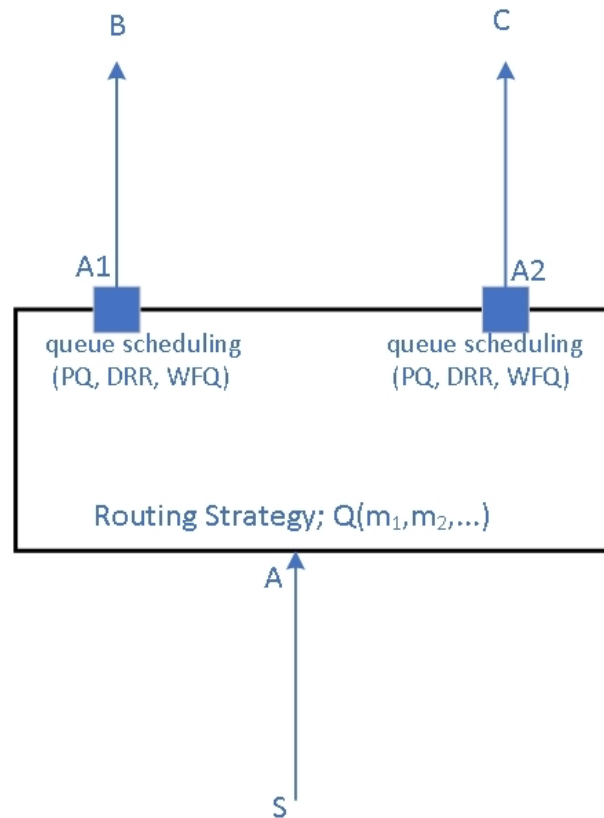


Figure 3. Routing strategy vs. queue scheduling algorithms.

### 3. Simulation Results and Analysis

This section presents an evaluation of the proposed queueing strategy across various network scenarios, including single-layer, multi-layer, and dynamic networks. The performance of the proposed approach is compared against that of the FIFO queueing rule to assess its impact on network transmission performance.

#### 3.1. Simulation on a Single-Layer Network

We first validate the efficiency of the proposed algorithm in both the small-world (SW) and Erdős–Rényi (ER) networks. The routing strategy used is the shortest path routing, with a network size of 400 nodes, an average degree of 4, and a rewiring probability of 0.3 for the SW model. The data points represent averages over 20 simulations. The error bars are omitted because the error values are minimal. We analyze the relationship between the parameter  $\eta$  and the network-generated packet  $R$ , as shown in Figure 4.

From the figure, it can be observed that as  $R$  increases, both the SW and ER models under SP routing quickly enter the congestion phase, whereas the GPQ strategy enters the congestion phase much more slowly. Under SP routing, the congestion thresholds for the SW and ER models are 50 and 40, respectively, while under the GPQ strategy, the congestion

thresholds are 500 and 400. This clearly shows that the GPQ strategy significantly improves the network’s transmission capacity.

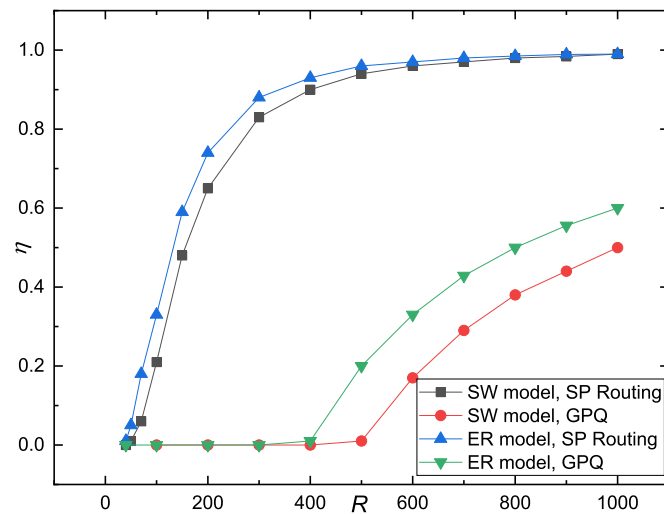


Figure 4. Order parameter  $\eta$  versus packet generation rate  $R$  in SW and ER models.

We then evaluate the proposed queueing policy in a single-layer network employing global dynamic routing. In the proposed priority service policy, the  $F$  function is defined as  $\min \sum_{j=0}^l [1 + n(x_j)]$ , where  $n(x_j)$  represents the queue length of node  $x_j$ . For any given packet  $m$ , path  $P(i \rightarrow d_m)$  with the minimum sum of cache queue lengths, as determined by the  $F$  function, is selected. The path  $PQ_k(i \rightarrow d_k)$  is subsequently selected from the set of paths  $P(i \rightarrow d_m) | m \in M$ , with a preference for the path that exhibits the smallest cumulative cache queue length among all available paths. The packet  $k$  is then preferentially transmitted. This selection process is executed iteratively at each time step, with the node continuing to select packets for transmission until its maximum sending capacity is reached.

Simulations were conducted to compare the transmission performance of the FIFO and GPQ policies under global dynamic routing conditions. The experiments were conducted on a scale-free network with the following parameters: network size  $N = 400$ , initial number of nodes  $N_0 = 5$ , average degree  $\langle k \rangle = 4$ , sending capacity  $C = 1$ , and network runtime  $T = 10,000$ . Figure 5 illustrates the variation in the order parameter  $\eta$  as a function of the packet generation rate  $R$ .

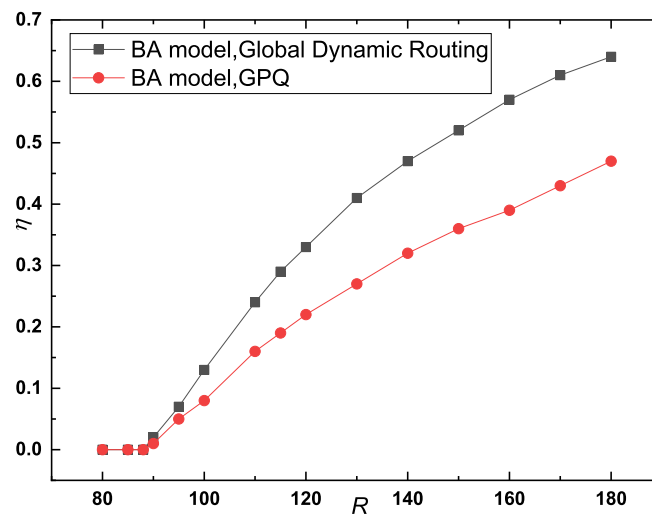


Figure 5. Order parameter  $\eta$  versus packet generation rate  $R$ .

It is evident from this figure that the critical packet generation rate  $R_c$  is similar for both queueing strategies. However, when  $R > R_c$ , the value of  $\eta$  in FIFO is greater than that in the GPQ strategy, with the difference becoming more pronounced as  $R$  increases. This suggests that under congested conditions, the GPQ strategy is more effective at alleviating network congestion than the FIFO strategy, with the performance disparity becoming more significant at higher values of  $R$ .

We also examined the relationship between packet arrival rate  $A$  and packet generation rate  $R$ . The rate  $A$  is defined as the ratio of the number of arrived packets to the number of generated packets, as follows [42]:  $A = N_{arrive} / N_{create}$ . Here,  $N_{arrive}$  represents the number of packets that have arrived, and  $N_{create}$  denotes the number of generated packets. The rate  $A$  serves as an indicator of system throughput. In a free-flowing state, in which packets are delivered to their destination without delay,  $A$  approaches 1. In contrast, in a congested state, in which packets accumulate in the network,  $A$  is less than 1.

Figure 6 shows that  $A$  is equal to 1 for both queueing strategies when  $R < R_c$ , indicating that all packets reach their destination without delay. For  $R > R_c$ ,  $A$  decreases gradually in both strategies, and a more gradual decrease is observed with the GPQ strategy. This result is consistent with the findings presented in Figure 5.

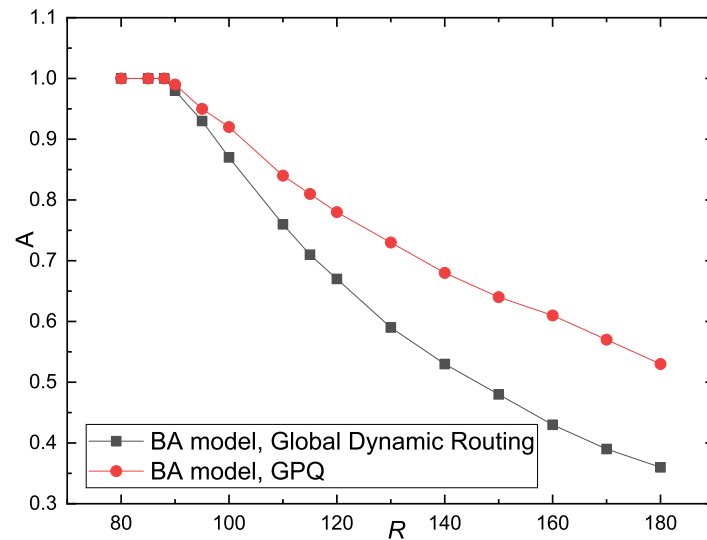


Figure 6. Packet arrival rate  $A$  versus packet generation rate  $R$ .

### 3.2. Simulation on a Two-Layer Network

Next, we compared the performance of the FIFO and GPQ queueing strategies in a two-layer network via shortest-path routing. In our priority service policy, function  $F$  is defined as  $\min \sum_{j=0}^l (w_j)$ , where  $w_j$  represents the weight of path segment  $j$ . For any packet  $m$ , path  $P(i \rightarrow d_m)$  with the minimal path length, as determined by function  $F$ , is selected. The queueing policy selects the path with the smallest length according to the  $F$  function for all  $P(i \rightarrow d_m) | m \in M$ . Assuming  $PQ_k$ , packet  $k$  is preferentially sent. We utilized a two-layer network model characterized by scale-free networks in both the logical and physical layers, with routing performed via shortest path algorithms. Nodes across the two layers were matched randomly. The network parameters are as follows: node size  $N = 400$ , initial number of nodes  $N_0 = 7$ , average degree  $\langle k \rangle = 6$ , sending capacity  $C = 3$ , and network runtime  $T = 10,000$ . Note that node cache queue lengths are considered unlimited. The simulation results were averaged over 20 networks of equal size.

Figure 7 illustrates the relationship between order parameter  $\eta$  and packet generation rate  $R$ . The results shown in Figure 7 are consistent with those presented in Figure 5, indicating that the critical packet generation rate  $R_c$  is approximately 6 for both queueing

strategies. When  $R > R_c$ , the order parameter  $\eta$  in FIFO is greater than that in the GPQ strategy, suggesting that the GPQ strategy effectively reduces network congestion compared with FIFO, with the benefit becoming more significant as  $R$  increases.

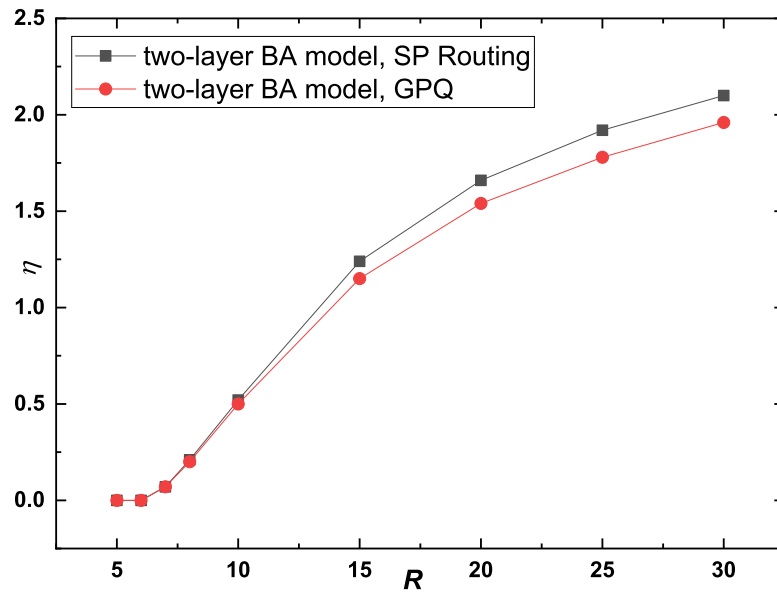


Figure 7. Order parameter  $\eta$  versus packet generation rate  $R$ .

Figure 8 illustrates the relationship between the rate of waiting time to travel time  $W$  and the packet generation rate  $R$ . The rate  $W$  is defined as follows [29]:

$$W = \left( \sum_{i=1}^{N_{arrive}} T_{i-wait} / T_{i-travel} \right) / N_{arrive}, \tag{8}$$

where  $T_{i-travel}$  represents the travel time of packet  $i$ ,  $T_{i-wait}$  denotes the waiting time of packet  $i$ , and  $N_{arrive}$  is the number of packets that have arrived. In general,  $W$  is an indicator of customer satisfaction. In various systems, such as airline operations and the Worldwide Web, users may exhibit increased impatience if the  $W$  is high.

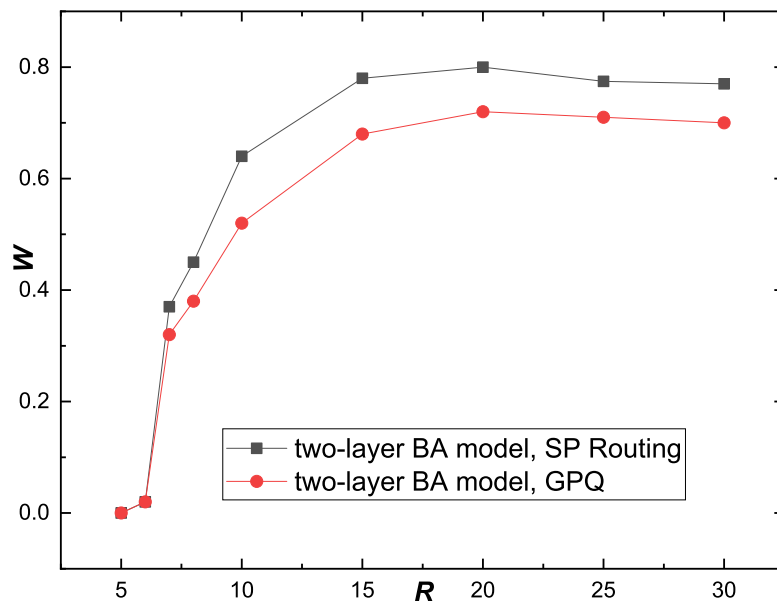


Figure 8. Rate of waiting time to travel time  $W$  versus packet generation rate  $R$ .

When  $R < R_c$ ,  $W$  is zero in both queueing strategies, indicating that packets do not experience any waiting in the free-flow state. However, when  $R > R_c$ ,  $W$  increases rapidly for both queueing strategies. Notably, in the GPQ strategy, the increase in  $W$  is more gradual, and this effect becomes more pronounced at higher values of  $R$ . This observation suggests that the GPQ strategy is more effective at reducing waiting times under congested conditions, particularly at higher values of  $R$ .

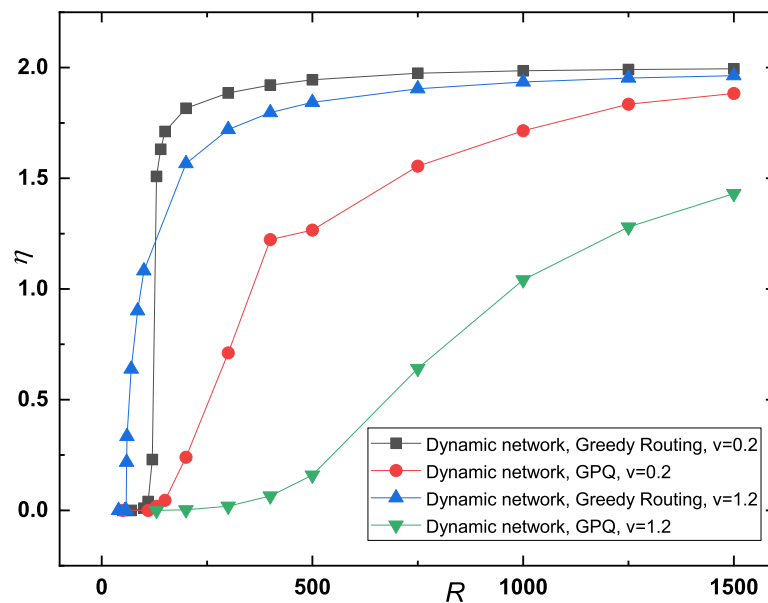
### 3.3. Simulation on a Dynamic Network

Finally, we examined the performance of the proposed priority service policy in a dynamic network employing a greedy routing strategy. In this context, the function  $F$  is defined as follows:

$$F = \min_{j \in J} \sqrt{[x_j(t) - x_d^m(t)]^2 + [y_j(t) - y_d^m(t)]^2}, \tag{9}$$

where  $J$  represents the set of neighboring nodes. For any packet  $m$ , path  $P(i \rightarrow d_m)$  with the minimum path length is selected on the basis of the  $F$  function. The queueing policy then determines the path with the smallest length among all possible paths  $P(i \rightarrow d_m) | m \in M$ , denoted as  $PQ_k$ . Consequently, packet  $k$  associated with  $PQ_k$  is given preferential treatment for transmission.

Figure 9 shows order parameter  $\eta$  versus packet generation rate  $R$  for different moving speeds  $v$  with  $\alpha = 1$ . The critical value  $R_c$  marks the transition from a free-flow state to congestion over a sharp interval of  $R$ . The threshold  $R_c$  of the GPQ strategy is greater than that of the greedy routing strategy, especially at higher speeds. For example,  $R_c$  for the greedy routing strategy is approximately 59 at  $v = 1.2$ , whereas for the GPQ strategy, it is approximately 300, indicating a substantial improvement in network throughput.



**Figure 9.** Order parameter  $\eta$  versus  $R$ . The network is composed of 800 agents, which are arranged in a square region of  $L \times L = 10 \times 10$  with a communication radius  $\alpha = 1$ , and each agent has a delivery capacity of  $C = 2$ .

Figure 10 illustrates the relationship between packet arrival rates  $A$  and  $R$ . In the free-flow state,  $A$  approaches 1 for both strategies. However, in the congested state, packet arrival rate  $A$  decreased more rapidly in the greedy routing strategy than in the GPQ strategy. For example, at  $R = 1500$ , in the greedy routing strategy,  $A$  approaches 0, whereas for the GPQ strategy, it remains approximately 0.06 and 0.28 at  $v = 0.2$  and  $v = 1.2$ , respectively.

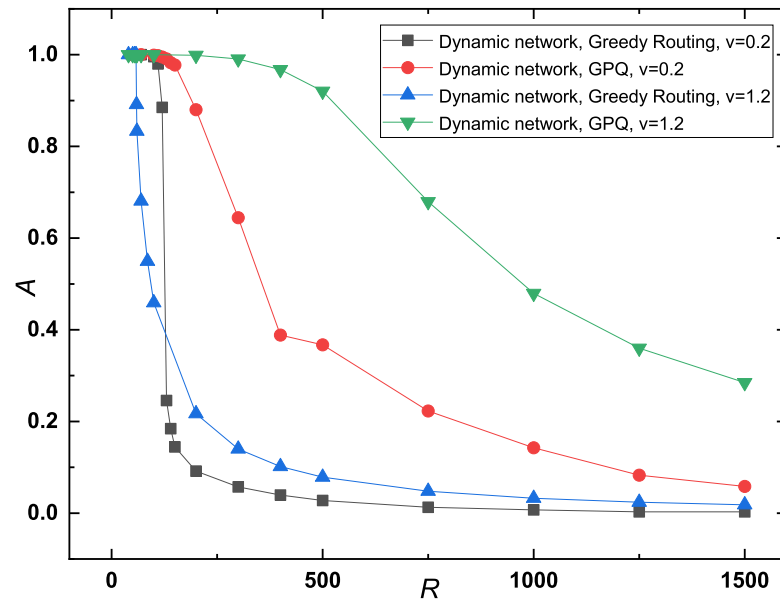


Figure 10. Packet arrival rate  $A$  as a function of packet generation rate  $R$ .

Figure 11 illustrates the relationship between the rate of waiting time to travel time  $W$  and packet generation rate  $R$ . When  $R$  is low, the rate  $W$  is minimal for both routing strategies. However, as  $R$  increases,  $W$  rises sharply, with the greedy routing strategy exhibiting a more pronounced growth compared to the GPQ approach.

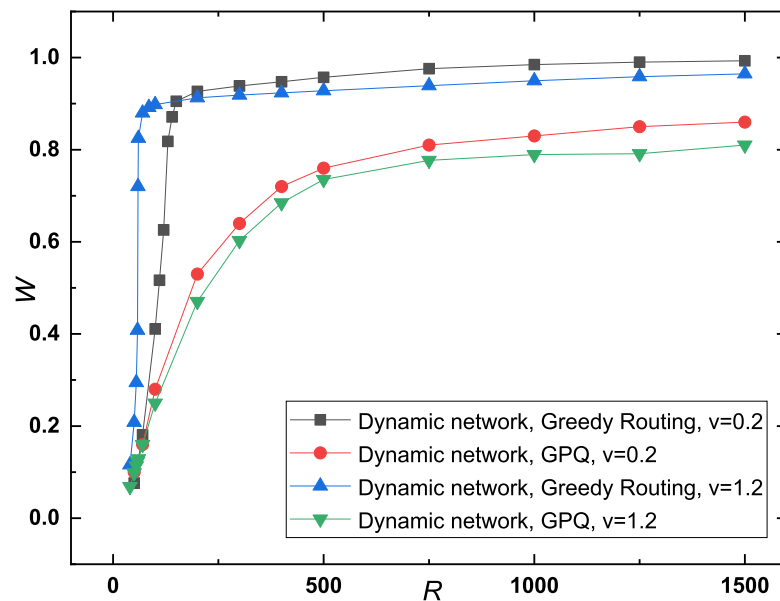


Figure 11. Rate of waiting time to travel time  $W$  as a function of packet generation rate  $R$ .

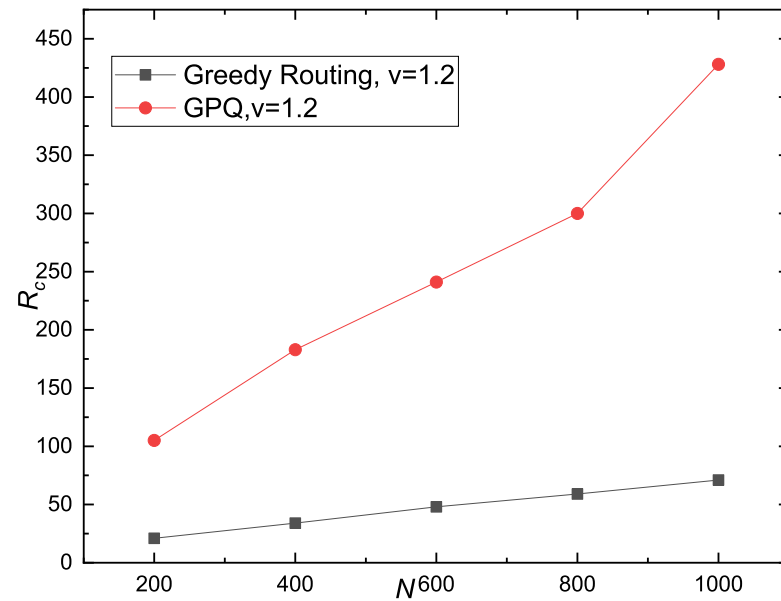
### 3.4. Algorithm Performance Under Different Network Sizes

We continue to validate the robustness of the proposed algorithm. We analyze the results of the algorithm under different network sizes, using a dynamic network and greedy routing. Figure 12 shows the relationship between the network congestion threshold  $R_c$  and network size.

From the figure, it can be seen that as the network size increases, the congestion threshold  $R_c$  for both algorithms increases as well, with the GPQ algorithm exhibiting faster growth in  $R_c$ . For all values of  $N$ , the  $R_c$  for the GPQ strategy is higher than that for greedy routing, and as  $N$  grows, the gap between them gradually widens. This



indicates that as the network size increases, the advantages of the GPQ algorithm become more pronounced.



**Figure 12.** Change in critical packet generation rate  $R_c$  with network size.

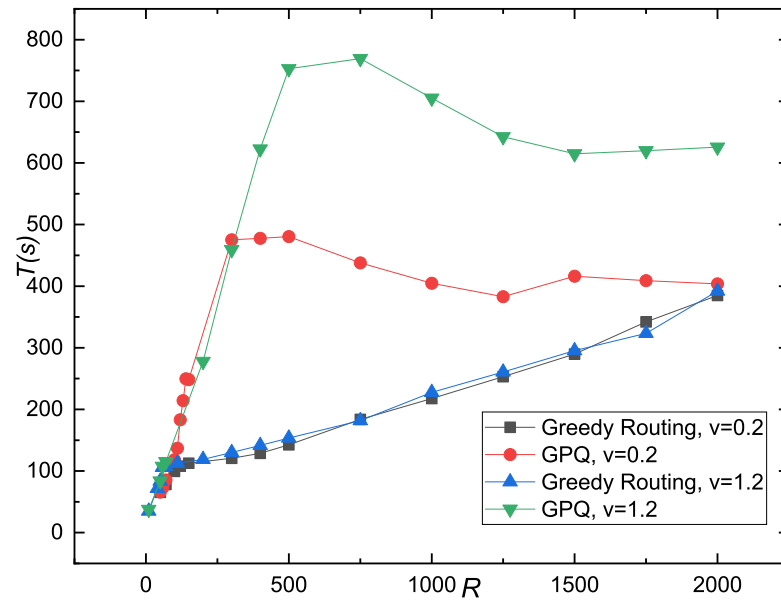
These results clearly indicate that compared with the FIFO-based routing strategy, the GPQ strategy improves network capacity, achieves a higher packet arrival rate, and reduces the rate of waiting time to travel time.

### 3.5. Empirical Overhead

Compared to the FIFO strategy, the GPQ algorithm mainly adds the computation for prioritizing the sending of packets. We evaluated the total overhead across all nodes in the network, with the computation time being the entire simulation period. The simulation was conducted in a dynamic network with a scale of 800. The calculations were performed on a standard server with an Intel Core i7-1360P CPU and 16 GB of memory (HP Inc., Palo Alto, CA, USA).

As shown in Figure 13, when the packet generation rate is low, the overhead of both the FIFO and GPQ algorithms is similar. This is because the network is in a steady state, and packets in the nodes can be transmitted in a timely manner, meaning that the GPQ strategy does not require additional computation. As  $R$  increases, the overhead of GPQ begins to rise rapidly, while the overhead of FIFO increases much more slowly. This is because as the packet generation rate increases, the network becomes more congested, which in turn increases the processing time.

When  $R$  is large, the overhead of the GPQ strategy stabilizes and then slowly decreases, while the FIFO overhead continues to increase. At very large values of  $R$ , when the node mobility rate  $v = 0.2$ , the overhead of GPQ approaches that of FIFO. This is because, when  $R$  is large, there are many packets waiting in the buffer. On one hand, some packets are directly prioritized for transmission because their waiting time exceeds the threshold. On the other hand, the probability that the destination address is a neighboring node increases, which in turn increases the likelihood of direct transmission. This reduces the number of times the GPQ strategy has to traverse the entire buffer queue, lowering execution time.



**Figure 13.** Relationship between execution time and  $R$ .

As seen in the figure, in the worst case, the additional node execution time of GPQ compared to FIFO is approximately 0.7 s. These results were obtained on a standard server. If the server has more powerful performance or if hardware implementation is used, the overhead of the GPQ strategy will be significantly reduced. Given the improvements the algorithm brings to network transmission performance, these overheads are well worth the cost.

#### 4. Conclusions and Discussion

This study presents a novel GPQ strategy designed to enhance the performance of FIFO-based routing approaches in complex networks. Through extensive empirical analysis on single-layer, two-layer, and dynamic network configurations, the GPQ strategy demonstrates significant improvements in key performance metrics, including network throughput, arrival rate, and waiting time, across various scenarios. These results validate the effectiveness of the proposed method in addressing congestion challenges inherent in diverse network environments.

The primary contribution of this work lies in the development of a universally adaptable queueing strategy that aligns seamlessly with underlying routing policies. By incorporating dynamic packet prioritization into the forwarding process, the GPQ strategy introduces a level of adaptability and optimization that has been overlooked in traditional FIFO-based methods. Additionally, its flexibility and scalability make it a promising solution for enhancing network transmission performance in applications ranging from engineered systems to natural networks.

Despite these benefits, we acknowledge that the GPQ strategy has limitations in certain scenarios:

- **High Computational Overhead:** In networks with extremely high packet generation rates or limited computational resources, the additional processing required for priority calculations and sorting may lead to delays.
- **Dynamic Network Environments:** In highly dynamic networks with rapidly changing topologies, the routing comparison function  $F$  might not adapt quickly enough to real-time changes, potentially reducing its effectiveness.
- **Compatibility with Specific Network Types:** The GPQ strategy is primarily designed for general-purpose networks. Its compatibility with highly specialized network types,

such as those with strict deterministic timing requirements (e.g., industrial control systems), may require further adaptation and testing.

The GPQ strategy also involves trade-offs between optimizing network performance and ensuring fairness. By dynamically prioritizing packets, the strategy may introduce delays for low-priority packets. To mitigate this, the GPQ algorithm incorporates a fairness mechanism, such as prioritizing packets with longer waiting times and introducing threshold-based rules to prevent indefinite delays. These measures balance efficiency and fairness, ensuring robust performance across diverse network scenarios.

Additionally, we have explored the potential integration of GPQ with other routing methods, such as greedy or shortest path routing. Hybrid strategies could combine the global optimization capabilities of GPQ with the simplicity and efficiency of traditional approaches, further enhancing network performance. This avenue highlights the versatility of the GPQ strategy and serves as a promising direction for future research.

We also discuss the potential contributions of the GPQ strategy to sustainability and energy efficiency in network systems. By dynamically prioritizing packets and reducing congestion, GPQ decreases the time packets spend in the network, thereby reducing the overall energy required for data transmission. Furthermore, the strategy optimizes resource utilization by balancing traffic loads, aligning with broader sustainability goals and contributing to greener network infrastructures.

Future work could involve performance comparisons with other priority-based strategies to further validate the effectiveness of the GPQ strategy. Additionally, optimizing the computational cost of GPQ, particularly through parallel processing or more efficient prioritization techniques, remains a key area of research. Another important direction is to explore how to appropriately set the maximum waiting time threshold to achieve an optimal balance between transmission efficiency and fairness. Finally, integrating machine learning or predictive analytics to dynamically adjust priorities based on real-time network conditions could further enhance the adaptability and scalability of the GPQ strategy.

In summary, the GPQ strategy offers a robust framework for improving network performance, balancing adaptability, fairness, and sustainability. Its flexibility across diverse network topologies and routing paradigms underscores its potential as a foundational tool for addressing modern network challenges.

**Funding:** This research was funded by the University-Industry Collaborative Education Program (Project No. 201901134025).

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Toroczkai, Z.; Bessler, K.E. Jamming is limited in scale-free systems. *Nature* **2004**, *428*, 716. [[CrossRef](#)] [[PubMed](#)]
2. Menezes, M.A.D.; Barabási, A.L. Fluctuations in network dynamics. *Phys. Rev. Lett.* **2004**, *92*, 028701. [[CrossRef](#)]
3. Zheng, M.; Ruan, Z.Y.; Tang, M.; Do, Y.H.; Liu, Z.H. Influence of periodic traffic congestion on epidemic spreading. *Int. J. Mod. Phys. C* **2016**, *27*, 1650048. [[CrossRef](#)]
4. Afrin, T.; Yodo, N. A survey of road traffic congestion measures towards a sustainable and resilient transportation system. *Sustainability* **2020**, *12*, 4660. [[CrossRef](#)]
5. Sánchez González, S.; Bedoya-Maya, F.; Calatayud, A. Understanding the effect of traffic congestion on accidents using big data. *Sustainability* **2021**, *13*, 7500. [[CrossRef](#)]
6. Ma, J.L.; Zhang, J.F.; Zhang, Y.Q. Effective gravitation path routing strategy on scale-free networks. *IEEE Access* **2021**, *9*, 96031. [[CrossRef](#)]
7. Zhang, M.Y.; Huang, T.; Guo, Z.X.; He, Z. Complex-network-based traffic network analysis and dynamics: A comprehensive review. *Physica A* **2022**, *607*, 128063. [[CrossRef](#)]

8. Bokaba, T.; Doorsamy, W.; Paul, B.S. A comparative study of ensemble models for predicting road traffic congestion. *Appl. Sci.* **2022**, *12*, 1337. [[CrossRef](#)]
9. Yin, R.; Song, X. Mitigation strategy of cascading failures in urban traffic congestion based on complex networks. *Int. J. Mod. Phys. C* **2023**, *34*, 2350022. [[CrossRef](#)]
10. Wu, G.H.; Yang, H.J. Traffic systems recovery from complete congestion by the targeted dropping of packets. *Mod. Phys. Lett. B* **2019**, *33*, 1950096. [[CrossRef](#)]
11. Chen, S.Y.; Huang, W.; Cattani, C.; Altieri, G. Traffic dynamics on complex networks: A survey. *Math. Probl. Eng.* **2012**, *2012*, 732698. [[CrossRef](#)]
12. Amin, A.; Tareen, W.U.K.; Usman, M.; Ali, H.; Bari, I.; Horan, B.; Mekhilef, S.; Asif, M.; Ahmed, S.; Mahmood, A. A review of optimal charging strategy for electric vehicles under dynamic pricing schemes in the distribution charging network. *Sustainability* **2020**, *12*, 10160. [[CrossRef](#)]
13. Ma, J.L.; Wei, J.; Tang, X.; Zhao, X. An improved efficient routing strategy on two-layer networks. *Pramana* **2022**, *96*, 95. [[CrossRef](#)]
14. López-Rourich, M.A.; Rodríguez-Pérez, F.J. Efficient data transfer by evaluating closeness centrality for dynamic social complex network-inspired routing. *Appl. Sci.* **2023**, *13*, 10766. [[CrossRef](#)]
15. Solé-Ribalta, A.; Gómez, S.; Arenas, A. Congestion induced by the structure of multiplex networks. *Phys. Rev. Lett.* **2016**, *116*, 108701. [[CrossRef](#)] [[PubMed](#)]
16. Mishra, A.; Wen, T.; Cheong, K.H. Efficient traffic management in networks with limited resources: The switching routing strategy. *Chaos Soliton Fract.* **2024**, *181*, 114658. [[CrossRef](#)]
17. Tadić, B.; Thurner, S. Search and topology aspects in transport on scale-free networks. *Physica A* **2005**, *346*, 183–190. [[CrossRef](#)]
18. Yang, X.; Liu, M.; Wang, X.; Hu, B.; Liu, M.; Wang, X. Ship Network Traffic Engineering Based on Reinforcement Learning. *Electronics* **2024**, *13*, 1710. [[CrossRef](#)]
19. Bai, J.; Sun, J.; Wang, Z.; Zhao, X.; Wen, A.; Zhang, C.; Zhang, J. An adaptive intelligent routing algorithm based on deep reinforcement learning. *Comput. Commun.* **2024**, *216*, 195–208. [[CrossRef](#)]
20. Semeria, C. Supporting differentiated service classes: Queue scheduling disciplines. *Juniper Netw.* **2001**, *27*, 11–14.
21. Abdollahi, S.; Deldari, A.; Asadi, H.; Montazerolghaem, A.; Mazinani, S.M. Flow-aware forwarding in SDN datacenters using a knapsack-PSO-based solution. *IEEE Trans. Netw. Serv.* **2021**, *18*, 2902–2914. [[CrossRef](#)]
22. Tadić, B.; Thurner, S. Information super-diffusion on structured networks. *Physica A* **2004**, *332*, 566. [[CrossRef](#)]
23. Tadić, B.; Thurner, S.; Rodgers, G.J. Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations. *Phys. Rev. E* **2004**, *69*, 036102. [[CrossRef](#)]
24. Tadić, B.; Rodgers, G.J.; Thurner, S. Transport on complex networks: Flow, jamming and optimization. *Int. J. Bifurcat. Chaos* **2007**, *17*, 2363. [[CrossRef](#)]
25. Andjelković, M.; Gupte, N.; Tadić, B. Hidden geometry of traffic jamming. *Phys. Rev. E* **2015**, *91*, 052817. [[CrossRef](#)]
26. Ling, X.; Wang, X.K.; Chen, J.J.; Liu, D.; Zhu, K.J.; Guo, N. Major impact of queue-rule choice on the performance of dynamic networks with limited buffer size. *Chin. Phys. B* **2020**, *29*, 018901. [[CrossRef](#)]
27. Kim, K.; Kahng, B.; Kim, D. Jamming transition in traffic flow under the priority queuing protocol. *Europhys. Lett.* **2009**, *86*, 58002. [[CrossRef](#)]
28. Tang, M.; Zhou, T. Efficient routing strategies in scale-free networks with limited bandwidth. *Phys. Rev. E* **2011**, *84*, 026116. [[CrossRef](#)]
29. Du, W.B.; Wu, Z.X.; Cai, K.Q. Effective usage of shortest paths promotes transportation efficiency on scale-free networks. *Physica A* **2013**, *392*, 3505. [[CrossRef](#)]
30. Zhang, X.J.; Guan, X.M.; Sun, D.F.; Tang, S.T. The effect of queueing strategy on network traffic. *Commun. Theor. Phys.* **2013**, *60*, 496. [[CrossRef](#)]
31. Wu, G.H.; Yang, H.J.; Pan, J.H. Efficient priority queueing routing strategy on networks of mobile agents. *Mod. Phys. Lett. B* **2018**, *32*, 1850137. [[CrossRef](#)]
32. Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [[CrossRef](#)] [[PubMed](#)]
33. Kurant, M.; Thiran, P. Layered complex networks. *Phys. Rev. Lett.* **2006**, *96*, 138701. [[CrossRef](#)]
34. Morris, R.G.; Barthelemy, M. Transport on coupled spatial networks. *Phys. Rev. Lett.* **2012**, *109*, 128703. [[CrossRef](#)] [[PubMed](#)]
35. Hill, S.A.; Braha, D. Dynamic model of time-dependent complex networks. *Phys. Rev. E* **2010**, *82*, 046105. [[CrossRef](#)] [[PubMed](#)]
36. Yang, H.X.; Wang, W.X.; Xie, Y.B.; Lai, Y.C.; Wang, B.H. Transportation dynamics on networks of mobile agents. *Phys. Rev. E* **2011**, *83*, 016102. [[CrossRef](#)] [[PubMed](#)]
37. Wang, W.X.; Wang, B.H.; Yin, C.Y.; Xie, Y.B.; Zhou, T. Traffic dynamics based on local routing protocol on a scale-free network. *Phys. Rev. E* **2006**, *73*, 026111. [[CrossRef](#)]
38. Arenas, A.; Díaz-Guilera, A.; Guimera, R. Communication in networks with hierarchical branching. *Phys. Rev. Lett.* **2001**, *86*, 3196. [[CrossRef](#)] [[PubMed](#)]

39. Ma, J.L.; Wang, H.L.; Zhang, Z.X.; Zhang, Y.; Duan, C.; Qi, Z.; Liu, Y. An efficient routing strategy for traffic dynamics on two-layer complex networks. *Int. J. Mod. Phys. B* **2018**, *32*, 1850155. [[CrossRef](#)]
40. Awerbuch, B. Shortest paths and loop-free routing in dynamic networks. *Comput. Commun. Rev.* **1990**, *20*, 4. [[CrossRef](#)]
41. Ling, X.; Hu, M.B.; Jiang, R.; Wu, Q.S. Global dynamic routing for scale-free networks. *Phys. Rev. E* **2010**, *81*, 016113. [[CrossRef](#)] [[PubMed](#)]
42. Gan, Y.; Tang, M.; Yang, H. Optimal forwarding ratio on dynamical networks with heterogeneous mobility. *Eur. Phys. J. B* **2013**, *86*, 209. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.