Preemptive Scheduler for Packet Processing
(May 2024)

Jatin Kesnani
K21-3204 Student
FAST University

Abstract

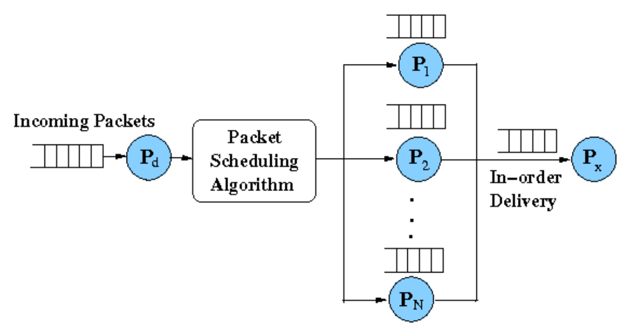
Wireless communications have become very pervasive over the time period. Developing packet scheduling algorithms in wireless networks can efficiently enhance delivery of packets through wireless links. Packet scheduling can guarantee quality of service and improve transmission rate in wireless networks. It is the process used to select which packet to be serviced or which to be dropped. This paper deals with various packet scheduling strategies in different wireless networks. Each wireless network has a different packet scheduling strategy and each has their own advantage and disadvantage. Most of the packet scheduling algorithm doesn't provide any sort of security. Only a few algorithms provide security but at the cost of schedulability so in this survey an algorithm which provides security by giving priority to schedulability is discussed.

1. Introduction

With the development of wireless technologies, wireless networks in recent years have been widely used in public places such as libraries, hotels, schools and airports due to the greater flexibility, increased efficiency, and reduced wiring costs. Wireless networks refer to any kind of computer network that does not involve any cables or wires. It is a technique that helps industrialists and telecommunications networks to save the cost of cables for networking in specific premises in their installations. Implemented and administration of transmission station is done via radio waves. This implementation is done at the physical level (layer) of the OSI model.

1.1 Packet Scheduling

Packet scheduling refers to the decision process used to select which packets should be serviced or dropped. Dropping of packets will be based on network characteristics like priority, packet arrival rate, deadline of packet and packet size. Scheduling will be done in scheduler. The scheduler will select certain packets based on various algorithms. Some of these algorithms have been selected for this survey. It's not possible for every packet to reach at destination some may be dropped along the way due to the effect of network characteristics mentioned previously. It is an efficient method to enhance the system performance. Packet scheduling is mainly applied to guarantee quality of service, improve transmission rate in wireless networks.



1.2 Problem Statement

The objective of this project is to explore and evaluate preemptive scheduling algorithms for packet processing in computer networks. Specifically, the project aims to implement and compare various preemptive packet scheduling algorithms, including Longest Remaining Time First (LRTF), Shortest Remaining Time First (SRTF), Priority, and Round Robin, in terms of their effectiveness in managing packet traffic.

This paper presents a study on different packet scheduling algorithms in different types of wireless networks. Section 2 describes various scheduling algorithms in different wireless networks, section 3 provides the Performance Analysis of Packet Scheduling Algorithms and section 4 presents the conclusion of this paper.

2. Packet Scheduling Algorithms

2.1 Shortest Remaining Time First Scheduler for Packet Processing

In the realm of packet processing within computer networks, the Shortest Remaining Time First (SRTF) scheduling algorithm emerges as a critical tool for optimizing resource utilization and minimizing packet processing delays. Unlike its counterpart in process scheduling, SRTF for packet processing prioritizes packets based on their remaining processing time, aiming to serve packets with the shortest remaining processing time first.

This scheduling paradigm aligns with the objective of minimizing packet processing latency, as packets with shorter remaining processing times are expedited through the network, leading to reduced queuing delays and improved overall network performance. By dynamically selecting the packet with the shortest remaining processing time for immediate processing, the SRTF scheduler ensures efficient utilization of network resources and timely delivery of data packets.

However, while SRTF offers advantages in terms of minimizing processing delays, it may also introduce challenges in scenarios where longer packets or flows experience starvation due to the continuous prioritization of short packets. Therefore, a comprehensive understanding of SRTF's implications, trade-offs, and its suitability for specific network environments is essential for effective deployment and optimization of packet processing operations.

Packet ID	Arrival Time	Processing Time	Finishing Time	Turnaround Time	Waiting Time
P1	3	6	40	37	31
P2	6	5	23	17	12
P3	4	4	9	5	1
P4	3	2	5	2	0
P5	10	5	28	18	13
P6	8	8	48	40	32
P7	7	4	18	11	7
P8	11	3	14	3	0
P9	9	2	11	2	0
P10	2	7	34	32	25

```
Average Waiting Time    : 12.1
Average Turnaround Time : 16.7
```

In this table, each row represents a packet, with columns indicating its packet number, arrival time, processing time, finishing time, and turnaround time. The turnaround time is calculated as the difference between the finishing time and the arrival time, representing the total time the packet spends in the system.

2.2 Longest Remaining Time First Scheduler for Packet Processing

In the realm of packet processing within computer networks, the Longest Remaining Time First (LRTF) scheduling algorithm stands as a significant approach to optimizing resource allocation and minimizing processing delays. Unlike its counterpart in process scheduling, where it prioritizes processes with the longest remaining execution time, LRTF for packet processing selects packets with the longest remaining processing time for immediate processing.

The LRTF scheduler aims to ensure efficient utilization of network resources by prioritizing packets that require more processing time. By dynamically selecting packets with longer processing times for immediate attention, LRTF can help minimize the risk of longer packets experiencing delays, thus contributing to reduced queuing and processing latencies in the network.

However, while LRTF offers advantages in terms of prioritizing packets with longer processing times, it may also introduce challenges in scenarios where shorter packets or flows experience starvation due to continuous prioritization of longer packets. Therefore, a comprehensive understanding of LRTF's implications, trade-offs, and its suitability for specific network environments is essential for effective deployment and optimization of packet processing operations.

Packet ID	Arrival Time	Processing Time	Finishing Time	Turnaround Time	Waiting Time
P1	3	6	40	37	31
P2	6	5	43	37	32
P3	4	4	42	38	34
P4	3	2	41	38	36
P5	10	5	47	37	32
P6	8	8	45	37	29
P7	7	4	44	37	33
P8	11	3	48	37	34
P9	9	2	46	37	35
P10	2	7	39	37	30

```
Average Waiting Time    : 32.6
Average Turnaround Time : 37.2
```

Similar to the presentation of SRTF results, each row in the table represents a packet, with columns indicating its packet number, arrival time, processing time, finishing time, and turnaround time. The turnaround time is calculated as the difference between the finishing time and the arrival time, representing the total time the packet spends in the system.

2.3 Priority Scheduler for Packet Processing

In the domain of packet processing within computer networks, the Priority scheduling algorithm holds significant importance in managing and optimizing the transmission of data packets. Unlike other scheduling algorithms that prioritize packets based on their arrival time or processing time, the Priority scheduler assigns priority levels to packets and processes higher-priority packets first.

The Priority scheduler aims to ensure that packets with higher importance or criticality are processed promptly, thereby meeting quality of service (QoS) requirements and ensuring the efficient utilization of network resources. By dynamically selecting packets based on their priority levels, the Priority scheduler can effectively handle scenarios where certain packets or flows require immediate attention over others.

However, while Priority scheduling offers advantages in terms of prioritizing critical packets, it may also introduce challenges related to fairness and resource starvation. Lower-priority packets or flows may experience delays or reduced throughput if higher-priority packets continually monopolize network resources. Therefore, careful consideration of priority assignment and scheduling policies is essential to maintain fairness and optimize network performance.

Packet ID	Arrival Time	Processing Time	Packet Priority	Finishing Time	Turnaround Time	Waiting Time
P1	3	6	9	48	45	39
P2	6	5	1	14	8	3
P3	4	4	8	42	38	34
P4	3	2	7	38	35	33
P5	10	5	4	24	14	9
P6	8	8	5	32	24	16
P7	7	4	6	36	29	25
P8	11	3	3	19	8	5
P9	9	2	2	16	7	5
P10	2	7	0	9	7	0

```
Average Waiting Time    : 16.9
Average Turnaround Time : 21.5
```

In this table, each row represents a packet, with columns indicating its packet number, arrival time, processing time, packet priority, finishing time, and turnaround time. The turnaround time is calculated as the difference between the finishing time and the arrival time, representing the total time the packet spends in the system.

2.4 Round Robin Scheduler for Packet Processing

In the landscape of packet processing within computer networks, the Round Robin scheduling algorithm serves as a fundamental method for managing and distributing processing resources among various packets. Unlike other scheduling algorithms that prioritize packets based on their characteristics or priorities, Round Robin scheduling allocates a fixed time quantum to each packet, ensuring fair and equitable utilization of network resources.

The Round Robin scheduler operates by cyclically rotating through packets in the queue and allowing each packet to execute for a predetermined time slice before moving on to the next packet. This approach ensures that all packets receive an equal opportunity for processing, regardless of their individual characteristics or priorities.

Round Robin scheduling is particularly well-suited for scenarios where fairness and equal resource allocation are essential considerations. By providing each packet with a fair share of processing time, Round Robin scheduling helps prevent starvation and ensures that all packets make progress towards completion.

However, while Round Robin scheduling offers advantages in terms of fairness and simplicity, it may also introduce challenges related to overhead and inefficiency. In scenarios with varying packet processing times or priorities, Round Robin scheduling may result in suboptimal performance compared to other scheduling algorithms.

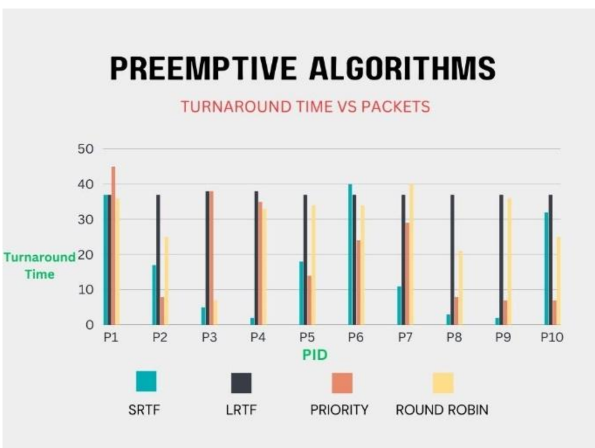
Packet ID	Arrival Time	Processing Time	Finishing Time	Turnaround Time	Waiting Time
P10	2	7	38	36	29
P1	3	6	28	25	19
P4	3	2	10	7	5
P3	4	4	37	33	29
P2	6	5	40	34	29
P7	7	4	41	34	30
P6	8	8	48	40	32
P9	9	2	30	21	19
P5	10	5	46	36	31
P8	11	3	36	25	22

```
Average Waiting Time    : 24.5
Average Turnaround Time : 29.1
```

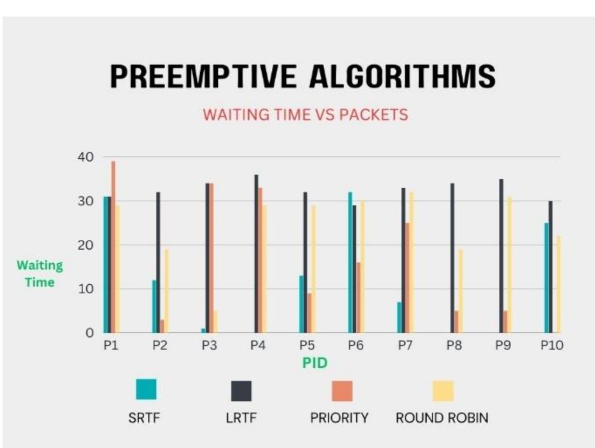
In this table, each row represents a packet, with columns indicating its packet number, arrival time, processing time, time quantum (3 unit) used for Round Robin scheduling, finishing time, and turnaround time. The turnaround time is calculated as the difference between the finishing time and the arrival time, representing the total time the packet spends in the system.

3. Performance Analysis of Packet Scheduling Algorithms

The first bar chart illustrates the turnaround time for each packet processed by the scheduling algorithms. Each bar represents a packet, with the x-axis indicating the packet number and the y-axis representing the corresponding turnaround time. This bar chart provides a visual overview of the distribution of turnaround times across packets, highlighting any variations or patterns in the data.



The second bar chart illustrates the waiting time experienced by each packet before being processed. Similar to the first bar chart, each bar represents a packet, with the x-axis indicating the packet number and the y-axis representing the waiting time. By examining this bar chart, we can assess the efficiency of the scheduling algorithms in minimizing waiting time and ensuring timely packet processing.



Together, these bar charts offer valuable insights into the performance of the packet scheduling algorithms, allowing for a comparative analysis of their effectiveness in terms of turnaround time and waiting time for the processed packets. Further interpretation and analysis of these bar charts will be conducted to draw meaningful conclusions regarding the performance and suitability of each scheduling algorithm for packet processing in computer networks.

4. Conclusion

In this study, we investigated and evaluated various preemptive scheduling algorithms for packet processing in computer networks. Through the implementation and analysis of Shortest Remaining Time First (SRTF), Longest Remaining Time First (LRTF), Priority, and Round Robin scheduling algorithms, we aimed to understand their impact on packet processing latency, throughput, and fairness.

Our findings reveal that each scheduling algorithm has distinct characteristics and performance implications. SRTF demonstrated efficiency in minimizing packet processing latency by prioritizing packets with the shortest remaining processing time. However, it may lead to starvation of longer packets or flows in certain scenarios. LRTF, on the other hand, prioritized longer packets but may suffer from fairness issues.

Based on the results obtained from our study, we can draw the following conclusions:

- The LRTF algorithm performs well in scenarios where long-running packets need to be prioritized.
- The SRTF algorithm is effective in reducing response time and overall processing time.
- Priority scheduling provides a flexible approach, allowing packets to be processed according to their assigned priority.
- Round Robin scheduling ensures fair processing by providing equal time slices to each packet.

The choice of preemptive scheduling algorithm depends on the specific requirements of the system and the workload characteristics. Factors such as responsiveness, fairness, and throughput should be considered when selecting an appropriate algorithm for a given scenario.