**Performance Metrics Calculation in Network**

**Unit I**

**1)**



**2)**  In a high-traffic web server, network buffer optimization is crucial to ensure efficient data transmission and minimize packet loss. Consider a scenario where you have a server that handles HTTP requests and responses. The server has a network interface with a buffer size of 512 KB (kilobytes). You are given the following data:

Average size of incoming HTTP requests: 8 KB

Average size of outgoing HTTP responses: 64 KB

Average round-trip time (RTT) for a request-response cycle: 50 ms (milliseconds)

Network bandwidth: 100 Mbps (megabits per second)

Assuming the server processes 1000 requests per second, calculate the following:

a) The average data rate (in Mbps) required to handle the incoming requests

b) The average data rate (in Mbps) required to handle the outgoing responses.

c) The total data rate (in Mbps) the server needs to handle both incoming requests and outgoing responses.

d) The time (in milliseconds) it would take to fill the network buffer at the current request rate.

e) Given the buffer size and the request rate, determine whether the current buffer size is sufficient to handle the traffic without overflow. If not, calculate the optimal buffer size required.

**3)** Consider an HTTP GET request sent from a client to a server. If the request line is as follows:

GET/index.html HTTP/1.1

And the headers of the request are:

Host: www.example.com

User-Agent: Mozilla/5.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Assume the size of each character is 1 byte. Calculate the total size in bytes of this HTTP GET request message, including the request line, headers, and the necessary carriage return and line feed characters (CRLF).

**4)** Your web application's average DNS lookup time is 150ms, TCP connection time is 120ms, TLS handshake time is 200ms, TTFB is 300ms, and content download time is 500ms. Analyze these metrics and identify which component is contributing the most to the overall response time of 1270ms. Propose a strategy to reduce the total response time by at least 20%.

**5)** Your server handles 500 requests per second under normal conditions. During peak hours, the number of requests increases to 1000 requests per second, causing a significant performance drop. The current CPU utilization is 70%, and memory usage is 80%. Analyze the relationship between resource utilization and throughput capacity, and recommend changes to hardware or software configurations to handle peak traffic without degradation.

**6)** Over the past week, your application has logged 10,000 requests, with 600 resulting in 5xx errors. Perform an analysis to identify patterns in the error occurrences. Assume that 70% of the 5xx errors are due to server overload, 20% due to database connection issues, and 10% due to application bugs. What are the potential underlying issues, and how would you address them to reduce the error rate to below 2%?

**7)** Your server currently supports 1,000 simultaneous connections, but performance degrades significantly beyond 800 active connections. Analyze the current concurrency model and suggest improvements to handle 1,500 simultaneous connections efficiently. Consider a scenario where switching to an asynchronous I/O model could potentially improve performance. What would be the expected outcome based on this analysis?

**8)** Users report high latency accessing your web application during peak hours, with the following breakdown: DNS lookup time is 100ms, TCP connection time is 80ms, TLS handshake time is 150ms, TTFB is 250ms, and content download time is 400ms. Identify the main contributors to the total latency of 980ms and propose strategies to reduce latency by 30%.

**9)** Your web application has a cache hit ratio of 60%, resulting in 4,000 cache hits and 2,667 cache misses out of 10,000 total requests. Analyze the current caching strategy and policies to identify areas for improvement. What changes would you recommend to increase the cache hit ratio to 80% and how would this impact overall performance?

**10)** Your web application uses Gzip compression, achieving an average compression ratio of 2:1 with a CPU overhead of 10%. Brotli compression offers a compression ratio of 3:1 with a CPU overhead of 15%. Analyze the trade-offs between Gzip and Brotli. How would switching to Brotli affect overall performance, and would you recommend making the switch?

**11)** A specific web page on your site has an average load time of 12 seconds, broken down as follows: HTML load time is 2 seconds, CSS load time is 1 second, JavaScript load time is 3 seconds, and image load time is 6 seconds. Analyze these components and suggest optimizations to reduce the total load time to below 5 seconds.

**12)** During peak traffic, your server's CPU usage is 90% and memory usage is 85%, leading to performance degradation. Analyze the server's resource utilization metrics under these conditions. Given that optimizing the application code could reduce CPU usage by 20% and memory usage by 15%, what actions would you take to improve resource utilization and overall performance?

**13)** Your web application is experiencing poor performance due to network conditions, with 5% packet loss and 30ms jitter. Analyze the impact of these network issues on application performance. Wyatt are the main causes of these issues, and what mitigation strategies would you recommend to improve performance under these conditions?