



Congestion Control in Computer Networks

Difficulty Level : Medium • Last Updated : 23 Aug, 2022

Read

Discuss

What is **congestion**?

A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

Effects of Congestion

- As delay increases, performance decreases.
- If delay increases, retransmission occurs, making situation worse.

Congestion control algorithms

- Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse.
- Congestive-Avoidance Algorithms (CAA) are implemented at the TCP layer as the mechanism to avoid congestive collapse in a network.
- There are two congestion control algorithm which are as follows:

- **Leaky Bucket Algorithm**

- The leaky bucket algorithm discovers its use in the context of network traffic shaping or rate-limiting.
- A leaky bucket execution and a token bucket execution are predominantly used for traffic shaping algorithms.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

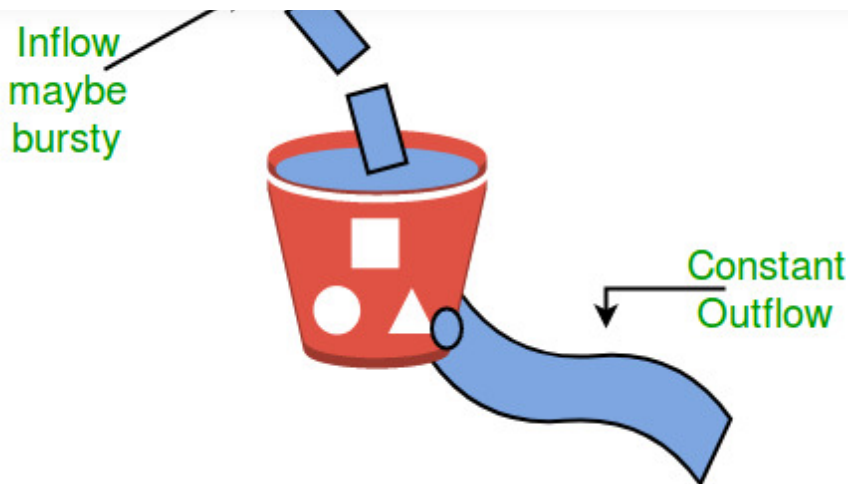
Got It !

effectively.

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.

DSA Data Structures Algorithms Interview Preparation Data Science Topic-wise Practice



Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

- **Token bucket Algorithm**

- The leaky bucket algorithm has a rigid output design at an average rate independent of the bursty traffic.
- In some applications, when large bursts arrive, the output is allowed to speed up. This calls for a more flexible algorithm, preferably one that never loses

size. Tokens in the bucket are deleted for the ability to share a packet.

- When tokens are shown, a flow to transmit traffic appears in the display of tokens.
- No token means no flow sends its packets. Hence, a flow transfers traffic up to its peak burst rate in good tokens in the bucket.

Need of token bucket Algorithm:-

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket. f
2. The bucket has a maximum capacity. f
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.

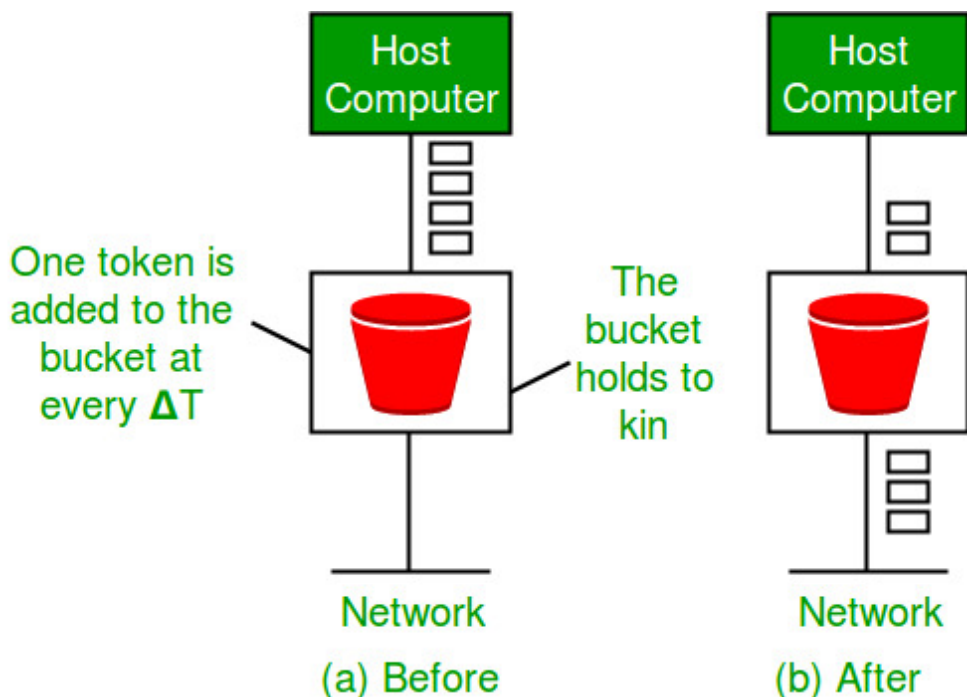
Let's understand with an example,

In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.

Ways in which token bucket is superior to leaky bucket: The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature. Some flexibility is introduced in the token bucket algorithm. In the token bucket, algorithm tokens are generated at each tick (up to a certain limit).

Formula: $M * s = C + \rho * s$ where S – is time taken M – Maximum output rate ρ – Token arrival rate C – Capacity of the token bucket in byte

Let's understand with an example,



Link to question on leaky bucket algorithm:

<https://www.geeksforgeeks.org/computer-networks-set-8/>

This article is contributed by [Vikash Kumar](#). Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Interview Series

Prepare for free
Every Sunday | 7 - 8:30 PM IST



Like 66

RECOMMENDED ARTICLES

Page : [1](#) [2](#) [3](#)

- 01** Congestion Control techniques in Computer Networks
27, Nov 17
- 02** Difference between Flow Control and Congestion Control
24, May 19
- 03** TCP Congestion Control
31, Oct 17
- 04** What is DCCP (Datagram Congestion Control Protocol)?
21, Mar 22
- 05** Congestion Control in Datagram Subnets
10, Jun 21
- 06** Slow Start Restart Algorithm For Congestion Control
01, Feb 22
- 07** Access Control Tactics in Computer Networks
09, Dec 19
- 08** What is Network Congestion? Common Causes and How to Fix Them?
15, Oct 21

Article Contributed By :



GeeksforGeeks

Vote for difficulty

Current difficulty : [Medium](#)

[Easy](#)[Normal](#)[Medium](#)[Hard](#)[Expert](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)[Improve Article](#)[Report Issue](#)

GeeksforGeeks

A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

- [About Us](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [Privacy Policy](#)
- [Copyright Policy](#)

News

- [Top News](#)
- [Technology](#)
- [Work & Career](#)
- [Business](#)
- [Finance](#)
- [Lifestyle](#)
- [Knowledge](#)

Learn

- [Algorithms](#)
- [Data Structures](#)
- [SDE Cheat Sheet](#)
- [Machine learning](#)
- [CS Subjects](#)
- [Video Tutorials](#)
- [Courses](#)

Languages

- [Python](#)
- [Java](#)
- [C++](#)
- [Golang](#)
- [C#](#)
- [SQL](#)
- [Kotlin](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

JavaScript

Bootstrap

ReactJS

NodeJS

Write Interview Experience

Internships

Video Internship

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !