Additive Increase Multiplicative Decrease (AIMD)

[Your Name]

5th Semester, Computer ScienceYour College/University Name>Your College/Unive

Course: Computer Networks

[Month, Year]

Contents

Introduction

In computer networks, congestion control is a critical aspect that ensures smooth and efficient data transmission. One of the fundamental congestion control mechanisms employed by protocols such as TCP (Transmission Control Protocol) is *Additive Increase Multiplicative Decrease* (AIMD). As a 5th semester Computer Science student studying Computer Networks, I have had the opportunity to understand how AIMD helps maintain network stability, fairness, and efficiency.

Concept of AIMD

AIMD is based on a simple yet effective approach to controlling the sending rate of data:

Additive Increase

Under normal conditions, when the network is not experiencing significant congestion, the sender gradually increases its congestion window (cwnd). Typically, this is done by adding one Maximum Segment Size (MSS) per Round Trip Time (RTT). This slow, linear growth allows the sender to carefully probe the available bandwidth and attempt to send more data without causing congestion.

Multiplicative Decrease

When the sender detects congestion (often inferred from packet loss or timeout events), it rapidly decreases its congestion window, commonly by halving it. This sharp reduction in cwnd helps the network recover by preventing further overload and allowing queued packets to clear.

Visualizing AIMD

To better understand AIMD, consider the behavior of the congestion window over time as shown in Figure 1.

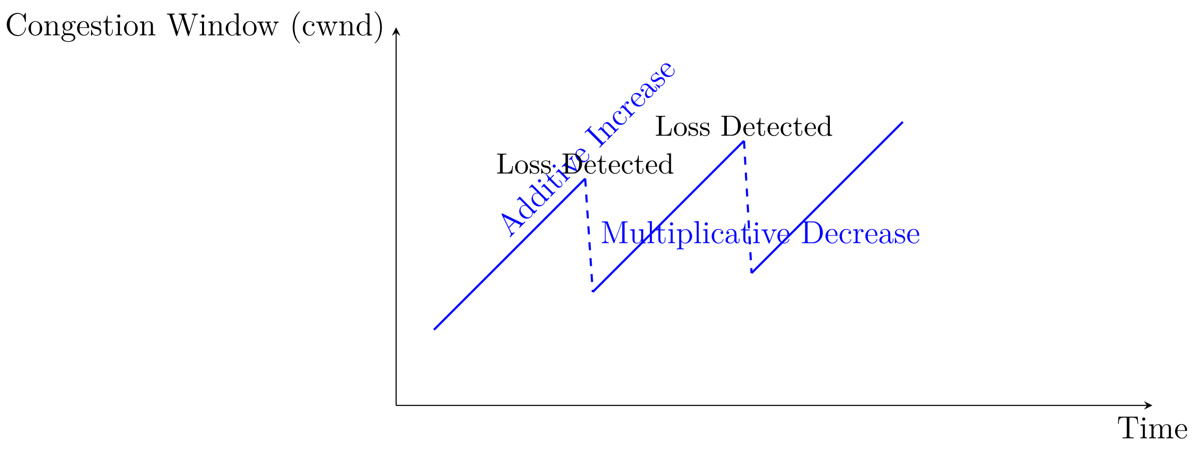

Figure 1: Illustration of AIMD: The congestion window increases additively until a loss is detected, after which it decreases multiplicatively, and then grows again.

In the figure, the congestion window size grows linearly during the additive increase phase. Once a loss occurs, the window is quickly reduced (multiplicative decrease), and then the cycle repeats as the window builds up again linearly.

Why Do We Need AIMD?

Without congestion control, all senders might aggressively transmit at their maximum rates, causing the network to become overloaded. Such scenarios lead to packet loss, delays, and inefficiency. AIMD ensures a balanced approach:

- **Fairness:** Each flow that uses AIMD eventually converges toward a fair share of the available bandwidth. Over time, faster senders are reined in by multiplicative decreases, while more cautious senders gradually speed up through additive increases, encouraging equilibrium.
- **Stability:** By combining gentle probing (additive increase) and decisive back-off (multiplicative decrease), AIMD prevents persistent congestion, leading to a more stable network environment.

How AIMD Works in Practice

1. **Initial Phase:** The congestion window starts with a small value (e.g., 1 MSS). The sender slowly increases it over consecutive RTTs, monitoring for signs of congestion.
2. **Detecting Congestion:** If a packet is lost or a timeout occurs, the sender assumes the network is congested. As a result, it immediately reduces the cwnd by a multiplicative factor (often halves it).

3. **Steady-State Operation:** After the reduction, the sender once again begins to increment cwnd additively, attempting to find a new stable equilibrium point. Over time, this process repeats, allowing the network to adapt dynamically to changing conditions.

Real-World Analogy

Imagine walking down a hallway that can only comfortably fit a certain number of people. At first, you walk slowly, observing if you can move faster without bumping into anyone. If the hallway seems clear, you increase your pace slightly (*additive increase*). But the moment you see a crowd ahead, you slow down drastically (*multiplicative decrease*) to avoid collisions and confusion. Once the hallway clears again, you can try speeding up slowly.

Conclusion

AIMD is a cornerstone algorithm in the world of computer networking. Its balanced approach ensures that while the network is utilized efficiently, no single flow dominates the available resources. By integrating slow, linear increases with abrupt, multiplicative decreases in response to congestion, AIMD helps maintain an equilibrium that benefits both the network and its users.

References

- Kurose, J. F., & Ross, K. W. (2017). *Computer Networking: A Top-Down Approach.* Pearson.
- Peterson, L. L., & Davie, B. S. (2011). *Computer Networks: A Systems Approach.* Morgan Kaufmann.
- Stevens, W. R., Fenner, B., & Rudoff, A. M. (2004). *UNIX Network Programming.* Addison-Wesley.