Principles of Congestion Control

Study-Ready Notes

Compiled by Andrew Photinakis

November 15, 2025

Contents

1 Introduction to Congestion Control

- Fundamental problem in computer networks
- Different from flow control (which handles sender-receiver speed mismatch)
- Considered one of the top-10 problems in networking
- Essential for network stability and fair resource sharing

1.1 Definition and Manifestations

- **Congestion**: "Too many sources sending too much data too fast for *network* to handle"
- **Manifestations**:
  - Long delays (queueing in router buffers)
  - Packet loss (buffer overflow at routers)

1.2 Congestion Control vs Flow Control

Congestion Control	Flow Control
Too many senders, sending too fast	One sender too fast for one receiver
Network-wide problem	End-to-end problem
Prevents network collapse	Prevents receiver buffer overflow
Affects all users sharing network	Affects single connection
Example: TCP congestion control	**Example**: TCP flow control (rwnd)

Table 1: Comparison of congestion control vs flow control

[Summary: Congestion control manages network-wide resource overutilization when too many senders overload network capacity, while flow control handles speed mismatches between individual sender-receiver pairs.]

2 Causes and Costs of Congestion: Three Scenarios

2.1 Scenario 1: Infinite Buffers, No Retransmissions

2.1.1 Setup

- Two flows sharing one router
- Infinite buffers at router
- Input/output link capacity: R

- No retransmissions needed (perfect channel)

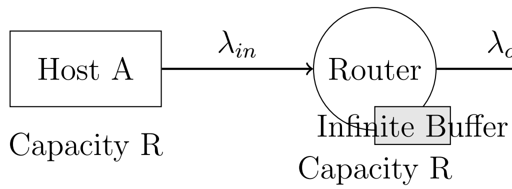

Figure 1: Scenario 1: Two flows sharing router with infinite buffers

2.1.2 Behavior and Costs

- Maximum per-connection throughput: R/2
- As arrival rate λ_{in} approaches R/2:
  - Large delays due to queueing
  - Throughput approaches R/2
  - No packet loss (infinite buffers)

$$\text{Maximum throughput per connection} = \frac{R}{2}$$

2.2 Scenario 2: Finite Buffers with Retransmissions

2.2.1 Setup

- One router with finite buffers
- Sender retransmits lost/timed-out packets
- Transport-layer input includes retransmissions: $\lambda'_{in} \geq \lambda_{in}$
- Application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$

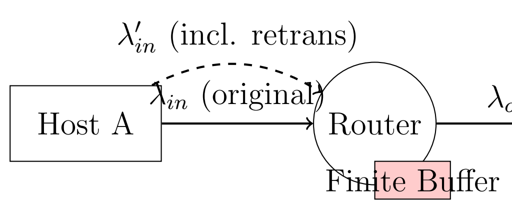

Figure 2: Scenario 2: Finite buffers with retransmissions

2.2.2 Ideal Case: Perfect Knowledge

- Sender sends only when router buffers available
- No unneeded retransmissions
- Still have "wasted" capacity due to needed retransmissions

2.2.3 Realistic Case: Unneeded Duplicates

- Premature timeouts cause unneeded duplicate transmissions
- Both original and duplicate copies may be delivered
- Further decreases effective throughput

2.2.4 Costs in Scenario 2

- More work (retransmissions) for given receiver throughput
- Unneeded retransmissions waste link capacity
- Decreased maximum achievable throughput

[Mnemonic: "Finite Buffers, Infinite Problems" - Finite buffers lead to packet loss, which causes retransmissions that further congest the network.]

2.3 Scenario 3: Multi-Hop Paths with Multiple Senders

2.3.1 Setup

- Four senders with multi-hop paths
- Timeout/retransmit mechanisms
- Complex interactions between flows

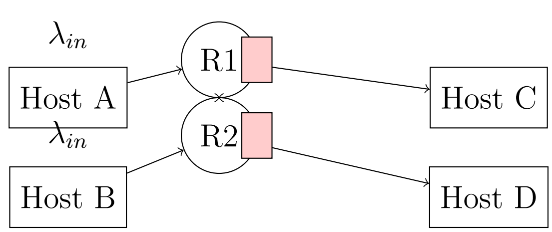

Figure 3: Scenario 3: Multiple senders with multi-hop paths

2.3.2 Critical Insight

- As red λ'_{in} increases, all arriving blue packets at upper queue are dropped
- Blue throughput approaches zero due to congestion collapse
- Upstream transmission capacity and buffering wasted for packets lost downstream

2.4 Summary of Congestion Costs

- Throughput can never exceed capacity
- Delay increases as capacity approached
- Loss/retransmission decreases effective throughput
- Unneeded duplicates further decrease effective throughput
- Upstream resources wasted for packets lost downstream
- Potential for congestion collapse (throughput $\to$ 0)

[Summary: Congestion costs include decreased throughput, increased delay, wasted resources on retransmissions, and potential congestion collapse where useful throughput drops to near zero despite high network load.]

3 Approaches to Congestion Control

3.1 End-to-End Congestion Control

- No explicit feedback from network
- Congestion *inferred* from observed loss and delay
- Approach taken by standard TCP
- Senders monitor ACK patterns and timing

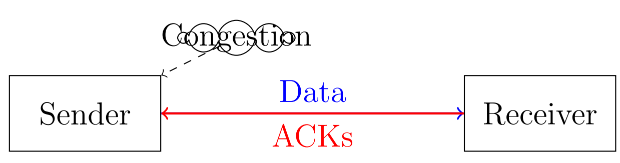

Figure 4: End-to-end congestion control: sender infers congestion from ACK patterns

3.2 Network-Assisted Congestion Control

- Routers provide direct feedback to sending/receiving hosts
- May indicate congestion level or explicitly set sending rate
- More explicit and potentially faster response

End-to-End Control	**Network-Assisted Control**
Congestion inferred from loss/delay	Explicit feedback from routers
Implemented at transport layer	Requires router participation
TCP standard approach	TCP ECN, ATM, DECbit
Slower to react	Faster, more precise reaction
Works with existing infrastructure	Requires router upgrades
Advantage: Deployment ease	**Advantage**: Performance

Table 2: Comparison of end-to-end vs network-assisted congestion control

3.3 Examples of Network-Assisted Approaches

- **TCP ECN (Explicit Congestion Notification)**: Routers mark packets to indicate congestion
- **ATM (Asynchronous Transfer Mode)**: Complex signaling for congestion control
- **DECbit**: Early research protocol using single bit for congestion indication

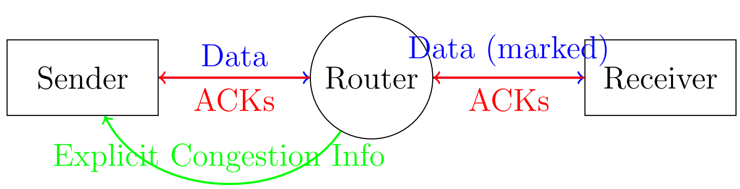

Figure 5: Network-assisted congestion control: routers provide explicit feedback

[Concept Map: Congestion Control Approaches → End-to-End (TCP: infer from loss/delay) vs Network-Assisted (ECN, ATM: explicit feedback) → Trade-off between deployment ease and performance → Both aim to prevent congestion collapse and ensure fair resource sharing.]

4 Congestion Control Insights and Principles

4.1 Fundamental Trade-offs

- **Fairness vs Efficiency**: Maximizing total throughput may not be fair to all users
- **Responsiveness vs Stability**: Quick reaction vs oscillation prevention
- **Simplicity vs Optimality**: Easy implementation vs perfect performance

4.2 Key Design Principles

1. **Conserve Packets**: Don't send unless network can deliver
2. **Adapt to Change**: Adjust rates based on network conditions
3. **Distribute Fairly**: All flows should get reasonable share
4. **Avoid Collapse**: Prevent throughput dropping to zero
5. **Scale Well**: Work in networks of all sizes

4.3 Mathematical Foundations

The relationship between offered load and achieved throughput can be modeled as:

$$\text{Effective Throughput} = \frac{\text{Offered Load}}{1 + \text{Retransmission Overhead}}$$

Where retransmission overhead increases non-linearly with offered load.

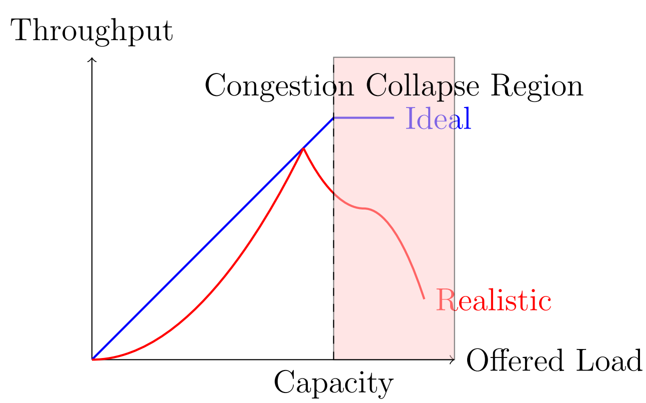

Figure 6: Relationship between offered load and throughput showing congestion collapse

[Summary: Effective congestion control must balance fairness, efficiency, and stability while preventing congestion collapse, using either end-to-end inference or network-assisted explicit feedback mechanisms.]

5 Study Aids and Exam Preparation

5.1 Key Concepts to Master

- Understand the three congestion scenarios and their specific costs
- Differentiate between congestion control and flow control
- Compare end-to-end vs network-assisted approaches
- Explain the causes of congestion collapse
- Describe the trade-offs in congestion control design

5.2 Practice Questions

1. **Compare and contrast** the three congestion scenarios discussed. What specific costs does each scenario illustrate, and how do they build upon each other?
2. Explain why **congestion control** is fundamentally different from **flow control**. Provide concrete examples of each and describe what problems they solve.
3. Describe the **congestion collapse** phenomenon. How can it occur in Scenario 3, and what mechanisms can prevent it?
4. Compare **end-to-end congestion control** with **network-assisted congestion control**. What are the advantages and disadvantages of each approach?
5. A network has a bottleneck link with capacity R. If N TCP connections share this link fairly, what is the maximum throughput each can achieve? What happens as N becomes very large?

[Mnemonic: "End-to-End Estimates, Network Notifies" - End-to-end control estimates congestion from observations, while network-assisted control provides explicit notifications.]

6 Summary

- Congestion occurs when network resources are overutilized by too many senders
- Three scenarios illustrate progressive complexity of congestion costs:
  - Scenario 1: Delay increases with infinite buffers
  - Scenario 2: Retransmissions waste capacity with finite buffers
  - Scenario 3: Congestion collapse possible in multi-hop networks
- Costs include decreased throughput, increased delay, and wasted resources
- Two main approaches:

  - End-to-end control (TCP standard): Infer congestion from observations
  - Network-assisted control (ECN, ATM): Explicit router feedback

- Effective congestion control must balance fairness, efficiency, and stability
- Prevention of congestion collapse is critical for network operation

7 References

- Kurose, J.F., & Ross, K.W. (2020). *Computer Networking: A Top-Down Approach (8th ed.)*. Pearson.
- Course: COMPSCI 453 Computer Networks, University of Massachusetts
- Professor: Jim Kurose, College of Information and Computer Sciences
- Jacobson, V. (1988). "Congestion Avoidance and Control"
- RFC 5681: TCP Congestion Control
- Textbook website: http://gala.cs.umass.edu/kurose_ross