Network Security

Session 3E

Congestion Control Protocols (DCCP, QUIC, DASH)

Mouli Sankaran

Session 3E: Focus

- Congestion Control Protocols
  - DCCP
  - QUIC
  - DASH
- Host-centric Vs Router-centric Protocols

Course page where the course materials will be posted as the course progresses:

Datagram Congestion Control Protocol (DCCP)

DCCP is a transport layer protocol defined by **RFC 4340**.

DCCP

- It is designed to provide congestion control for **connectionless datagram-based** applications, bridging the gap between the reliability of TCP and the low-overhead of UDP.
- **Congestion Control:** Unlike UDP, which lacks built-in congestion control, DCCP incorporates mechanisms to manage network congestion, making it suitable for applications that require timely delivery without the overhead of ensuring reliability.
- **Unreliable Delivery:** DCCP does not guarantee the delivery of packets, allowing applications to handle any necessary retransmissions or error corrections.

- DCCP is a separate protocol (like TCP/UDP).
- Protocol number **33** (distinct in the IP header).
- It has its own header (does not reuse the UDP header)
- It has three-way handshake connection establishment
- 48-bit seq number for congestion control nor reliability.

DCCP Header Fields (simplified view)

Field	Size
Source Port	16 bits
Destination Port	16 bits
Data Offset	8 bits
CCVal / ResFlags	8 bits
Sequence Number	48 bits
Acknowledgement	48 bits (optional)
Payload	Variable

DCCP: Explained

- DCCP uses acknowledgments to inform the sender if its packets have arrived and if they were marked by Explicit Congestion Notification (ECN)
  - ECN is notified by intermediate routers while forwarding the packets
- Blind attackers try to guess the 32-bite sequence numbers used by TCP, DCCP increases it to 48-bit with additional guard bits, to make it harder for the attackers to predict the sequence numbers used by the connections.
- DCCP provides support for acknowledgments both in bidirectional and uni-directional data transfers.
- DCCP is designed to work well on networks with high levels of bandwidth fluctuation.
- It's intended for applications like streaming media, VOIP, online gaming

Quick UDP Internet Connections (QUIC)

QUIC

- QUIC is a transport layer network protocol developed by Google, built on top of UDP, aiming to reduce latency and improve performance compared to TCP
- Incorporates features like multiplexing, encryption, and improved congestion control within the protocol itself.
- **Fast Handshake:** Reduces connection establishment time, enabling quicker data transmission
- **Web Browsing:** Enhances loading times and responsiveness for web pages.
- **Streaming Services**: Provides smoother video and audio streaming experiences.
- **Real-Time Applications:** Supports applications requiring rapid data transmission, such as video conferencing and online gaming.

QUIC: Explained

- QUIC supports multiple independent streams within a single connection, reducing head-of-line blocking
- It has built-in selective acknowledgment, which allows it to adjust its congestion control more granularly than TCP.
- It uses a hybrid slow start algorithm that dynamically adjusts the window size based on round-trip times.
- Slow start uses smaller window size to study the network congestion before using larger window sizes
- It estimates bandwidth in each direction to avoid congestion.
- It can seamlessly migrate connections between different networks, such as Wi-Fi and mobile data.

Dynamic Adaptive Streaming over HTTP (DASH)

DASH

- DASH is an application layer protocol designed for streaming multimedia content over the Internet.
- It dynamically adjusts the quality of the video stream based on network conditions, ensuring smooth playback without buffering.
- **Adaptive Bitrate Streaming**: Adjusts video quality in real-time to match the user's available bandwidth.
- **Segmented Media Delivery**: Breaks video content into small segments, allowing for quick adjustments to streaming quality.
- **HTTP-Based**: Utilizes standard HTTP protocols, ensuring compatibility across various devices and platforms.
- Applications: Video streaming, live broadcasting, etc.

Host-centric Vs Router-centric Protocols

Is TCP host-centric or router-centric protocol? **ANS: Host-centric**

Host-centric Vs Router-centric Protocols

Host-centric	Router-centric
This category of protocols that focus on the end devices (hosts) **Note**: TCP is a good example of a protocol which is Host-centric	It describe approaches, protocols, or functionalities that primarily focus on the network infrastructure (routers)
Examples: TCP, application-layer protocols such as, HTTP, FTP, etc.	**Examples**: Routing protocols such as, OSPF and BGP, Quality of Service (QoS)
Protocols and mechanisms that are managed and controlled by the end hosts. These often involve data transmission, flow control, error handling, and application-specific functionalities	Protocols and mechanisms that are managed by network devices to control traffic flow, manage congestion, enforce policies, and optimize routing

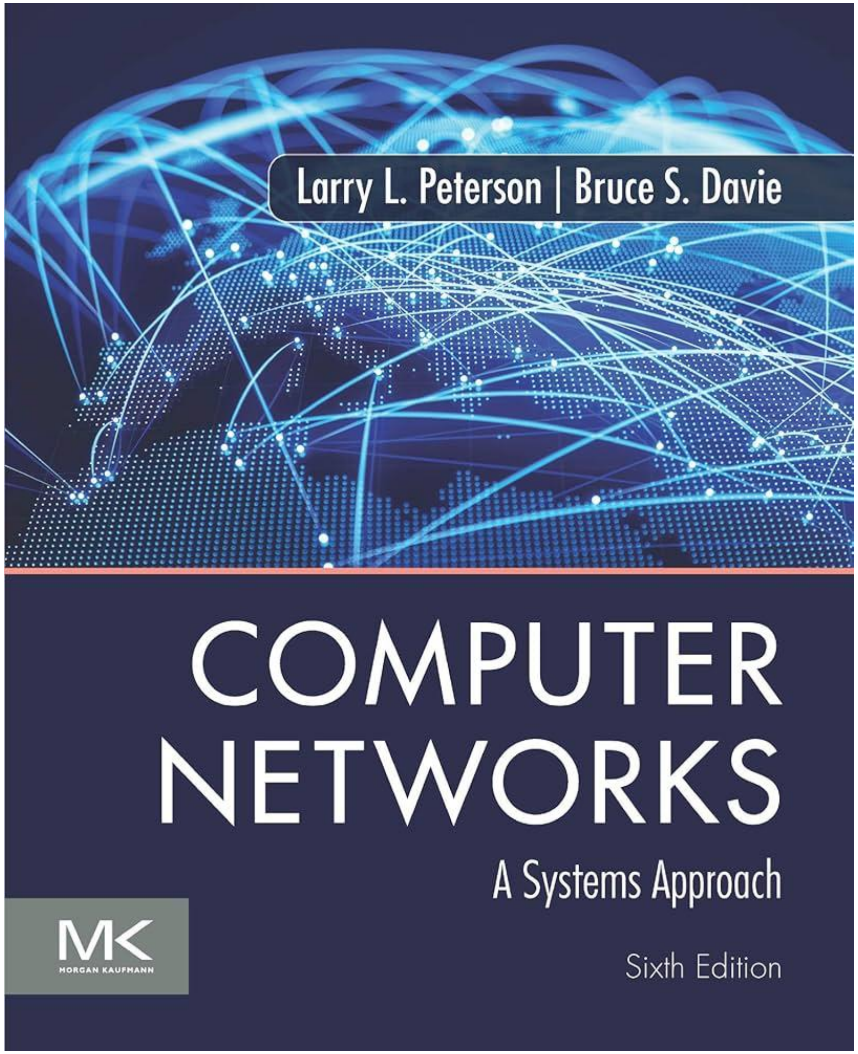
Emerging protocols such as QUIC, SDN provide a more integrated approach between host-centric and router-centric functionalities. QUIC allows for more seamless interactions with modern network features like Explicit Congestion Notification (ECN) from routers, that enables more optimized routing, bridging host and router-centric functionalities.

Session 3E: Summary

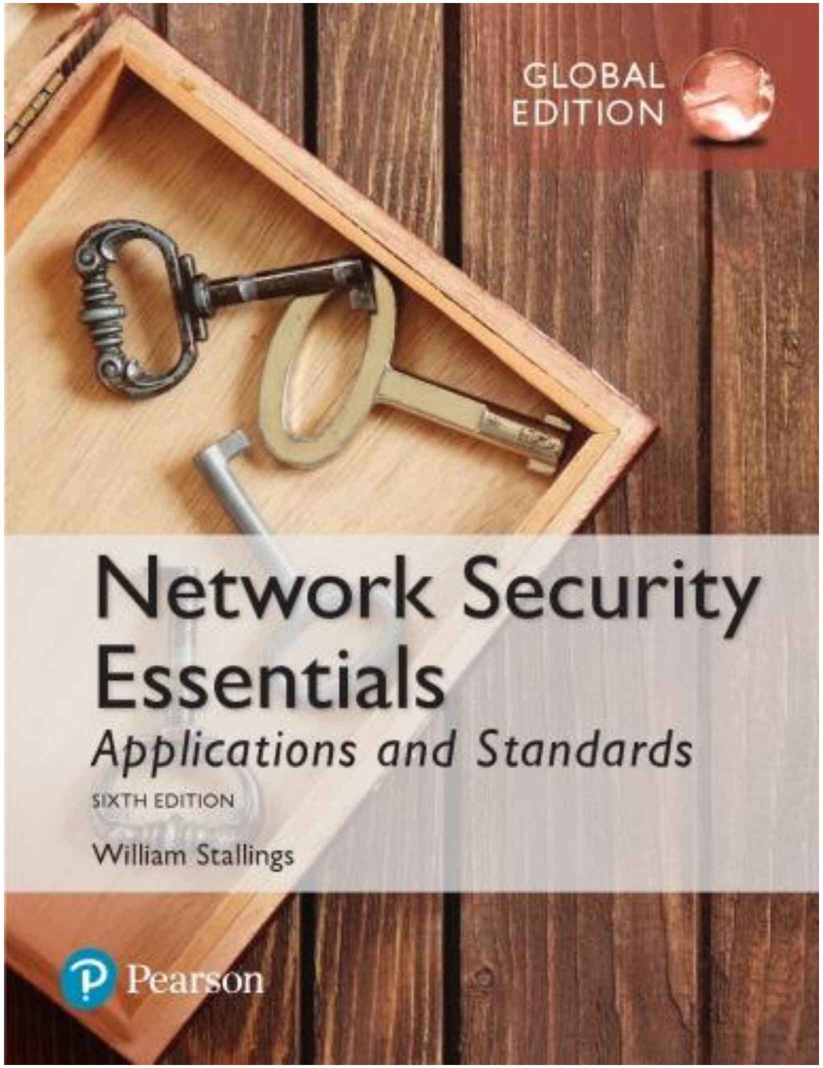
- Congestion Control Protocols
  - DCCP
  - QUIC
  - DASH
- Host-centric Vs Router-centric Protocols

Textbooks

Textbook 1

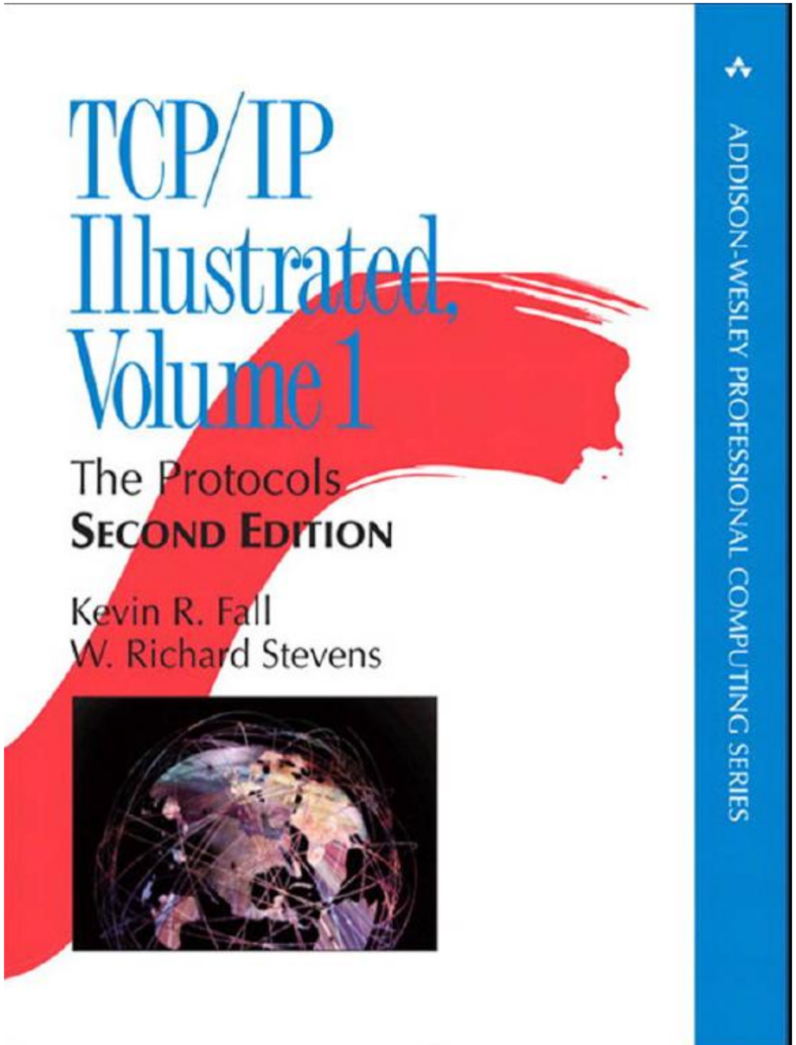


Textbook 2



References

Ref 1



Ref 2

[TCP Congestion Control: A Systems Approach]

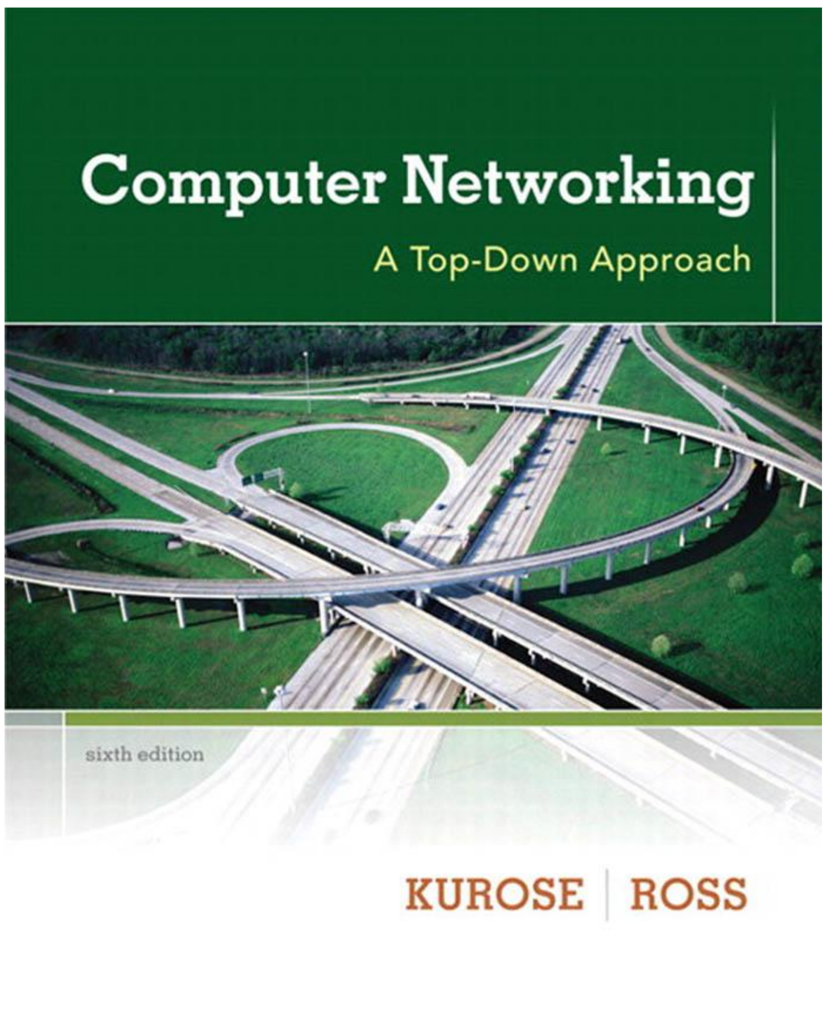
HTTPS://SYSTEMSAPPROACH.ORG

TCP Congestion Control: A Systems Approach

Peterson, Brakmo, and Davie

References

Ref 3



Ref 4