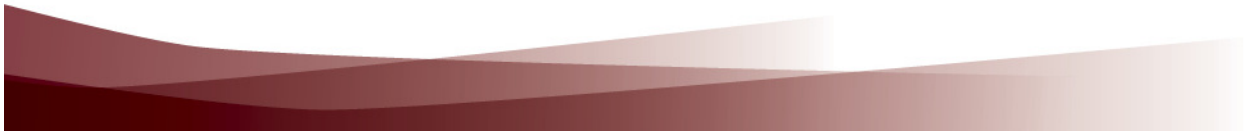
Data Center Networking: Challenges and Traditional Protocols

(ENCS 691K – Chapter 6)

Roch Glitho, PhD

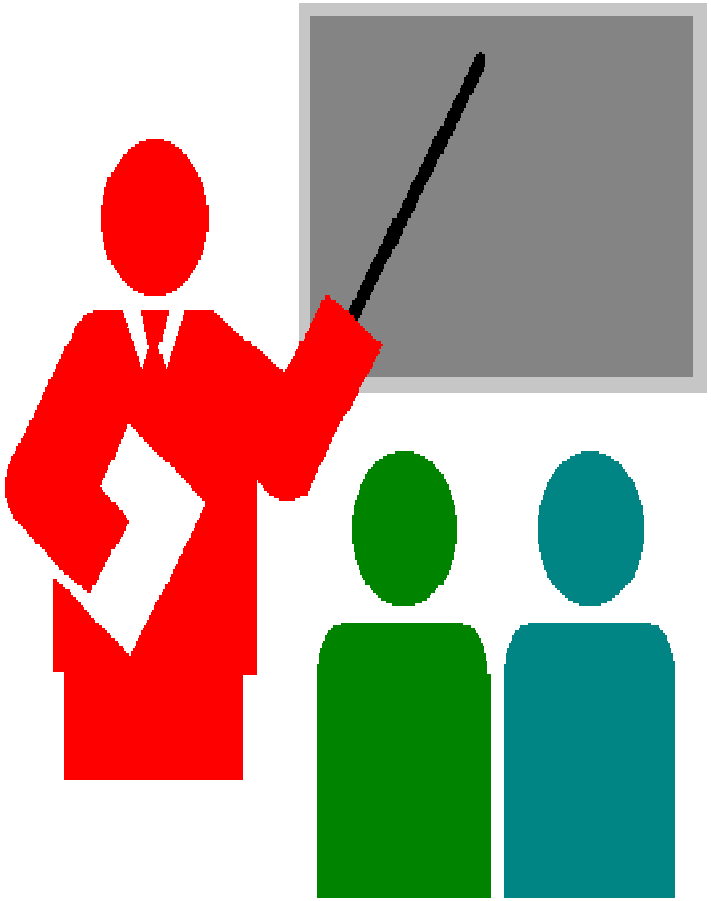
Associate Professor and Canada Research Chair

My URL - http://users.encs.concordia.ca/~glitho/



Concordia University
Engineering and Computer Science
Concordia Institute for Information Systems Engineering

Data Center Networking: Challenges and Traditional Protocols



- Data Center Networking Challenges
- On Networking
- On Transport Layer
- Traditional Transport Protocols (Beyond TCP / UDP)
- Traditional Transport Protocols vs. Challenges

Data Center Networking Challenges

References

1. K. Kant, Towards a Virtualized Data Center Transport Protocol, Infocom Workshop, 2008
2. M Alizadeh, Data Center TCP, ACM Sigcom 2011

Data Center Networking Challenges

Why is it necessary to re-think networking in cloud data center settings?

- Very high data rates (e.g. 100 Gb/sec Ethernet)
  - TCP can hardly cope with 10 GB/sec
    - New techniques are needed to make TCP cope, e.g.
      - Hardware acceleration
  - Need for QoS mechanisms
    - A single MAC pipe can carry data with different QoS requirements

Data Center Networking Challenges

Why is it necessary to re-think networking in cloud data center settings?

- Wide range of physical layer
  - Wired
  - Wireless
  - Optical
- Emerging PHY/MAC layers, e.g.
  - Ultra Wide Band
    - Huge amount of data over a short distance

Data Center Networking Challenges

Why is it necessary to re-think networking in cloud data center settings?

- Multiple level virtualization and cluster enabled applications
  - Real time applications / soft real time applications vs. other applications

Data Center Networking Challenges

An illustration:

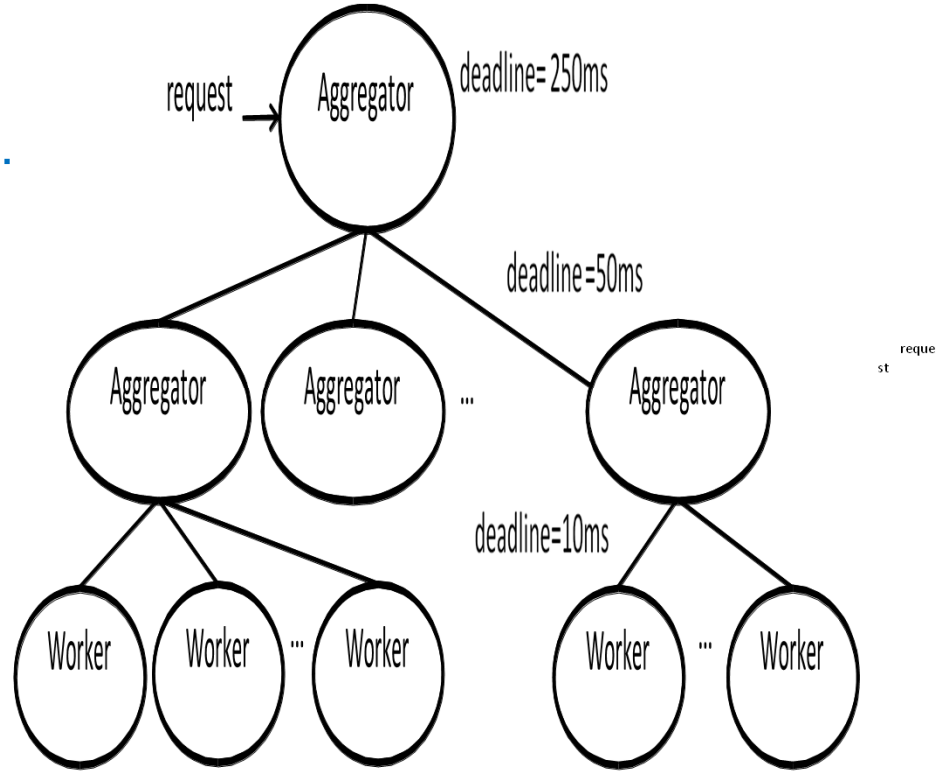
Soft real time applications, e.g.

- Web search
- Advertisement
- Retail

Data Center Networking Challenges

Partition / Aggregate pattern

.



Data Center Networking Challenges

An illustration:

Examples of requirements:

- Low latency
- High burst tolerance

Important: Many other applications with conflicting requirements reside in the same data center

Data Center Networking Challenges

Let us focus on transport layer protocols requirements

- High data rate support (Up to 100 GB/s)
- User Level Protocol Indicator Support
- QoS friendly
- Virtual cluster support
- Data center flow / cong. Control
- High availability
- Compatibility with TCP/IP base
- Protection against DoS

On Networking

References

1. A. Tanenbaum, Computer Networks, Fourth Edition, Prentice Hall, 2003 (Introduction)
2. V. Strivasta and M. Montani, Cross Layer Design: A Survey and The Road Ahead, IEEE Communications Magazine, December 2005, Vol. 43, Issue 12, pp. 112 – 119

Layered Architectures

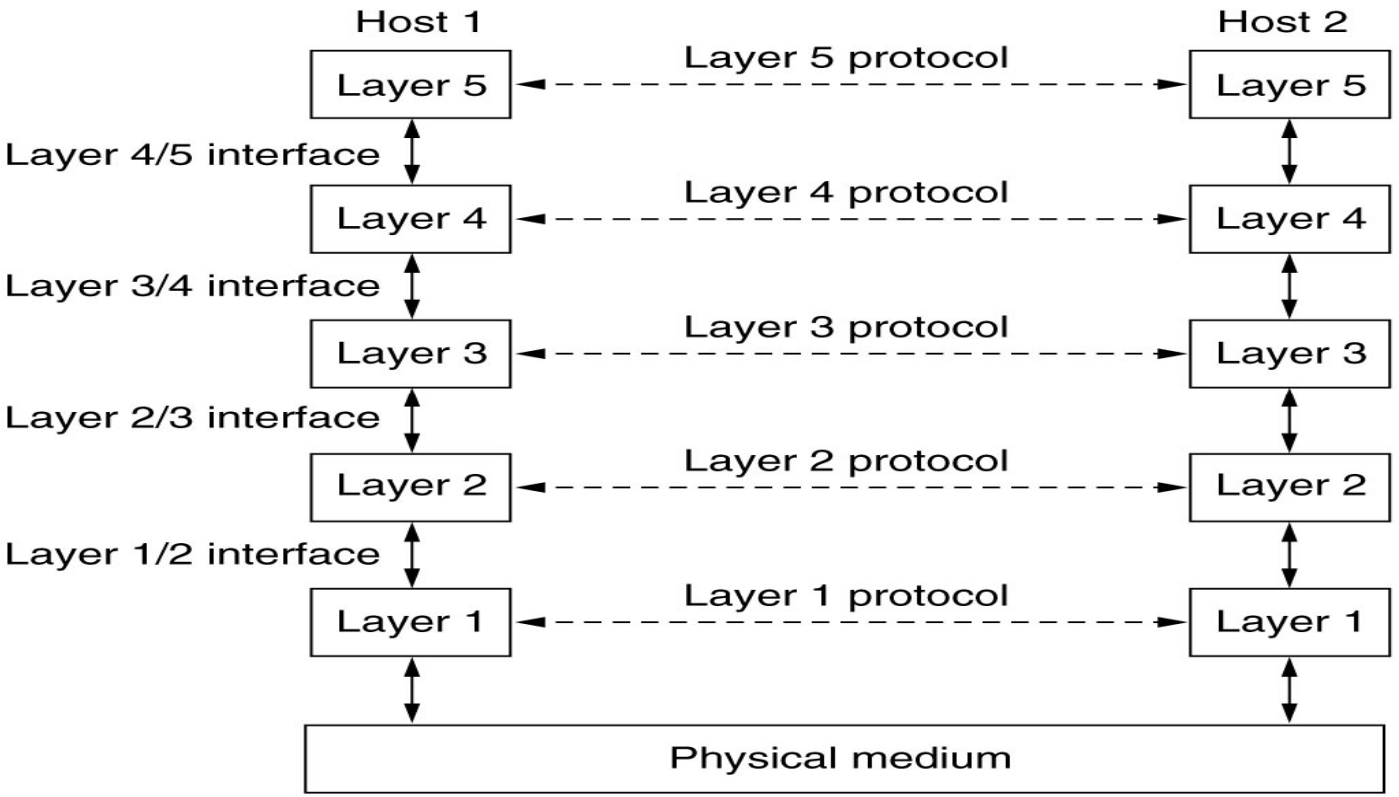


Figure 1.13 (Reference [1])

Layered Architectures

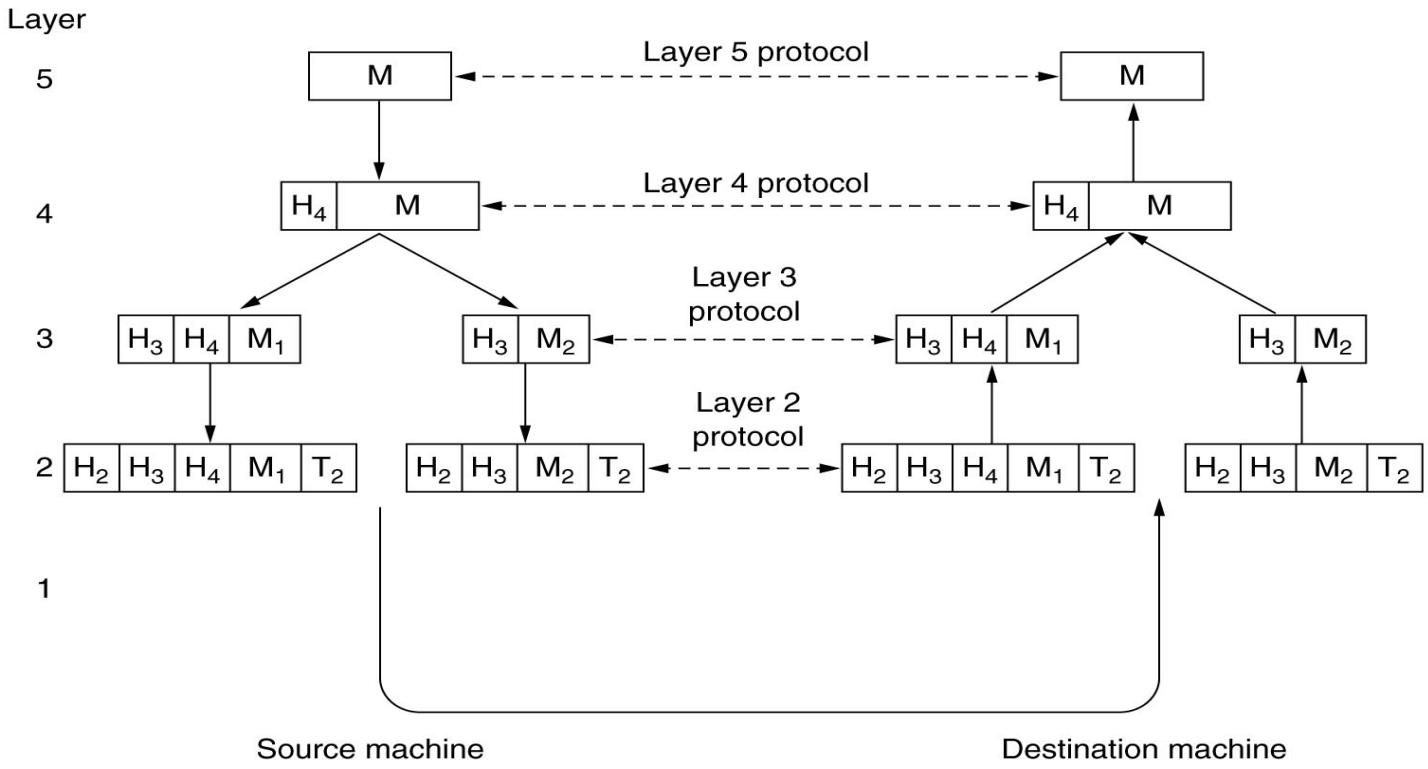


Figure 1.15 (Reference [1])

Cross Layered Architecture

- Definition of cross layer design
  - Violation of the principles of layered protocol architectures
    - Examples
      - Allowing communications between non adjacent layers
      - Sharing variables between layers
      - Designing protocols that span several layers

Cross Layered Architectures

Main motivation for cross layer design

- Performance improvements, especially in wireless environments
  - An example
    - TCP sender assumes packet errors are indicators of networks congestion and slow down sending rates
      - Case of wired links: true
        - » Need to slow down

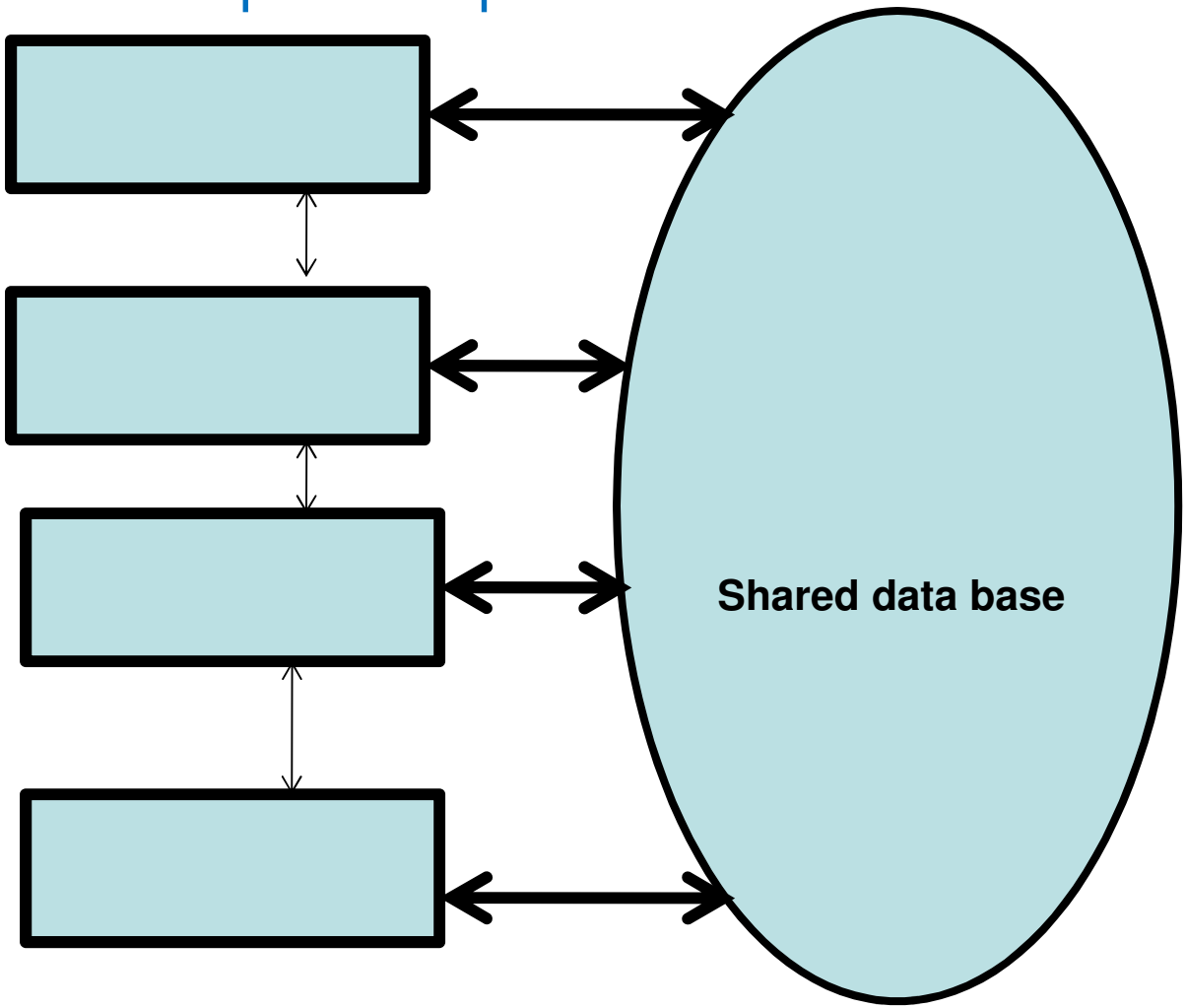
Cross Layered Architectures

Main motivation for cross layer design

- Performance improvements, especially in wireless environments
  - An example
    - Case of wireless links
      - » Not always true
      - » May be indicators of errors on physical and data link layers
      - » Information from physical and data link layers to transport layer (i.e. TCP) needed to make correct decision (i.e. slow down or speed up)

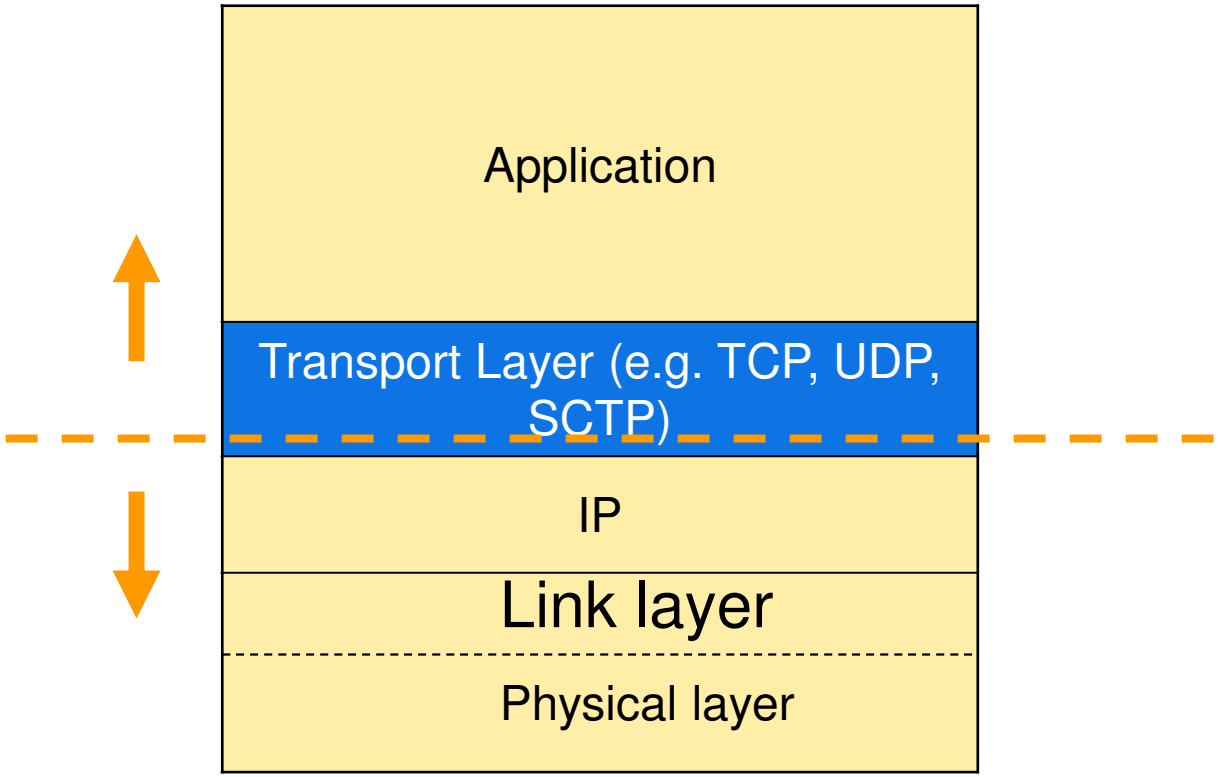
Cross Layered Architectures

A shared data base example of implementation



On Transport Layer

On The Transport Layer



On The Transport Layer

- Provide service to application layer by using the service provided by network layer
- Hide physical network
  - Hide processing complexity
  - Hide different network technologies and architectures
- Provides host-to-host transport

On The Transport Layer

- Addressing
- Connection Establishment
- Connection Release
- Flow Control
- Error Detection and Crash Recovery

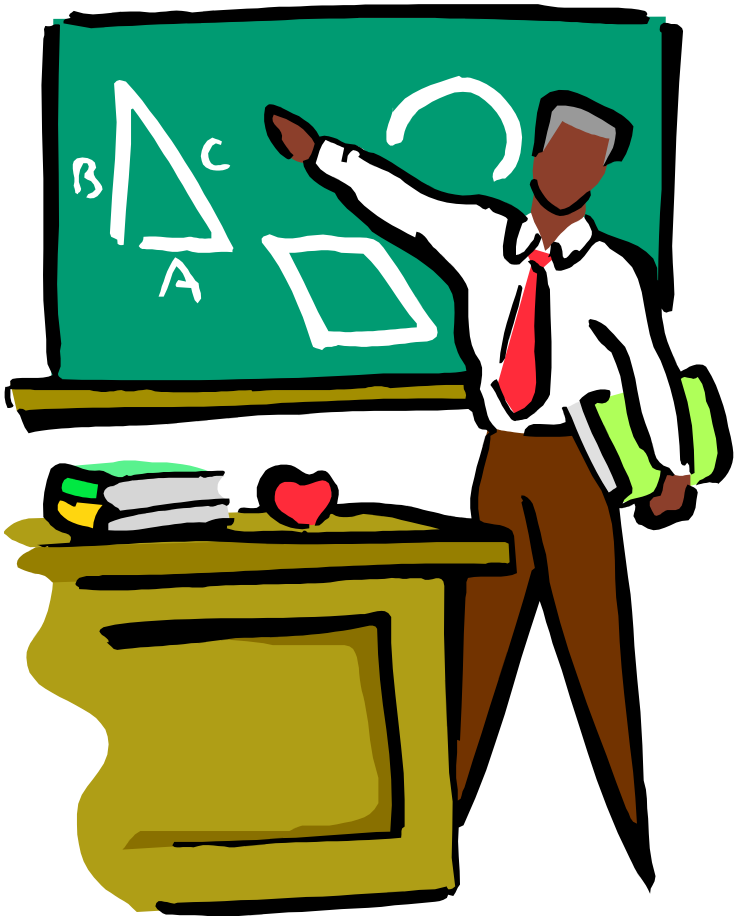
Traditional Transport Layers (Beyond TCP / UDP)

References

1, IETF RFC 3550, RTP / RTCP

2. A. Caro et al., SCTP: A Proposed Standard for Robust Internet Data Transport, IEEE Computer November 2003
3. S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004
4. P. Natarajan et al., SCTP: What, Why and How? IEEE Internet Computing, September / October 2009
5. Y-C Lai, DCCP: Transport Protocol with Congestion Control and Unreliability, IEEE Internet Computing, September / October 2008

The Other Transport Protocols



- 1 - Motivations and taxonomy
- 2 - Building on UDP: RTP / RTCP
- 3 - Building from scratch: SCTP
- 4 - Building from scratch: DCCP

Motivations and Taxonomy

Key characteristics of TCP

- Reliability
  - Three way handshake connection
  - Re-transmission
- Congestion control
  - Windows
    - Transmission rate reduction
- Uni-homing

Motivations and Taxonomy

Key characteristics of UDP

- No reliability
- No congestion control
- Uni-homing

Motivations and Taxonomy

The one size (either TCP or UDP) fits all philosophy does not always work

- What about
  - Applications requiring reliability but real time delivery (i.e. no retransmission)?
    - Interactive audio/video (e.g. conferencing)
  - Applications requiring more reliability than what is provided by TCP?
    - Multimedia session signalling
  - Applications requiring real time delivery, low reliability, but congestion control?
    - Multi party games

Motivations and Taxonomy

Two possible approaches

- Build a new transport protocol that complements / runs on top of existing transport protocols (e.g. UDP)
  - RTP/RTCP on top of UDP and application using RTP/RTCP
- Build a new transport protocol from scratch (i.e. runs on top of IP)
  - SCTP
  - DCCP

RTP / RTCP

Two complementary protocols

- Early 90s
- Primary goal: Real time media delivery with a focus on multimedia conferencing

Two complementary protocols

- Actual transportation of real time media
  - Real-time Transport Protocol (RTP)
- Control of transportation:
  - Real Time Transport Control Protocol (RTCP)

RTP / RTCP

Main characteristics

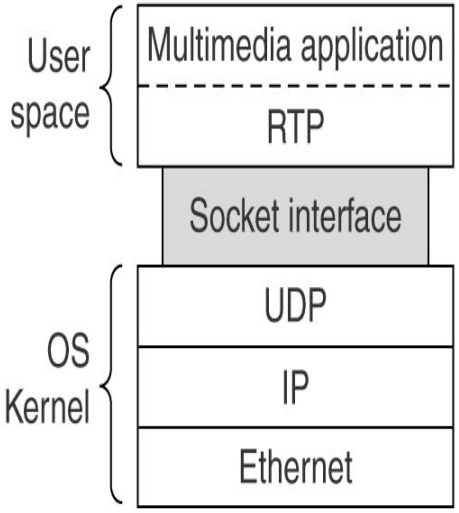
RTP:

- No provision for Quality of service
- No guarantee for out of sequence delivery
- Typically runs on top of UDP but may run on top of other protocols

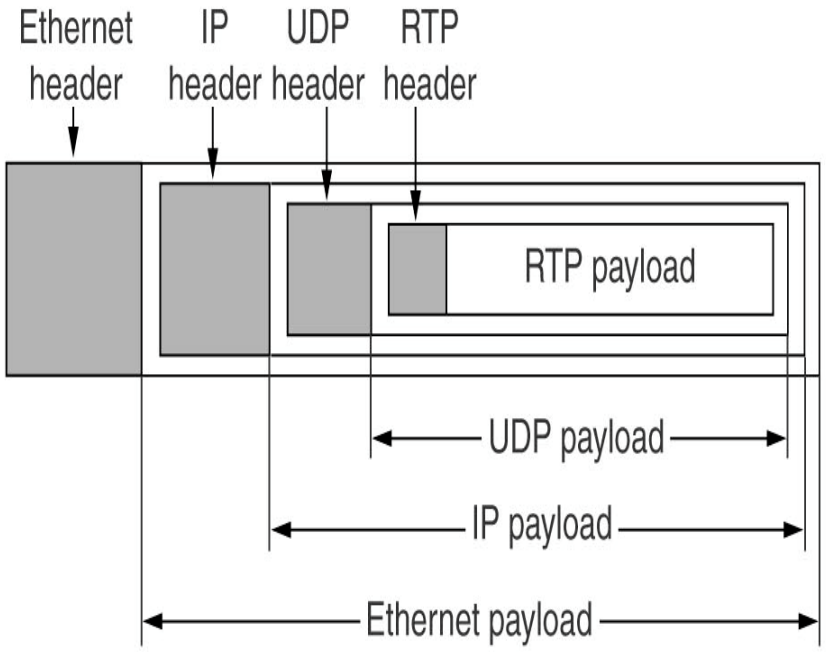
RTCP:

- Help in providing control by providing information on packets sent, received
  - Information may be used by application to build whatever it thinks is necessary (e.g. reliability, congestion control)

RTP


User space
Multimedia application
RTP
Socket interface
OS Kernel
UDP
IP
Ethernet

(a)


Ethernet header
IP header
UDP header
RTP header
RTP payload
UDP payload
IP payload
Ethernet payload

(b)

DCCP A DCCP B ------ NA NB ------ +-----------------+ +-+ +-+ +-----------------+ |(1) Initiation | | | | | | | |DCCP-Request --> +--+-+---X| | | | |<-+-+----+-+--+<-- DCCP-

RTP

Mixers / translators

- Intermediate systems
- Connect 2 or more transport level clouds
  - End systems
  - Mixers / translators
- Use cases
  - Centralized conference bridges
  - Heterogeneous conferences
    - Low speed connection
    - High speed connection
    - Different encoding schemes
  - Some participants behind firewalls

RTP

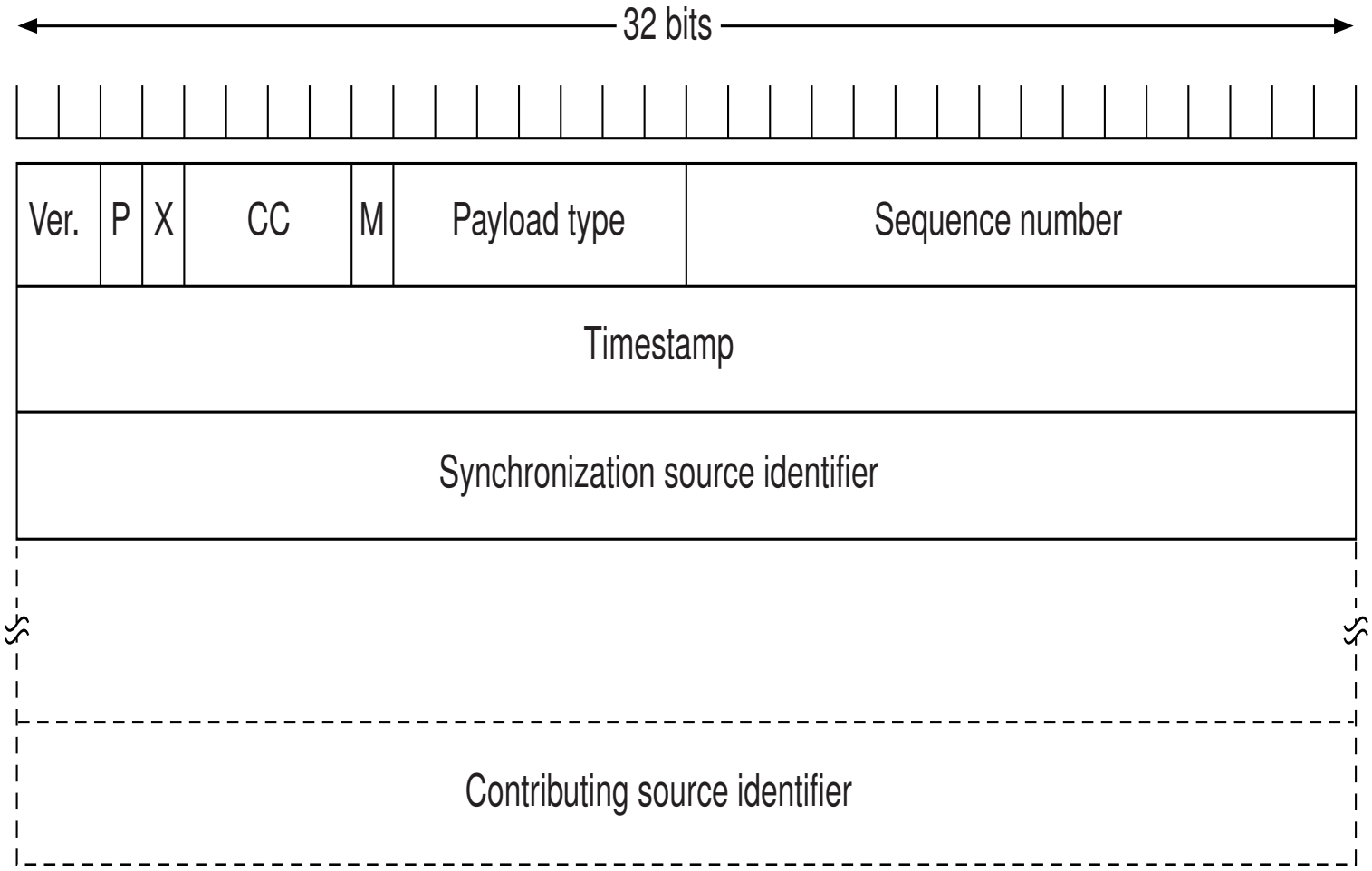
Synchronization source (SSRC)

- Grouping of data sources for playing back purpose (e.g. voice vs. video)
- An end system can act as several synchronization sources (e.g. IP phone with video capabilities)
- Translators forward RTP packets with their synchronization source intact

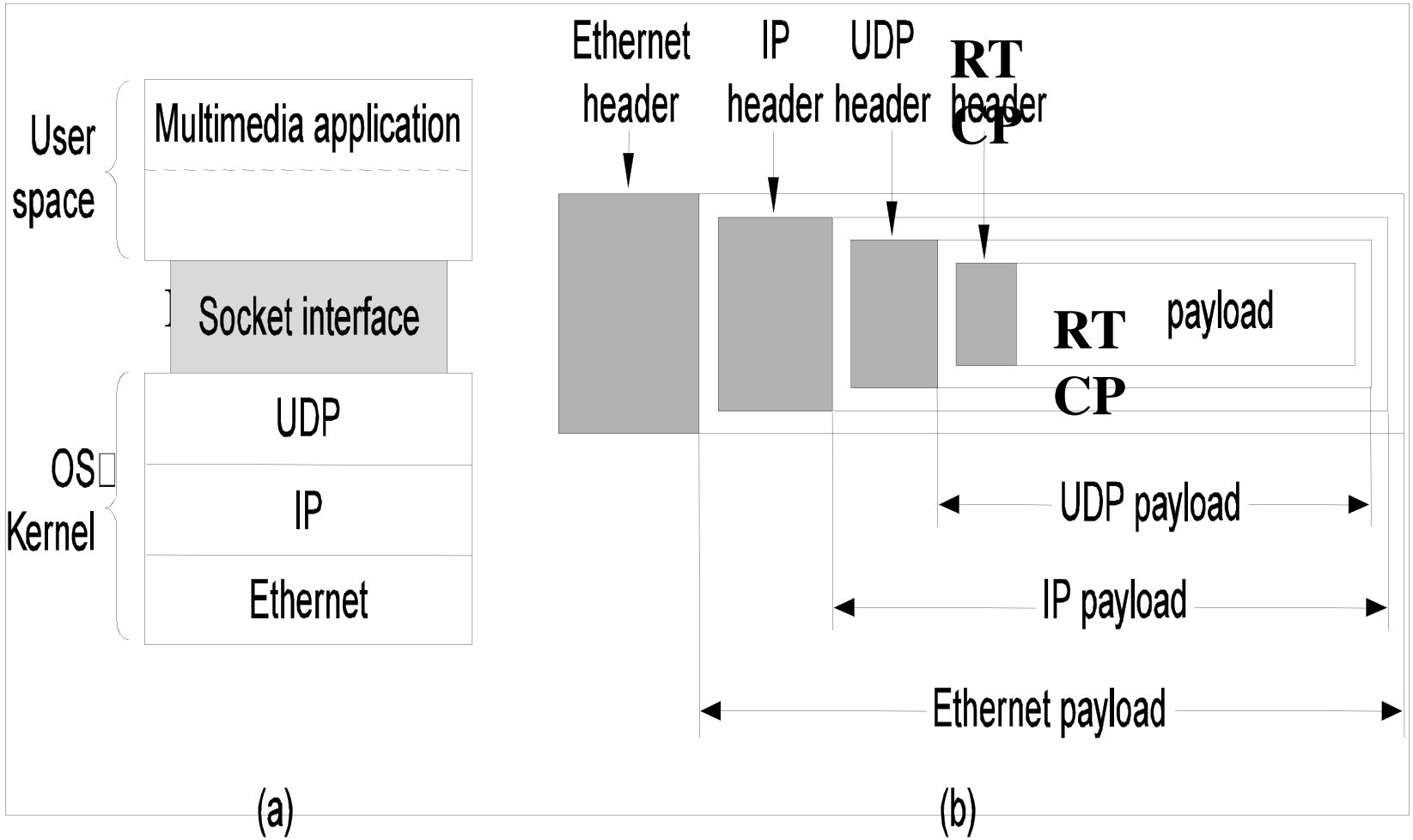
Contributing source (CSRC)

- A source of a stream of RTP packets that has contributed to the combined stream produced by an RTP mixer
- Mixers insert the list of contributing sources in the packets they generate

RTP


32 bits
Ver.
P
X
CC
M
Payload type
Sequence number
Timestamp
Synchronization source identifier
Contributing source identifier

RTCP


User space
Multimedia application
Socket interface
OS Kernel
UDP
IP
Ethernet
(a)
Ethernet header
IP header
UDP header
RT header
RTCP
payload
RTCP
UDP payload
IP payload
Ethernet payload
(b)

RTCP concepts

Monitor:

- Application that receives RTCP packets sent by participants in an RTP session

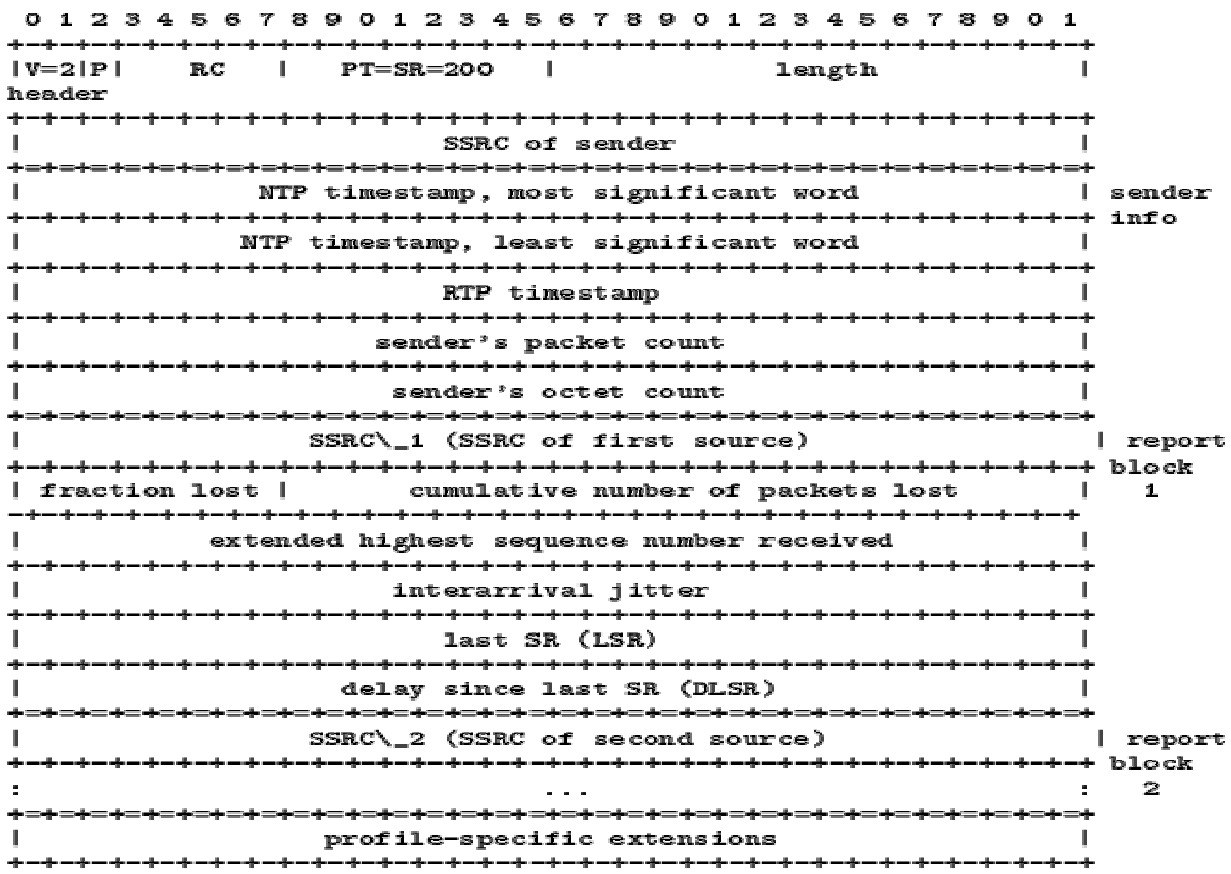
Reports

- Reception quality feedback
- Sent by RTP packets receivers (which may also be senders)
  - May be used to build reliability, congestion control or whatever the application deems necessary

RTCP packets

- Receiver report
  - Version
- Time stamp
- Sender's packet count
- Reception report blocks

RTCP packets

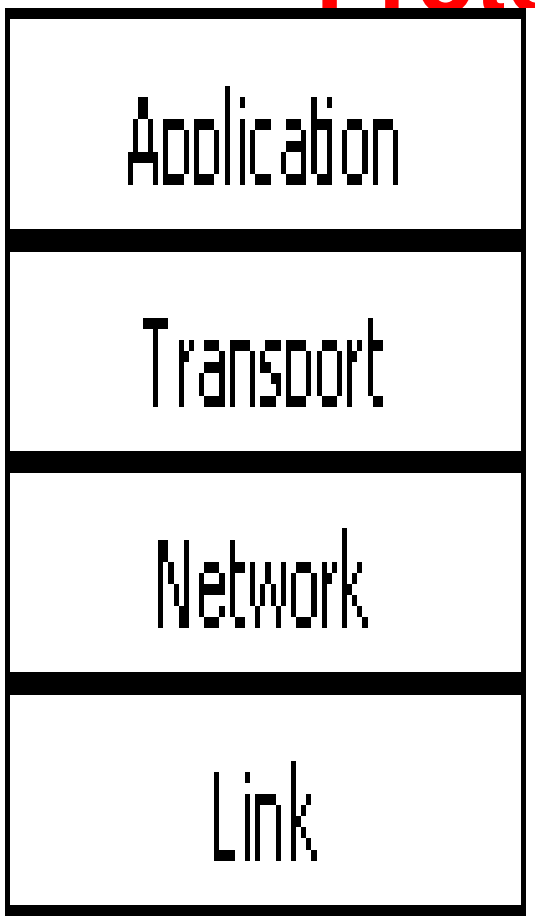


Stream Control Transmission Protocol (SCTP)

Designed in early 2000s to carry multimedia session signaling traffic over IP, then subsequently extended to meet the needs of a wider range of application

- Design goals much more stringent than TCP design goals (e.g. redundancy, higher reliability)
- Offer much more than TCP
- A sample of additional features
  - Four way handshake association instead of three way handshake connection
  - Multi-homing instead of uni-homing
  - Multi-streaming instead of uni-streaming

Stream Control Transmission Protocol



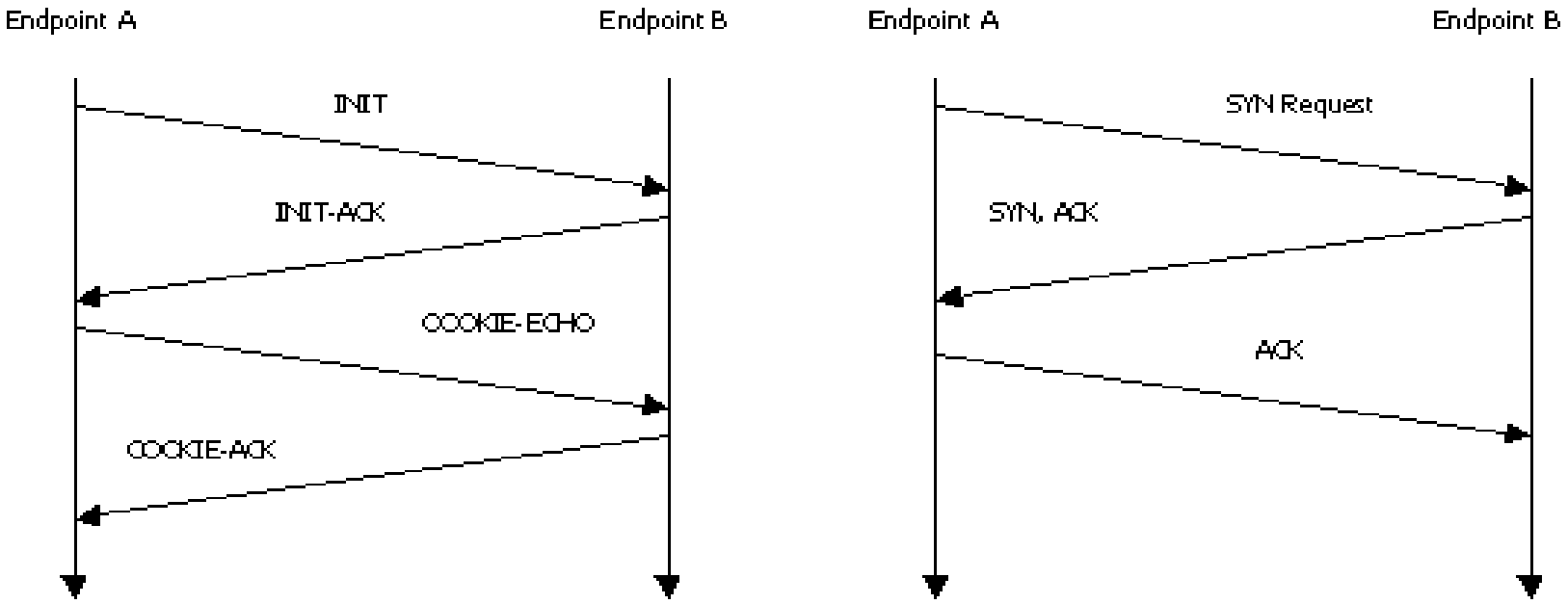
SCTP, TCP, UDP

IP

Four way handshake

Why?

- Key reason: Make SCTP resilient to denial of service (DOS) attacks, a feature missing in TCP



SCTP

TCP

Multi-homing

Why?

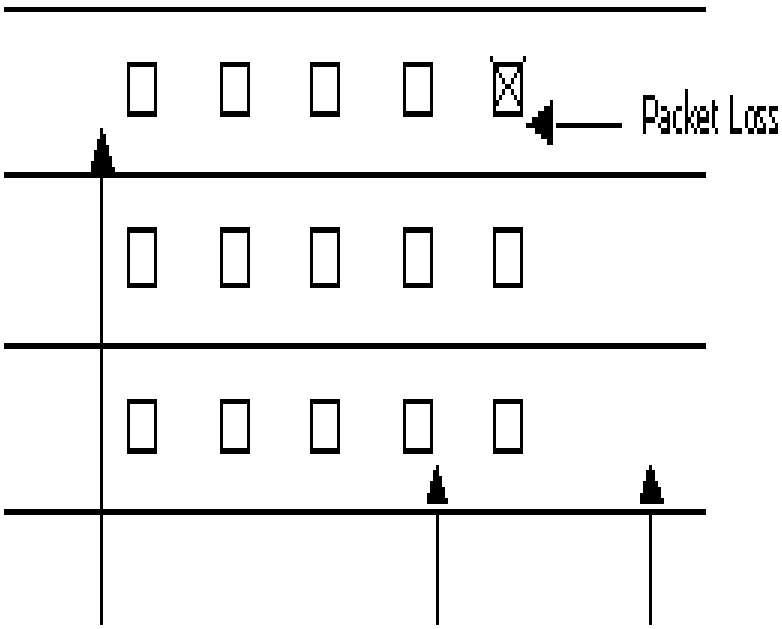
- Key reason: Make SCTP resilient in resource failures, a feature missing in TCP (High availability)
  - Multi-homed host: Host accessible via multiple IP addresses
  - Use cases
    - Subscription to multiple ISP to ensure service continuity when of the ISP fails
    - Mission critical systems relying on redundancy
    - Load balancing

Multi-homing

Why?

- Key reason: Make SCTP resilient in resource failures, a feature missing in TCP
  - Multi-homing with SCTP (only for redundancy)
    - Multi-homed host binds to several IP addresses during associations unlike TCP which binds to a single IP address
      - Retransmitted data is sent to an alternate IP address
      - Continued failure to reach primary address leads to the conclusion that primary address has failed and all traffic goes to alternate address

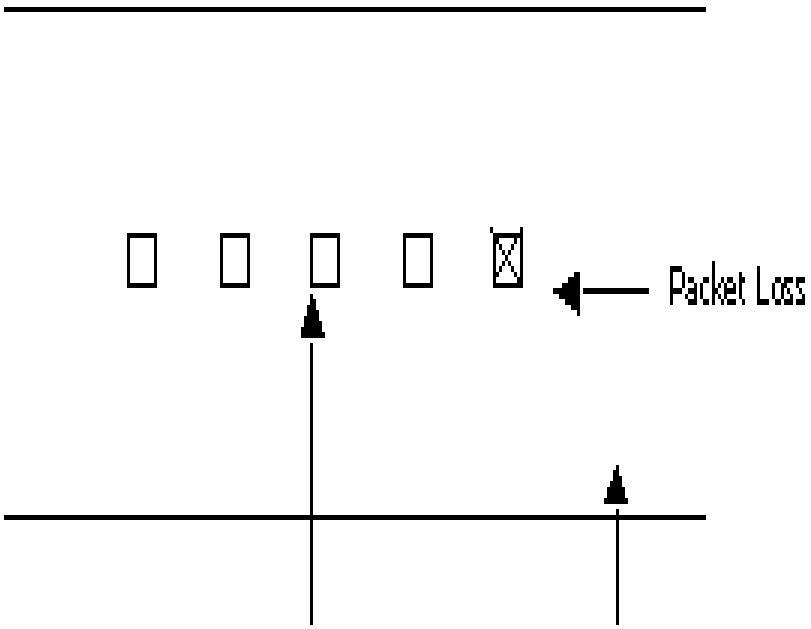
Multi-streaming



Only data packets in this stream are blocked. Remaining streams continue to send data normally

Data Packet

SCTP Stream



Data packets blocked by packet loss up ahead. Head of Line Blocking occurs in entire connection.

TCP Stream

Data Congestion Control Protocol (DCCP)

One of the most recent transport protocols (Second half of the 2000s)

- Primary goal:
  - Delivery of real time media (somehow similar to the goal assigned to RTP / RTCP)
- Build on the experience acquired in protocol design / deployment since the design of RTP / RTCP (ie. Early 1990s)
  - Some examples of improvements:
    - Congestion control incorporated in the transport protocol (unlike RTP/RTCP)
    - Possibility to avoid DoS

Overall view

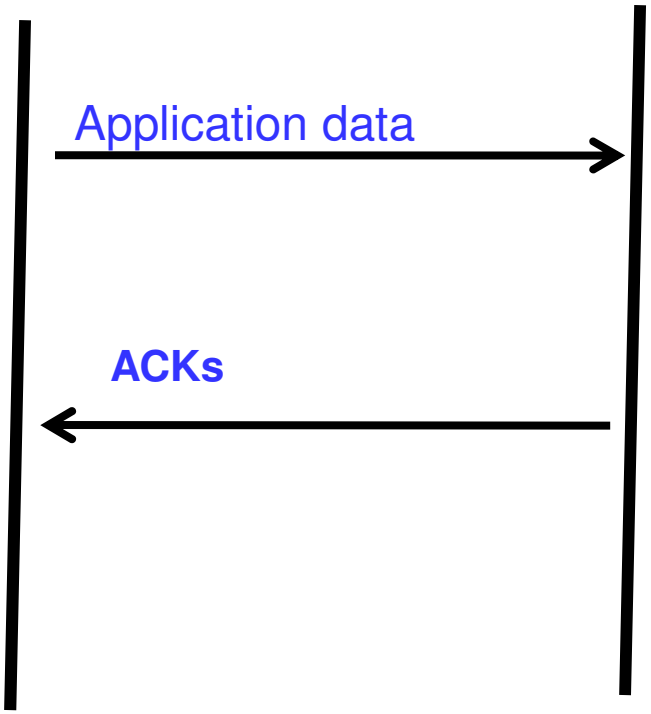
- Three way handshake connection like TCP
  - In-built possibility to use cookies during response phase to avoid DoS
  - A connection can be seen as two half-connections (i.e. uni-directional connections)
    - Possibility for a receiver to send only ACK
- Reliable connection establishment and feature negotiation
- Unreliable data transfer (no retransmission)
- Feature negotiation

The protocol states

```
Client                                          Server
------                                           ------
                (0) No connection
CLOSED                                          LISTEN

                (1) Initiation
REQUEST      DCCP-Request -->
                      <-- DCCP-Response     RESPOND
PARTOPEN     DCCP-Ack or DCCP-DataAck -->

                (2) Data transfer
OPEN          <-- DCCP-Data, Ack, DataAck -->      OPEN

                (3) Termination
                      <-- DCCP-CloseReq     CLOSEREQ
CLOSING      DCCP-Close -->
                        <-- DCCP-Reset          CLOSED
TIMEWAIT
CLOSED
```

Half connection

Use case: Unidirectional streams (e.g. Streaming applications)



-

Data transfer

- Packets have sequence numbers
  - Client – server and server – client sequence numbers are independent
    - Tracking on both sides is possible
- Acknowledgements report last received packet
- Data drop option
  - Examples
    - Application not listening
    - Receiver buffer
    - Corrupt
  - May help in selecting congestion control mechanism

Data transfer

- Packets have sequence numbers
  - Client – server and server – client sequence numbers are independent
    - Tracking on both sides is possible
- Acknowledgements report last received packet
- Data drop option
  - Examples
    - Application not listening
    - Receiver buffer
    - Corrupt
  - May help in selecting congestion control mechanism

Feature negotiation

- Enable dynamic selection of congestion mechanism
  - Data drop option may help
    - Tracking on both sides is possible
  - TCP congestion control may be used
  - Other mechanisms may also be used

Traditional Transport Protocols vs. Challenges

References

1. K. Kant, Towards a Virtualized Data Center Transport Protocol, Infocom Workshop, 2008

Traditional Transport Protocols vs. Challenges (Ref. 1.)

Feature	TCP	SCTP	IBA
Scalability to 100 Gb/s	difficult	difficult	Easy?
Msg. based & ULP support	No	Yes	Yes
QoS friendly transport?	No	No	Yes
Virtual cluster support	No	No	limited
DC centric flow/cong. control	No	No	limited
Power aware transmission	Limited	limited	No
High availability features	Poor	Fair	Fair
Compatible w/ TCP/IP base	Yes	Yes	No
Protection against DoS attacks	Poor	Good	No

The End