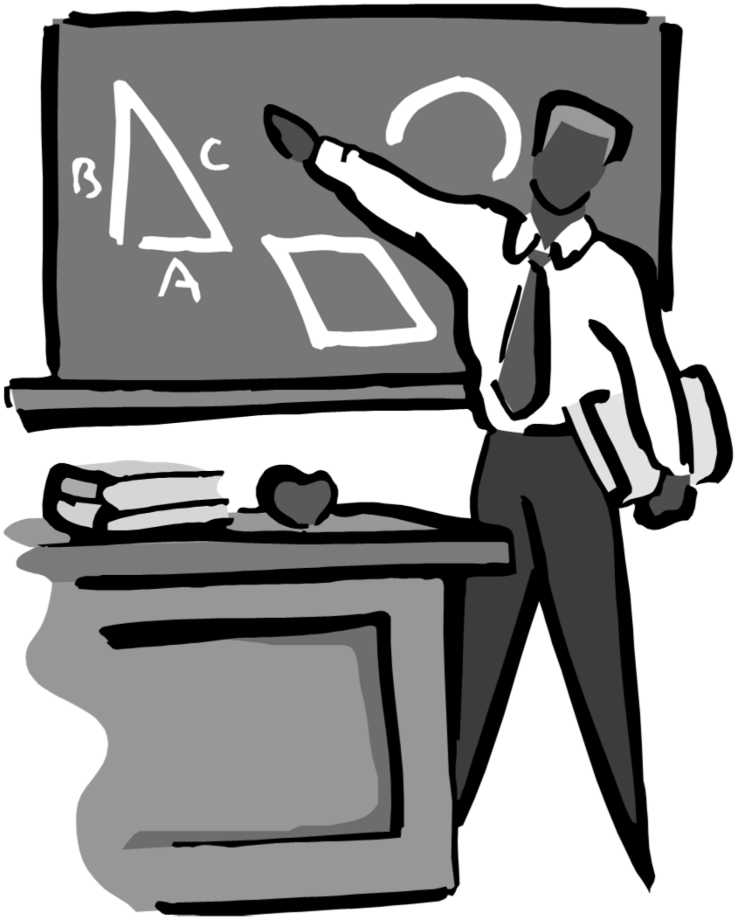
Chapter VI
The Other Transport Protocols

The Other Transport Protocols



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

Motivations and Taxonomy

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 more reliability than what is provided by TCP?
    - Multimedia session signalling
  - Applications requiring real time delivery, low reliability, but congestion control?
    - Video conferences, 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)
  - Build in user space
    - RTP/RTCP on top of UDP and application using RTP/RTCP
    - QUIC on top of UDP
- Build a new transport protocol from scratch (i.e. runs on top of IP)
  - Build in operating system kernel space
    - SCTP
    - DCCP

Building on UDP: RTP / RTCP

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

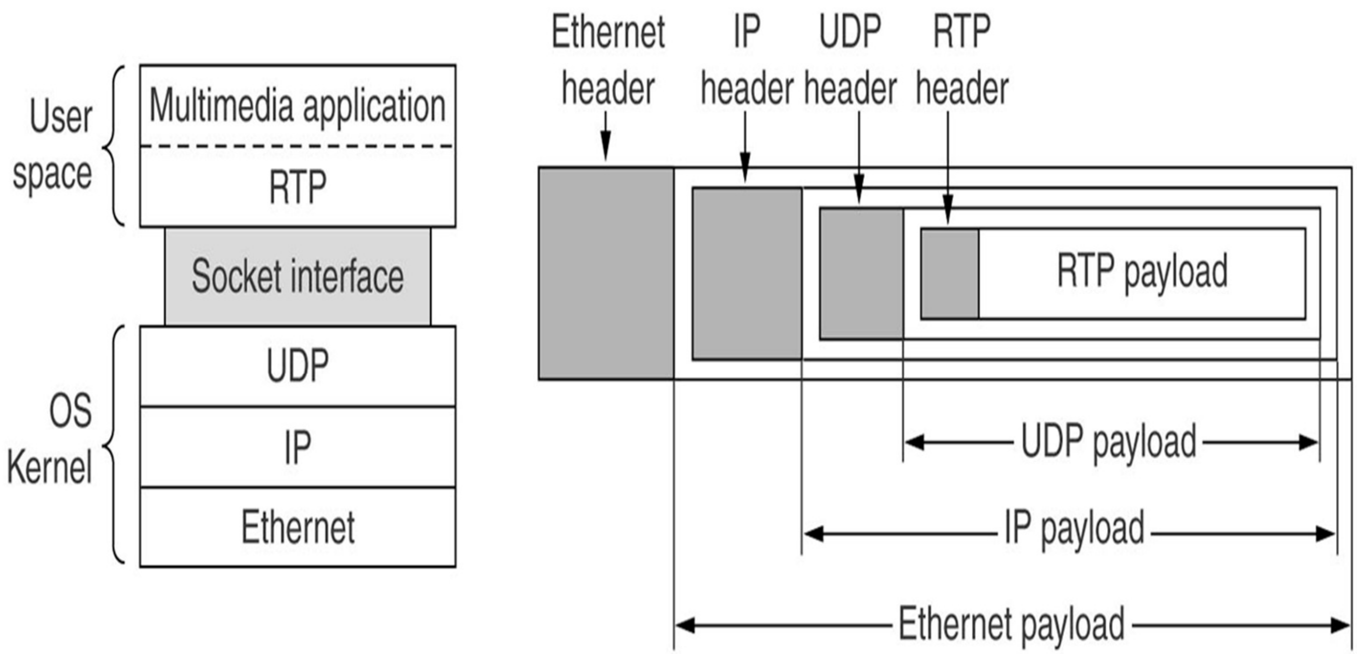
RTP:

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

(a) (b)

.

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

RTP

Mixers / translators

- Intermediate systems
  - 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

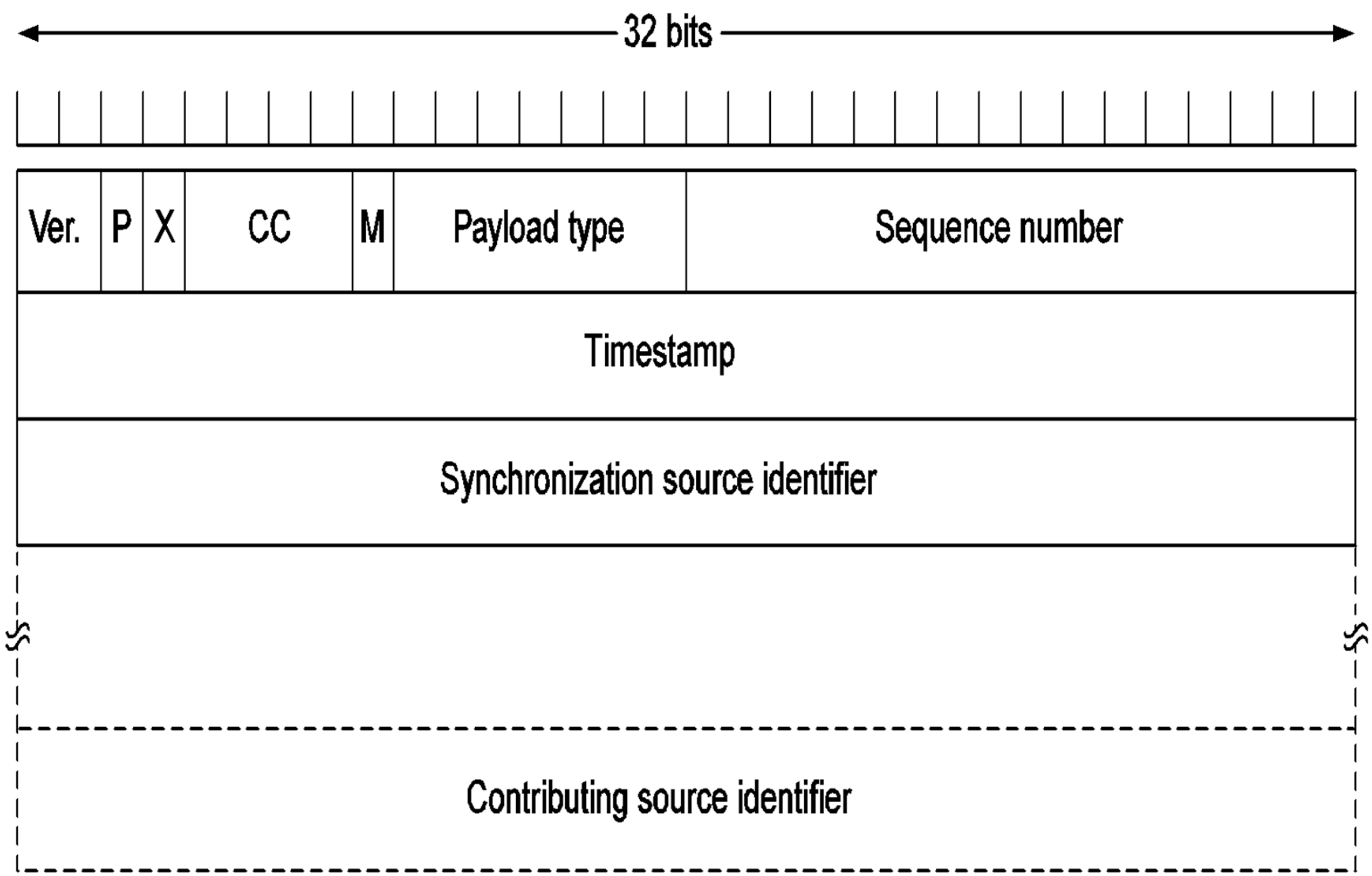
Synchronization source (SSRC)

- 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

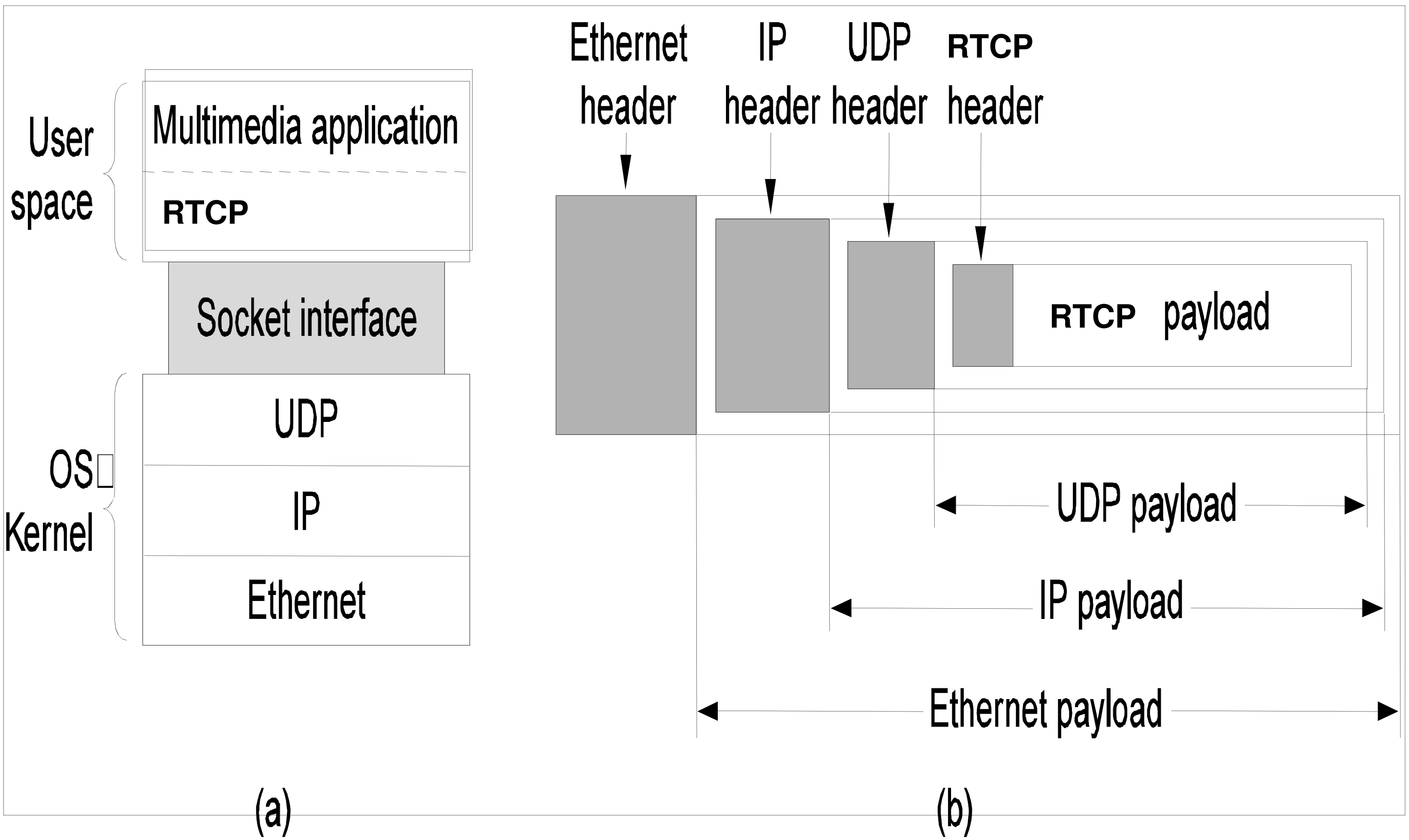
Contributing source (CSRC)

- 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
RTCP
Socket interface
OS Kernel
UDP
IP
Ethernet
Ethernet header
IP header
UDP header
RTCP header
RTCP payload
UDP payload
IP payload
Ethernet payload
(a)
(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

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|    RC   |   PT=SR=200   |             length            |
header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SSRC of sender                        |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|              NTP timestamp, most significant word             | sender
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ info
|             NTP timestamp, least significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP timestamp                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     sender's packet count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      sender's octet count                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC\_1 (SSRC of first source)                | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
| fraction lost |       cumulative number of packets lost       |   1
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           extended highest sequence number received           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC\_2 (SSRC of second source)               | report
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
:                               ...                             :   2
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                  profile-specific extensions                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

.

Building on UDP: QUIC

References:

1. Y. Cui et al, Innovating Transport with QUIC: Design Approaches and Research Challenges, IEEE Internet Computing, March April 2017

2. A. Langley et al., The QUIC Transport Protocol: Design and Internet Scale Deployment, Sigcomm 2017

Quick UDP Internet Connection (QUIC)

One of the most recent effort in transport protocol design

- Still under standardization (IETF)
- Initially designed by Google
  - First experimental deployment in 2013
  - Now runs on most Google clients (e.g. Chrome) and servers (e.g. Youtube)
    - 7% of Internet Traffic
    - 30% of Google Egress traffic

Quick UDP Internet Connection (QUIC)

Motivations

- Problems related to the use of TCP with HTTPs
  - TCP handshake delay
  - Head-of -Line Blocking delay
    - A loss of a segment blocks all the other segments which arrives till the lost segment is received
  - Coupling of TCP with the operation system
    - Design in user space vs. design in OS kernel space

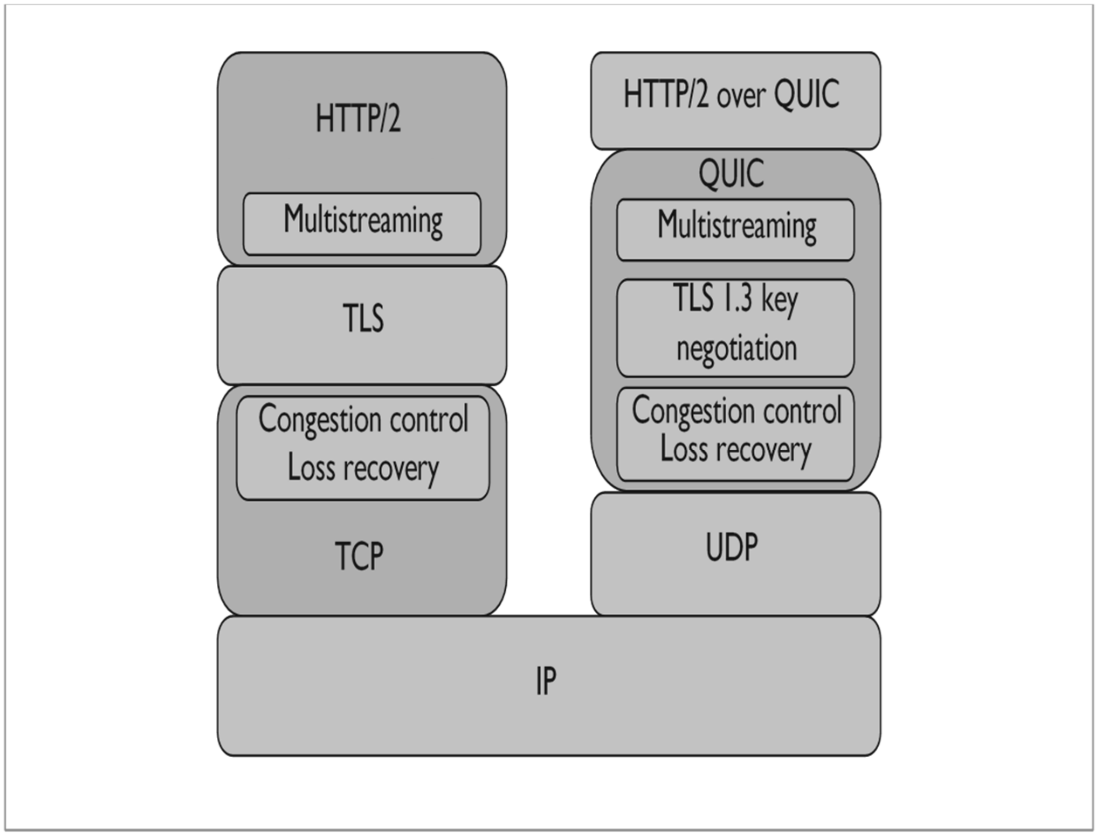
Quick UDP Internet Connection (QUIC)

Overview

- On Transport Layer Security (TLS) / Secure Socket Layer (SSL)
  - Client server protocol Used with HTTP
  - Ensures
    - Authentication
    - Integrity
    - Confidentiality

Quick UDP Internet Connection (QUIC)

Overview



Y. Cui et al, Innovating Transport with QUIC: Design Approaches and Research Challenges, IEEE Internet Computing, March April 2017

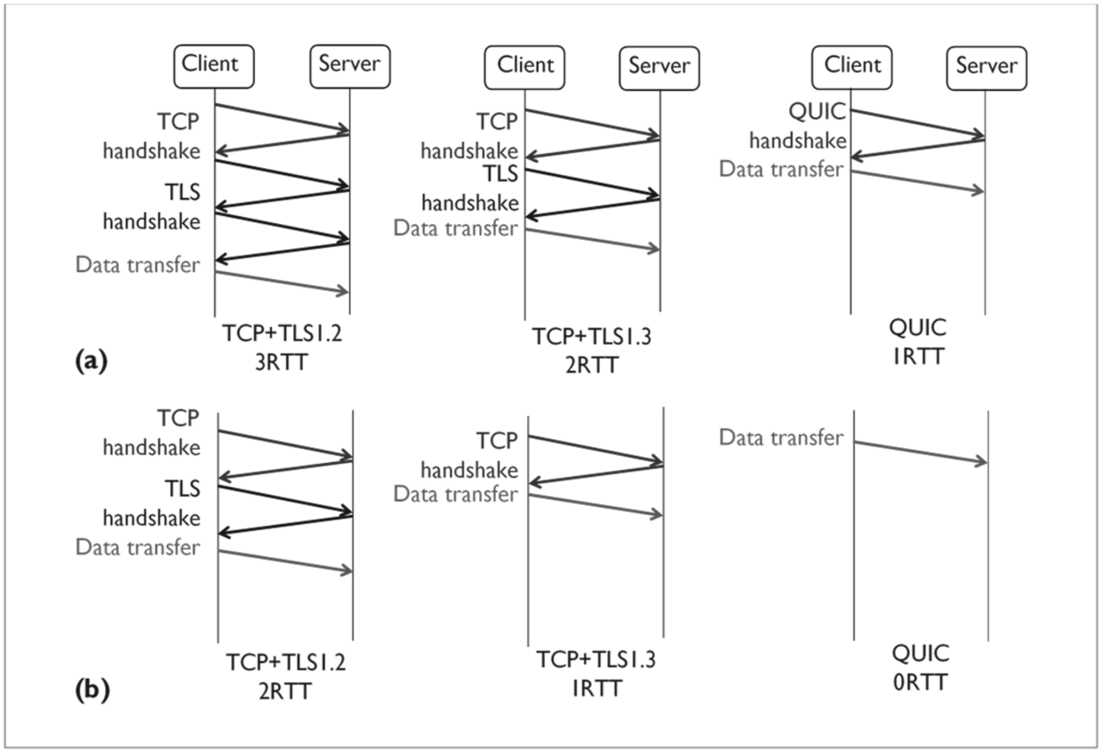
Quick UDP Internet Connection (QUIC)

Key features

- Fast connection establishment
- Multi-streaming

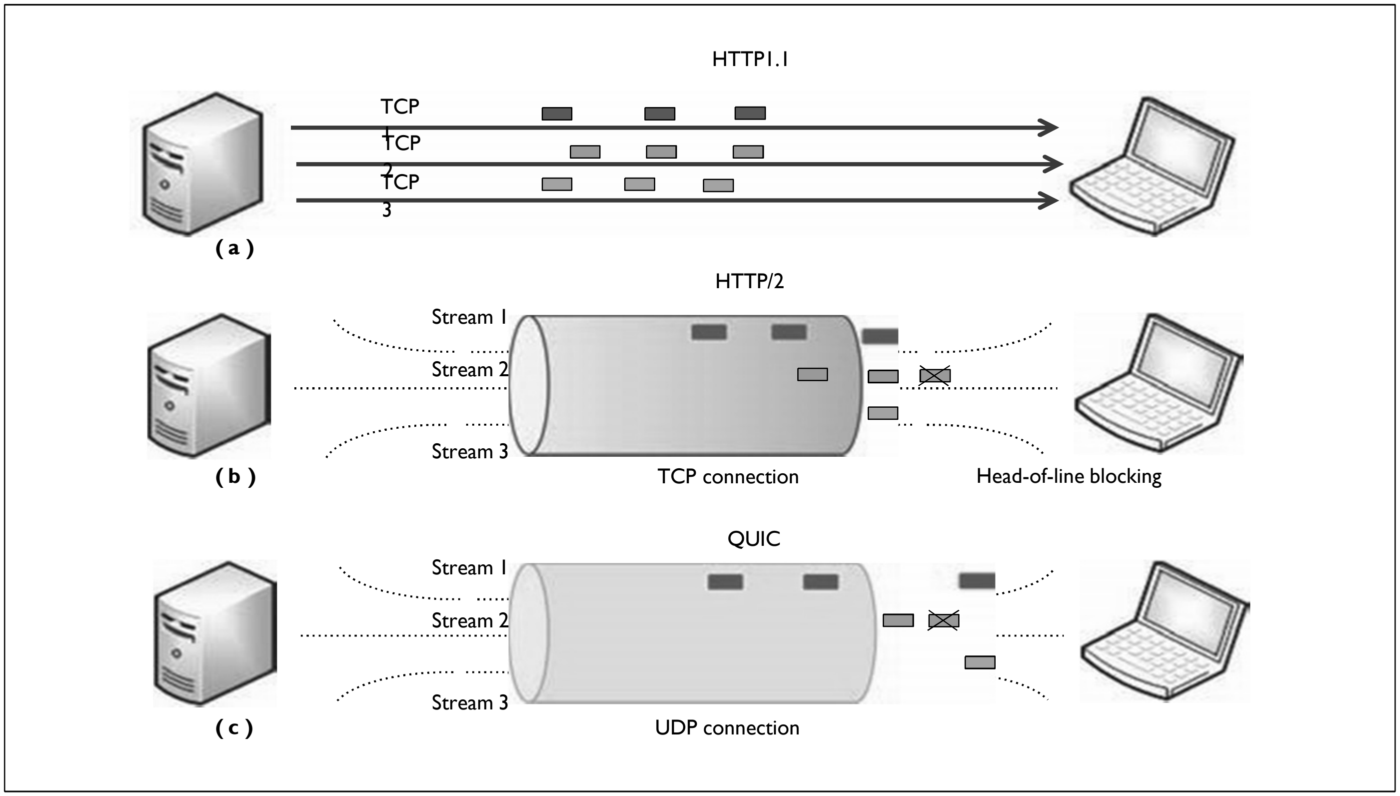
Quick UDP Internet Connection (QUIC)

Fast connection establishment



Quick UDP Internet Connection (QUIC)

Multi-streaming



Building from scratch: SCTP

References

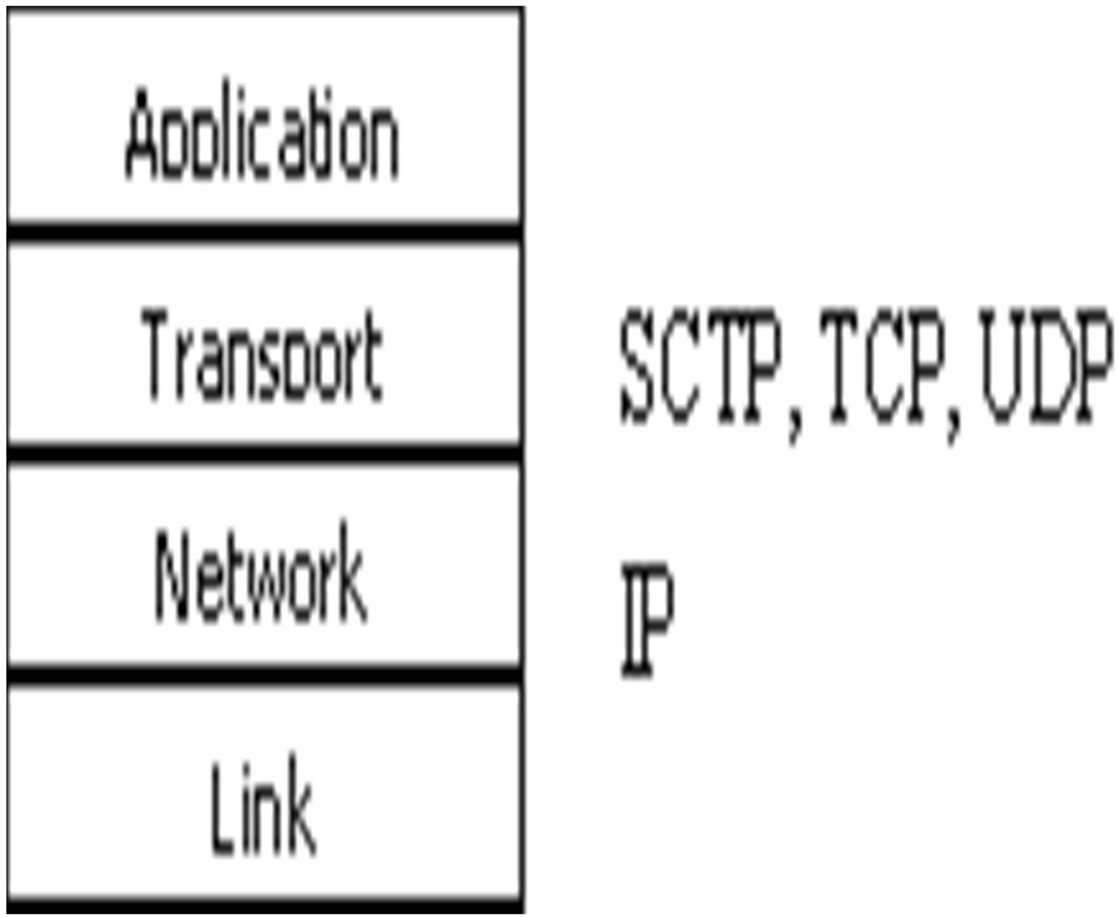
2. A. Caro et al., SCTP: A Proposed Standard for Robust Internet Data Transport, IEEE Computer November 2003
3. S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004
4. P. Natarajan et al., SCTP: What, Why and How? IEEE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

Stream Control Transmission Protocol (SCTP)

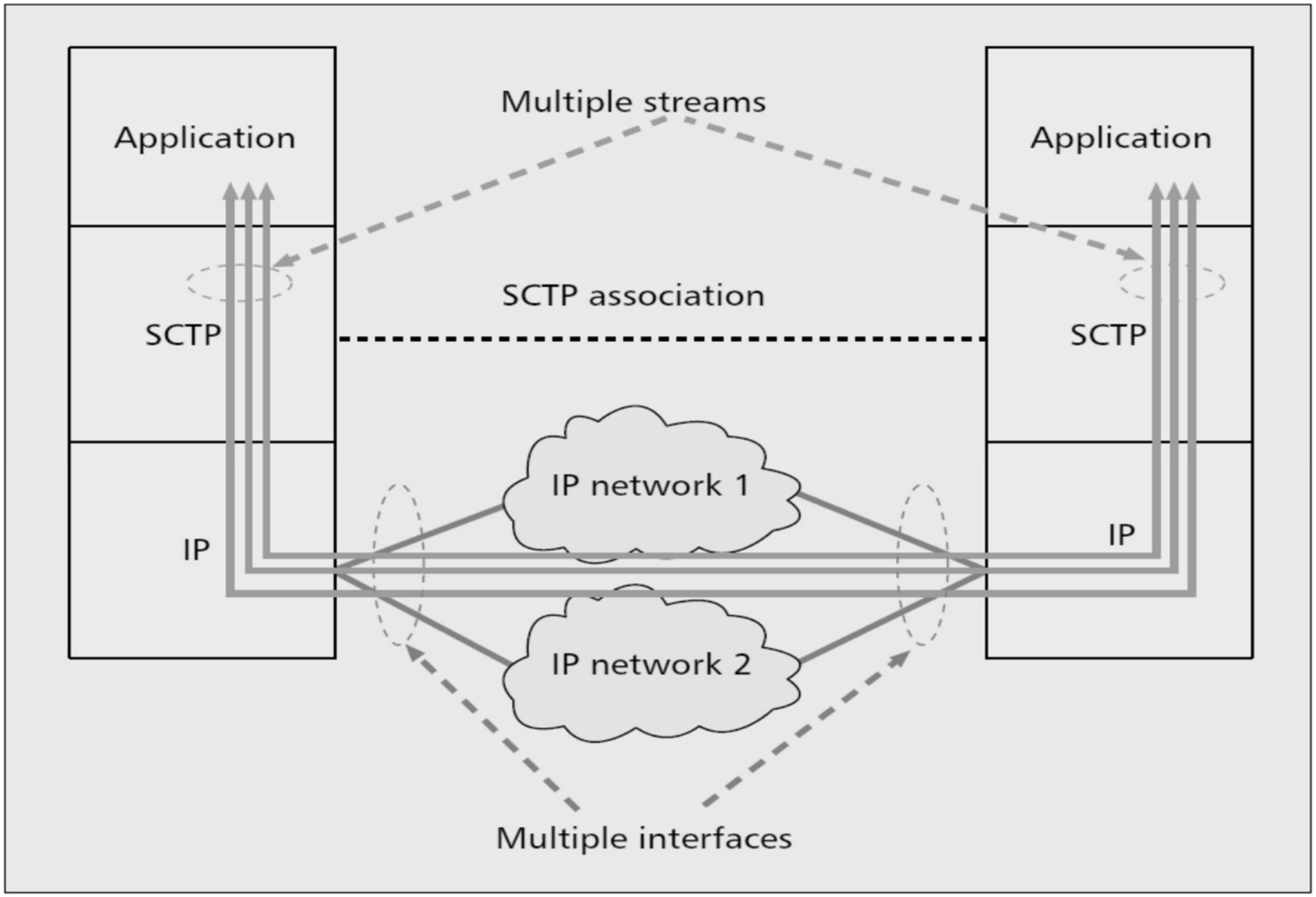
Designed in early 2000s to carry multimedia session signaling traffic over IP, then subsequently extended to meet the needs of a wider range of application

- Design goals much more stringent than TCP design goals (e.g. redundancy, higher reliability)
- Offer much more than TCP
- A sample of additional features
  - 4 Way handshake instead of 3 way handshake
  - Multi-homing instead of uni-homing
  - Multi-streaming instead of uni-streaming

Overview



Overview



■ **Figure 1** *A schematic view of an SCTP association*

Four way handshake

Why?

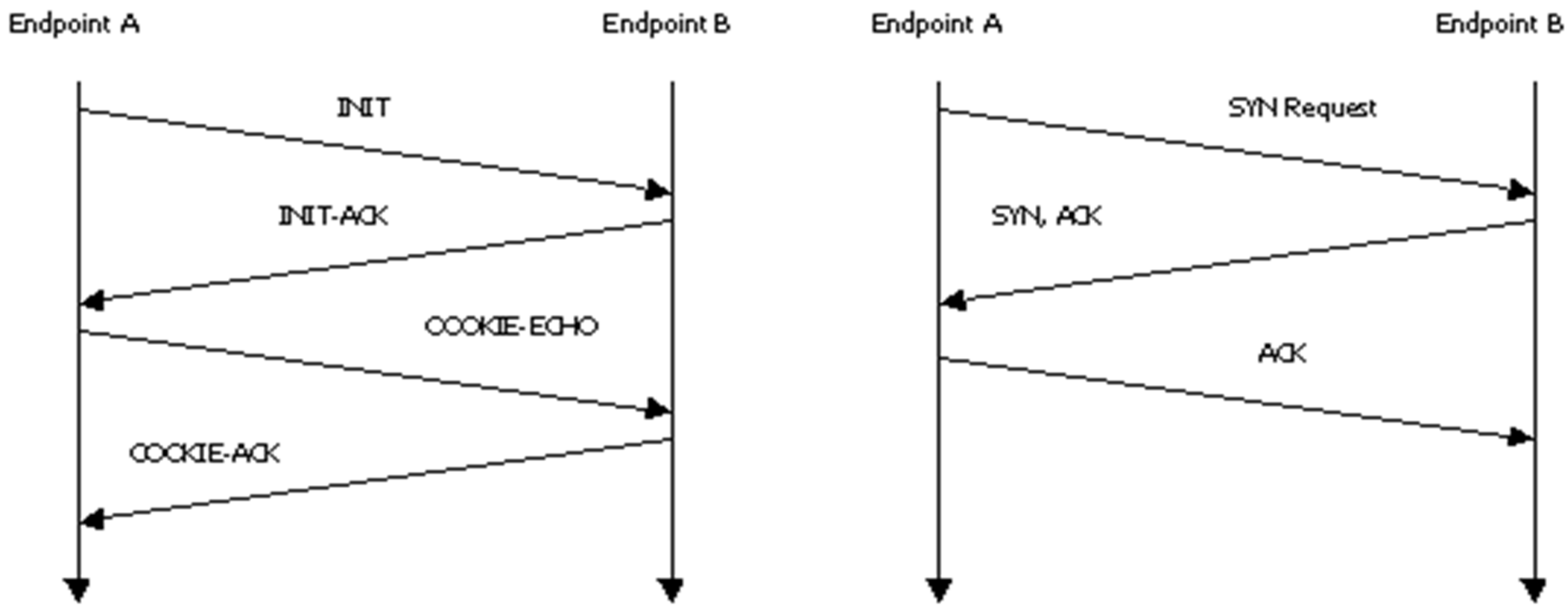
- **Key reason: Make SCTP resilient to denial of service (DoS) attacks, a feature missing in TCP**
  - DoS (SYN attack in the case of TCP)
    - Root cause: TCP maintains in memory useless state information regarding each pending connection
    - Memory get eventually exhausted
  - Potential solution: 4 way handshake
- --

SCTP **TCP**

Four way handshake

Why?

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

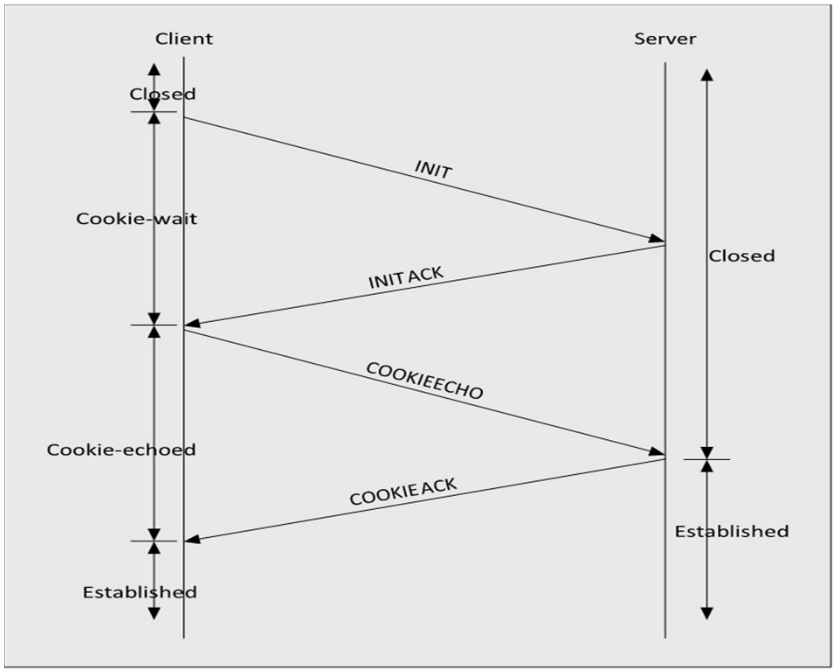


SCTP **TCP**

Four way handshake

Why?

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



S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004

Multi-homing

Why?

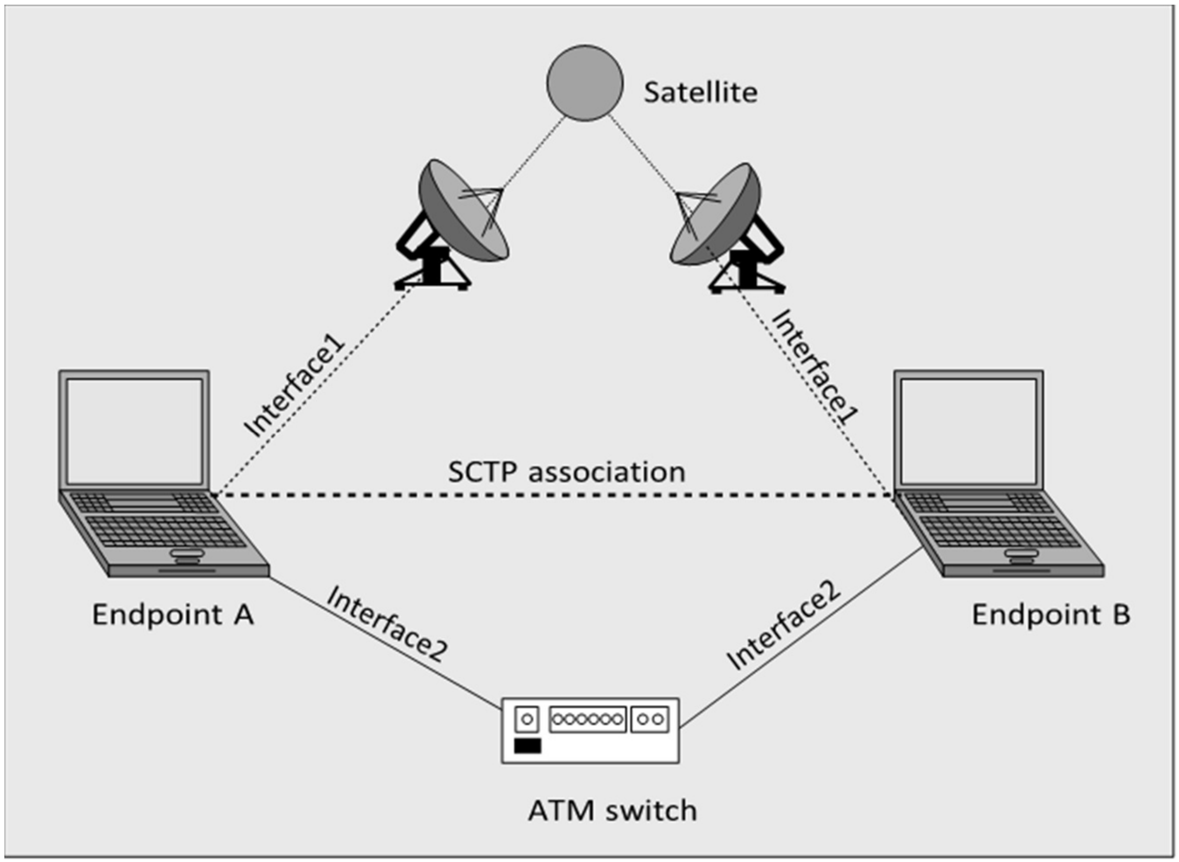
- Key reason: Make SCTP resilient in resource failures, a feature missing in TCP
  - Multi-homed host: Host accessible via multiple IP addresses
  - Use cases
    - Subscription to multiple ISP to ensure service continuity when of the ISP fails
    - Mission critical systems relying on redundancy
  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

Multi-homing

Why?

- Multi-homing with SCTP (Redundancy use case)
  - 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-homing

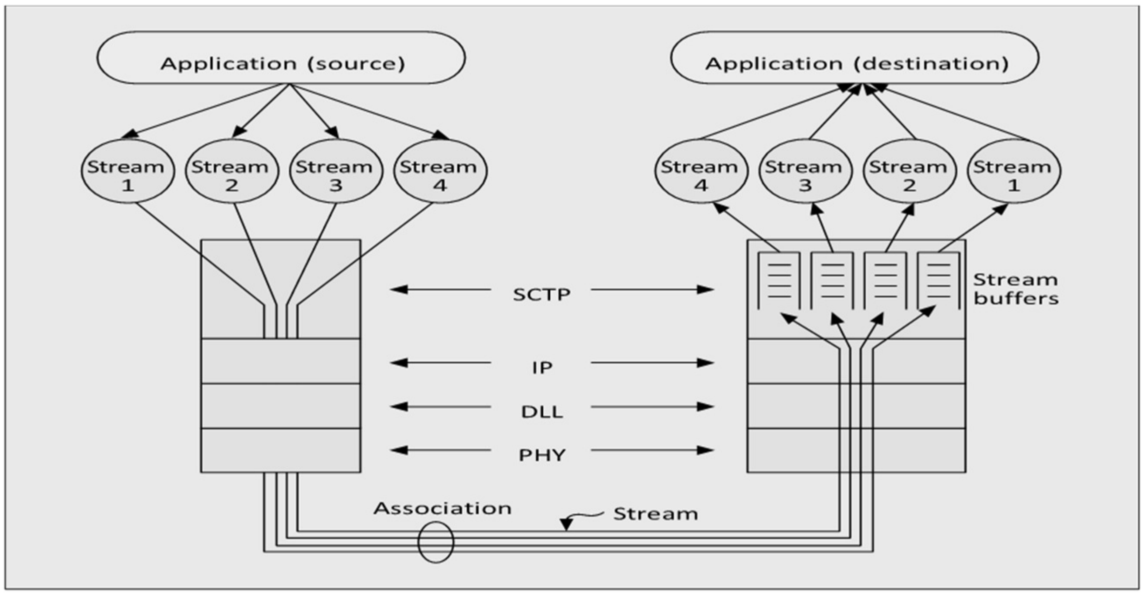


S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004

Multi-streaming

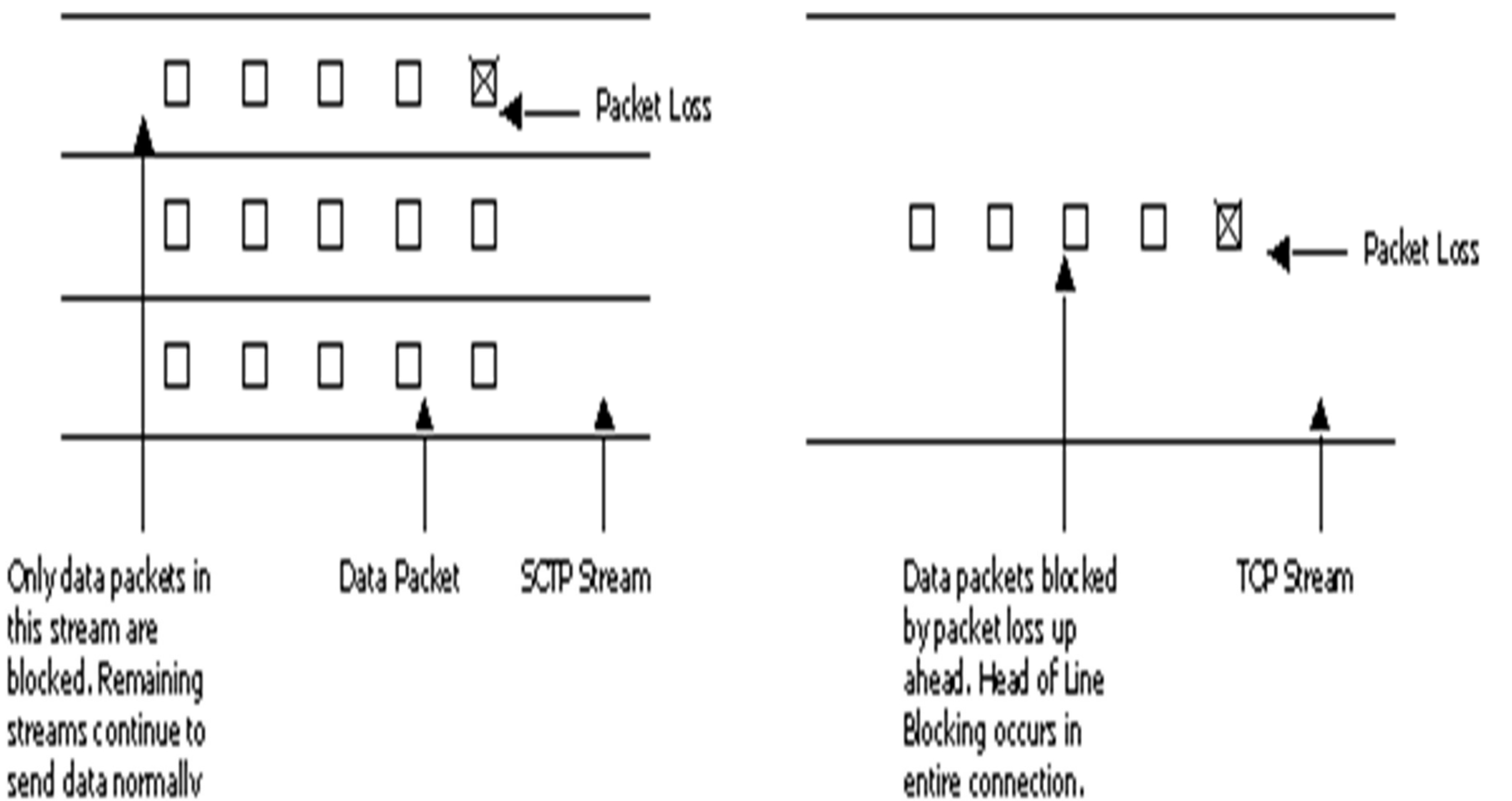
- Data from the application layer is multiplexed onto the association
  - Sequencing done within a stream
  - Segment lost within a stream is fully handled within that stream without affecting the other streams, i.e.
    - Segments following the lost one are stored / queued until the lost one is received

Multi-streaming



S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004

Multi-streaming



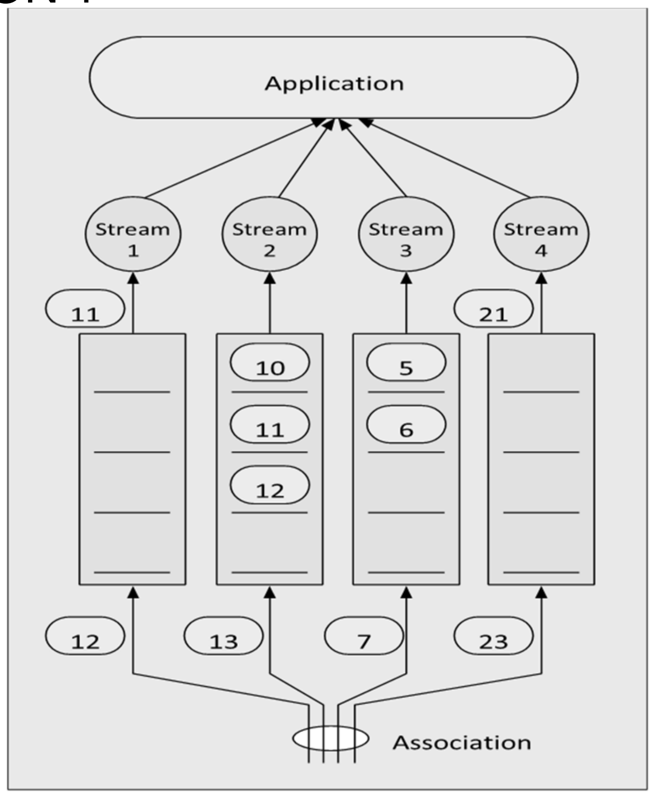
Multi-streaming

ILLUSTRATION 1

- Application with 4 streams (Stream 1, stream 2, stream 3, stream 4)
- Assumptions
  - Stream 1
    - SSN 11 has been delivered and SSN 12 arrives
  - Stream 2
    - SSN 9 is lost
  - Stream 3
    - SSN4 of stream 3 is missing
  - Stream 4
  - 21 has been delivered and 23 arrives

Multi-streaming

ILLUSTRATION 1



S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004

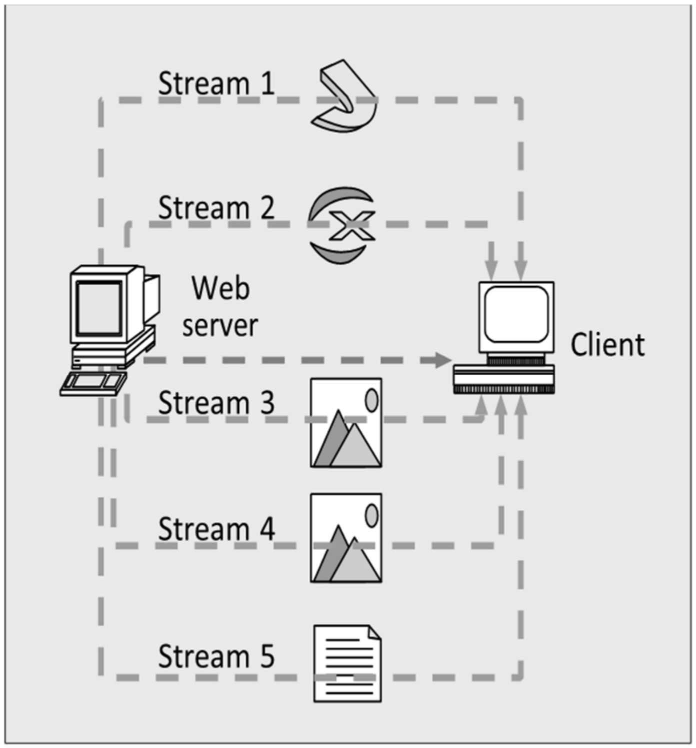
Multi-streaming

ILLUSTRATION 2

- Web browsing
  - HTML page split in four:
    - Java applet
    - Active X control
    - Two images
    - Plain text

Multi-streaming

ILLUSTRATION 2



S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004

Stream Control Transmission Protocol

Table 1. *Comparison of TCP and SCTP.*

Protocol		TCP		SCTP
Setup messages		Three-way hands	hake	Four-way handshake
Shutdown messag	es	Four-way hands	hake	Three-way handshake
Half-open support		Supported		Not supported
Ordered delivery		Strict ordered		Ordered within a stream
Unordered delivery		Not supported		Supported
Message boundary		No boundary		Boundary preserved
		Stream-oriented		Message-oriented
Multihoming		Not supported		Supported
SACK support		Optional		Mandatory
Keep-alive heartbe	at	Optional		Mandatory
Heartbeat interval		≥ Two hours		30 seconds by default

S. Fu and M. Atiqzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Communications Magazine, April 2004

Building from scratch: DCCP

References

1. Y-C Lai, DCCP: Transport Protocol with Congestion Control and Unreliability, IEEE Internet Computing, September / October 2008
2. E. Kohler et al., Designining DCCP: Congestion Control Without Reliability, Sigcomm 2006

Data Congestion Control Protocol (DCCP)

Relatively "new" (Second half of the 2000s)

- Main goal
  - Delivery of real time media (somehow similar to the goal assigned to RTP / RTCP)
  - Suitable for applications such as:
    - Voice Over IP
    - Video conferencing
    - Online games
    - Video on demand

Data Congestion Control Protocol (DCCP)

Relatively "new" (Second half of the 2000s)

- Target applications require:
    - Real time delivery
    - Unreliability (No re-transmission)
    - Delay sensitivity

Data Congestion Control Protocol (DCCP)

Two main functions:

1. Establishment, management and tear of unreliable connections

2. Unreliable data transfer but with congestion control
   -

Data Congestion Control Protocol (DCCP)

Connection Establishment (3 Way Handshake but with built-in features to avoid DoS attacks)

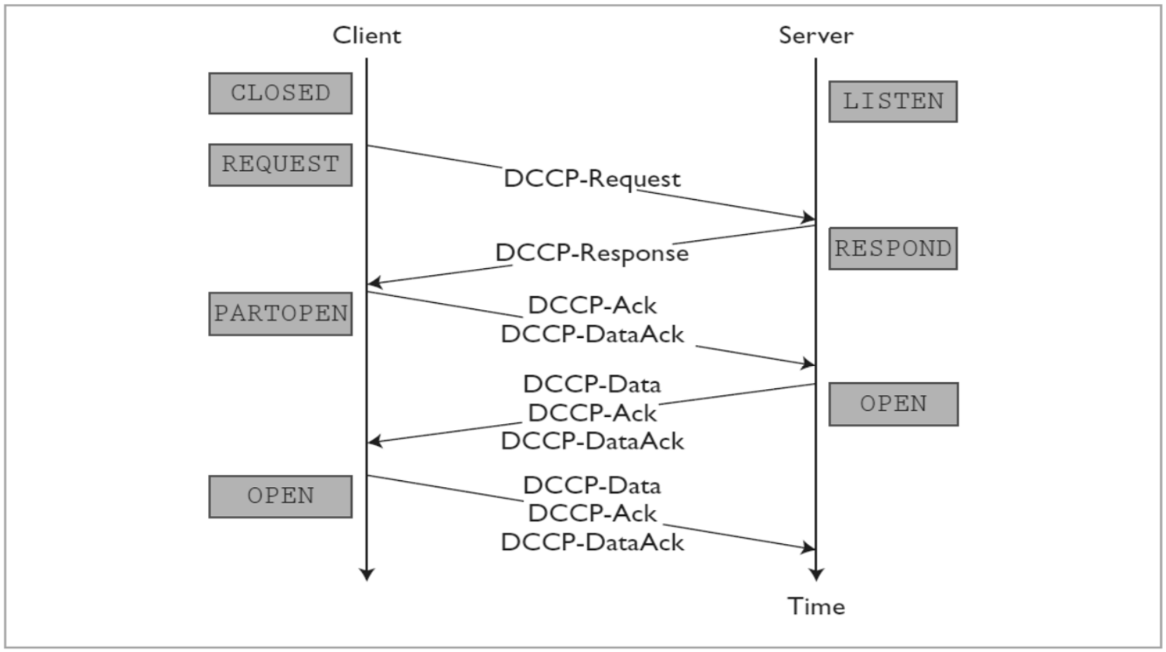
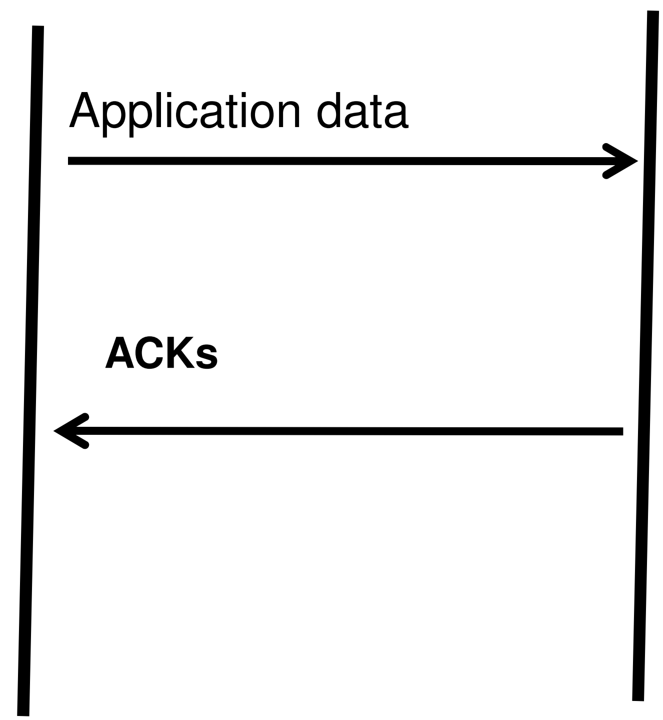


Figure 2. Datagram Congestion Control Protocol connection establishment. DCCP uses a three-way handshake to establish a connection.

Y-C Lai, DCCP: Transport Protocol with Congestion Control and Unreliability, IEEE Internet Computing, September / October 2008

Data Congestion Control Protocol (DCCP)

Note: A connection is a set of two unidirectional half-connections.
Possibility of Unidirectional streams (e.g. Streaming applications)



Data Congestion Control Protocol (DCCP)

Data transfer

- **Enabling congestion control**
  - Packets have sequence numbers
    - Client – server and server – client sequence numbers are independent
      - Tracking on both sides is possible
  - Acknowledgements report last received packet
- **Congestion control mechanisms**
  - Several options including a TCP like option

Data Congestion Control Protocol (DCCP)

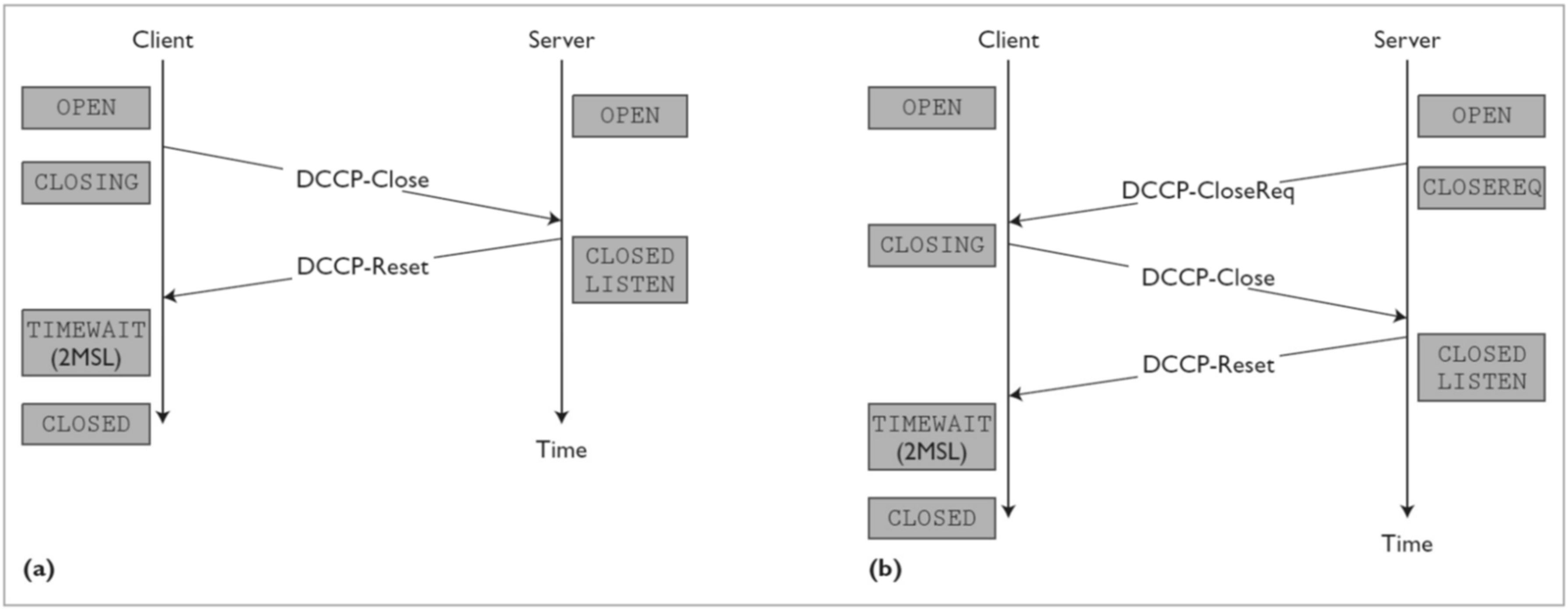


Figure 4. Datagram Congestion Control Protocol connection termination. DCCP uses two- or three-way handshakes to terminate a connection.

2. Y-C Lai, DCCP: Transport Protocol with Congestion Control and Unreliability, IEEE Internet Computing, September / October 2008

.

The End