University of Duisburg-Essen
Networked Embedded Systems Group
Institute for Computer Science and
Business Information Systems (ICB)
Schützenbahn 70
D-45127 Essen, Germany

Kommunikationsnetze 2
5 – Transport beyond UDP and TCP

Prof. Dr. Pedro José Marrón

Original slides from Prof. E.P. Rathgeb and M. Becke

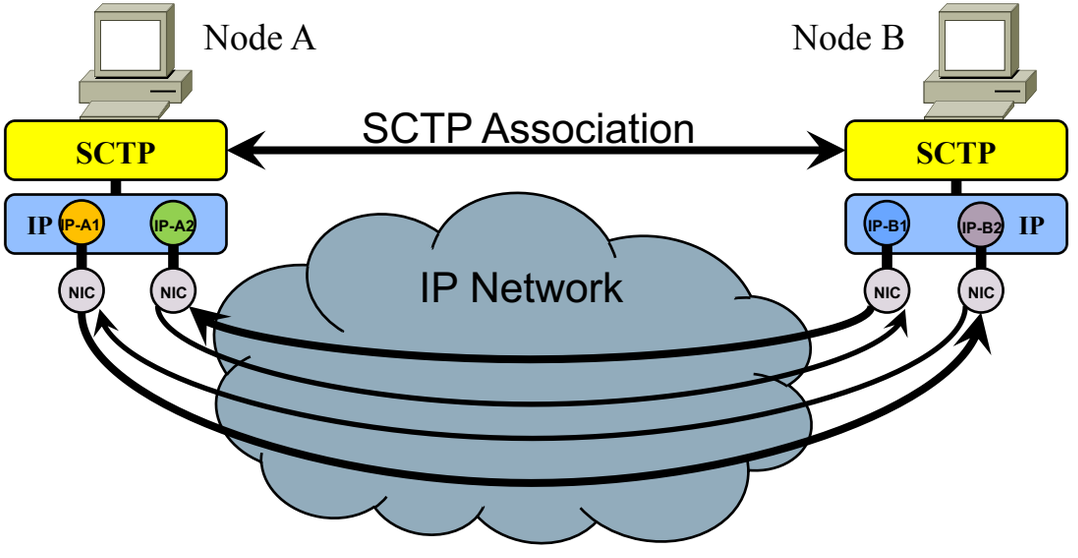
SCTP: Addressing TCP Limitations

- SCTP – Stream Control Transmission Protocol
  - Addresses TCP limitations with
    - No head-of-line blocking
    - No stream-oriented data transfer
    - Multihoming
    - (Limited) Protection against denial of service attacks
  - History
    - October 2000: Official SCTP standard [RFC 2960]
    - September 2002: Change of checksum algorithm [RFC 3309]
    - April 2007: Corrections and clarifications [RFC 4460]
    - September 2007: Updated standard [RFC 4960]

Properties of SCTP

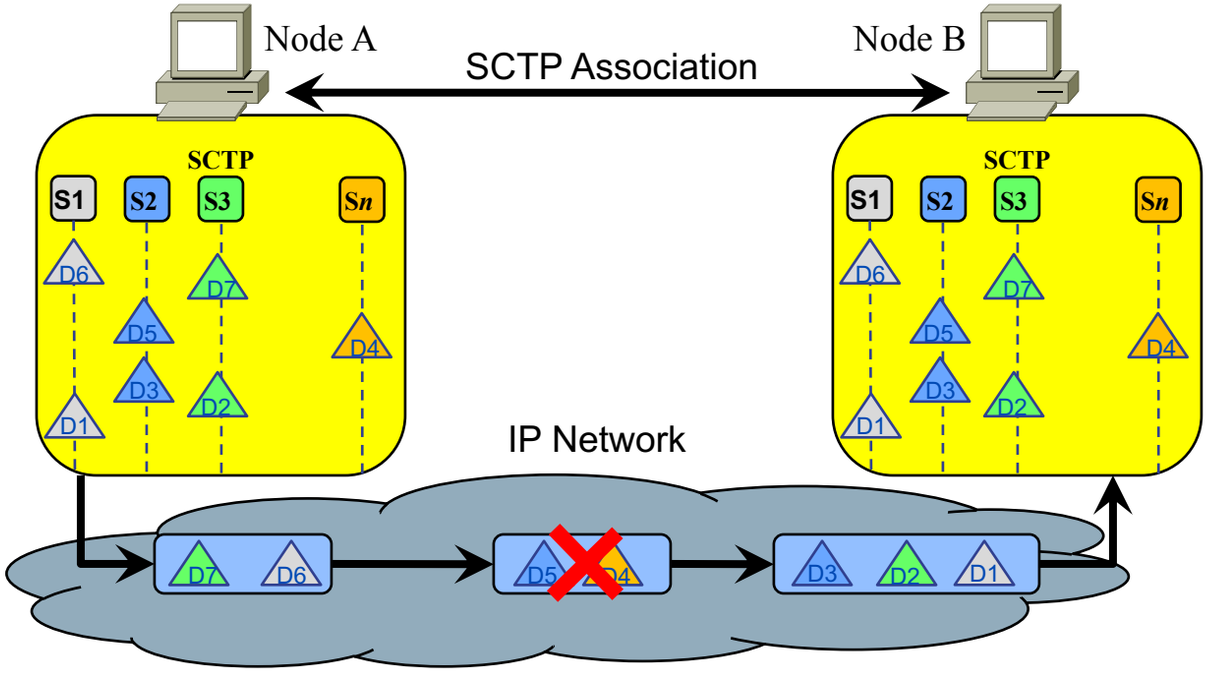
- Security
  - 4-way handshake
  - Verification tag
- Resilience and reliability
  - Multi-homing
  - CRC-32 checksum
- Practical features for application development
  - Message-oriented instead of stream-oriented
  - Multi-streaming
  - Support for IPv4 and IPv6 simultaneously (from multi-homing)
- (Some) Extensions
  - PR-SCTP: Per-message “unreliable transport” (limited or no-retransmissions)
  - “Add-IP”: Adding/removing IP addresses to/from association

Multi-Homing



- Association = connection between two nodes
- Path defines unidirectional relation to one address od the remote node
- Primary path = selected path used for data transmission
  - Other paths only used for error recovery
    - Heartbeat mechanism to monitor availability of other paths
  - Break of primary path ➔ switchover to another path

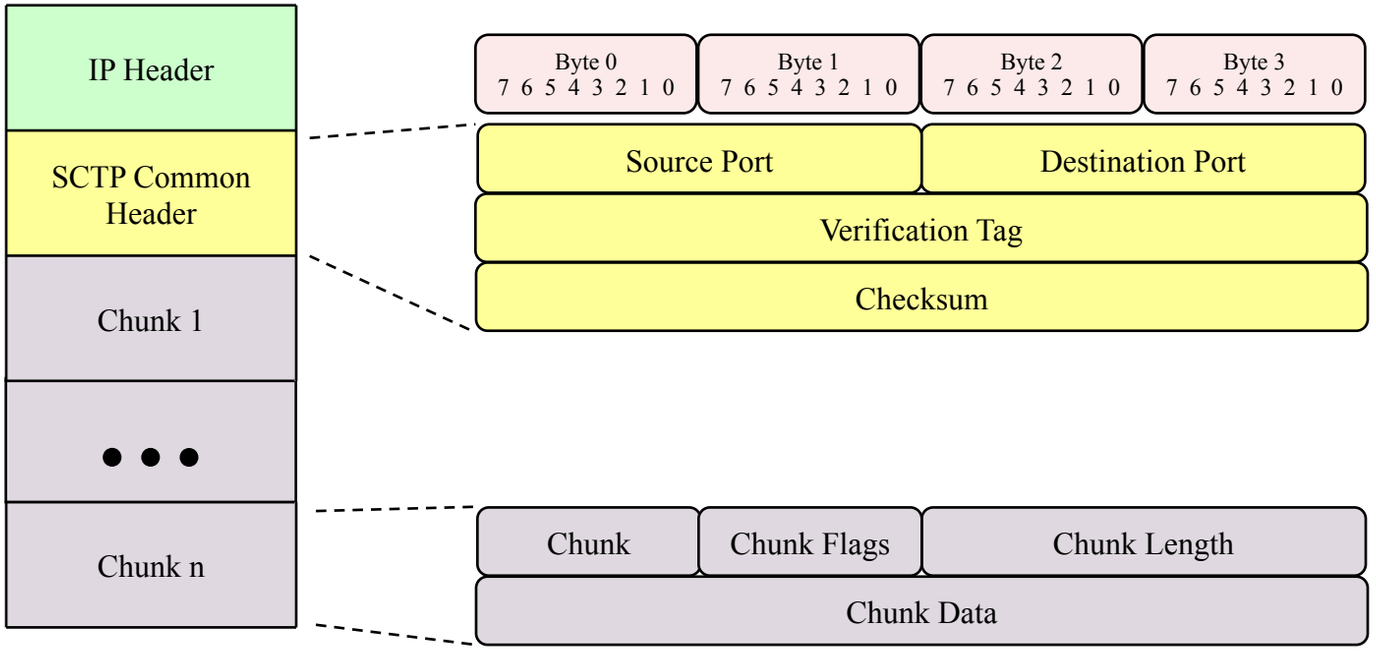
Multi-Streaming



- Stream = unidirectional flow over an SCTP association
- Loss of packet containing D5, D4
- Node B can already deliver D7, D6 to upper layer…
- … before D4 and D5 are retransmitted (i.e., no “head of line blocking”)

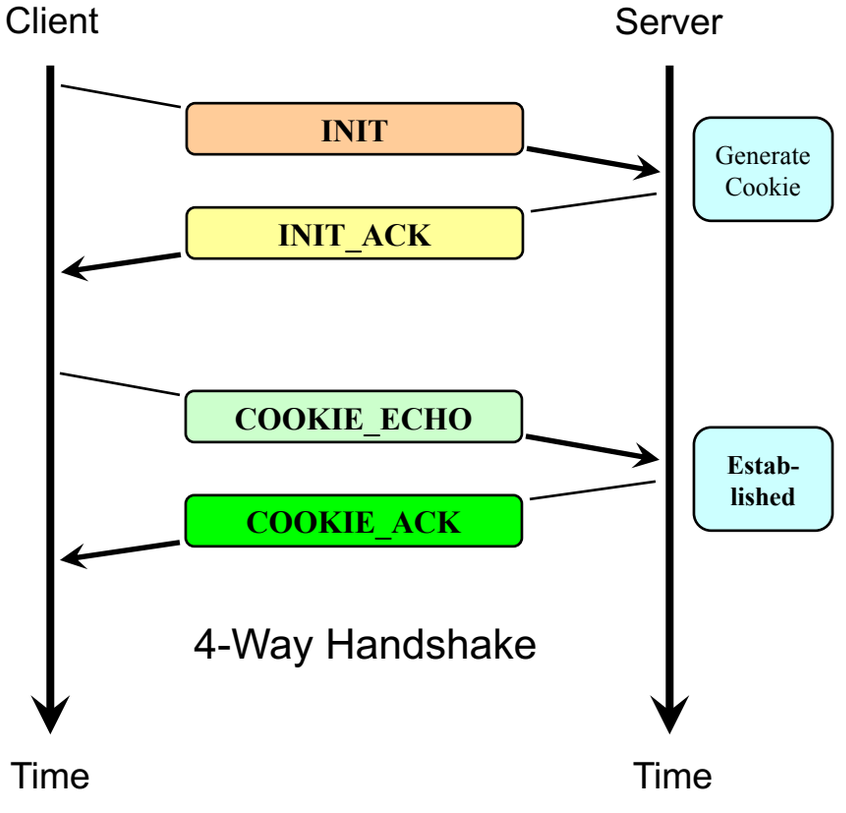
SCTP Packet Format

- SCTP Packet [RFC 4960]
  - Common header
  - Chunks for control information and user data transport



Association Setup (1): 4-Way Handshake

- INIT chunk
  - Request for association establishment
  - Server generates cookie with all association parameters
  - Cookie signed/encrypted with secret key
- INIT_ACK chunk
  - Contains cookie
  - Server releases all resources reserved for the association
- COOKIE_ECHO chunk
  - Contains the cookie
  - Signature with secret key – no modification by client possible
  - Server reads connection parameters, creates association
- COOKIE_ACK chunk
  - Confirmation of association establishment

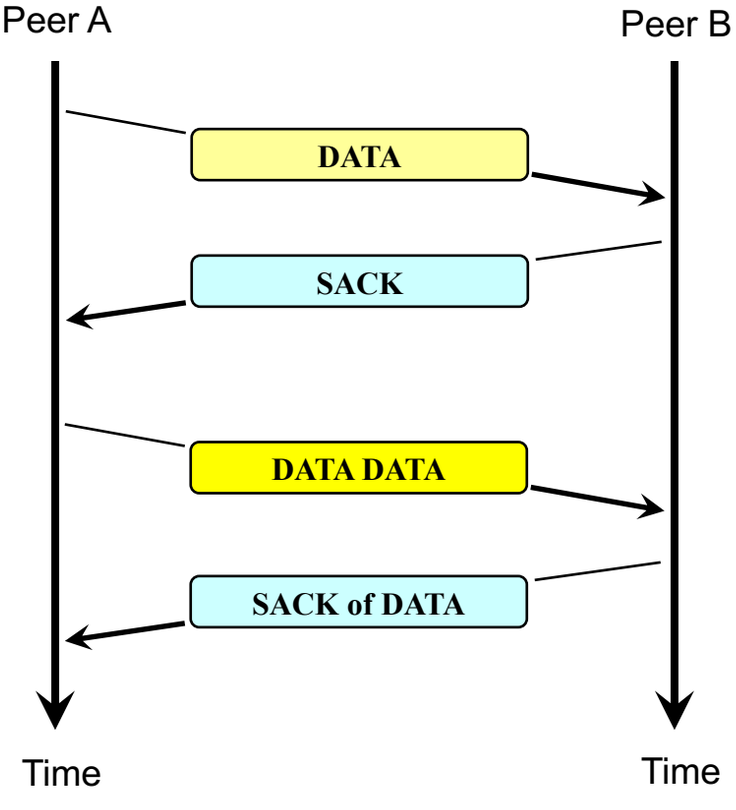


Association Setup (2): Parameter Negotiation

- Several parameters are negotiated during association establishment
- Paths
  - Telling remote endpoint all own addresses to be used
- Number of streams
  - In each direction
  - From 1 (usual case) to 65,535
- Verification tag
  - In each direction
  - 32-bit random number (with “good” generator)
  - Must be contained in all later packets
  - Reason
    - An attacker must guess IP address/port number and verification tag for successful blink packet injection
- Protocol extensions
  - Support (e.g., Add-IP, unreliable transport)
  - Usage (e.g., ECN – Explicit Congestion Notification)

Data Transmission

- Segmentation of messages
- DATA chunk
  - Message segment
  - Transport Sequence Number (TSN)
  - Flags
    - U = “unordered” (delivery may be out of sequence)
    - B = “begin of message”
    - E = “end of message”
  - Stream identifier
  - Stream sequence number (SSN)
  - Payload protocol identifier
- Bundling
  - Multiple DATA chunks per packet (e.g., small messages or from different streams)

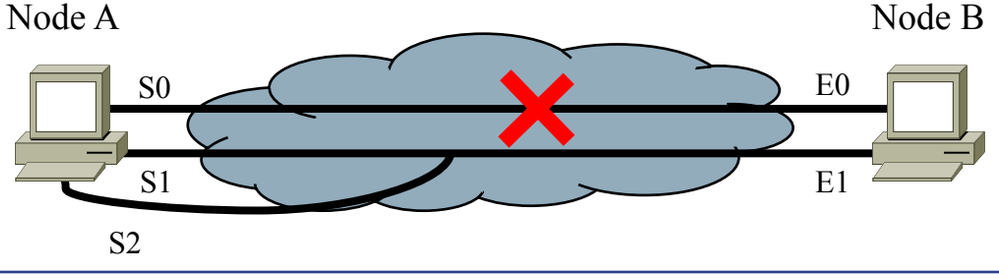


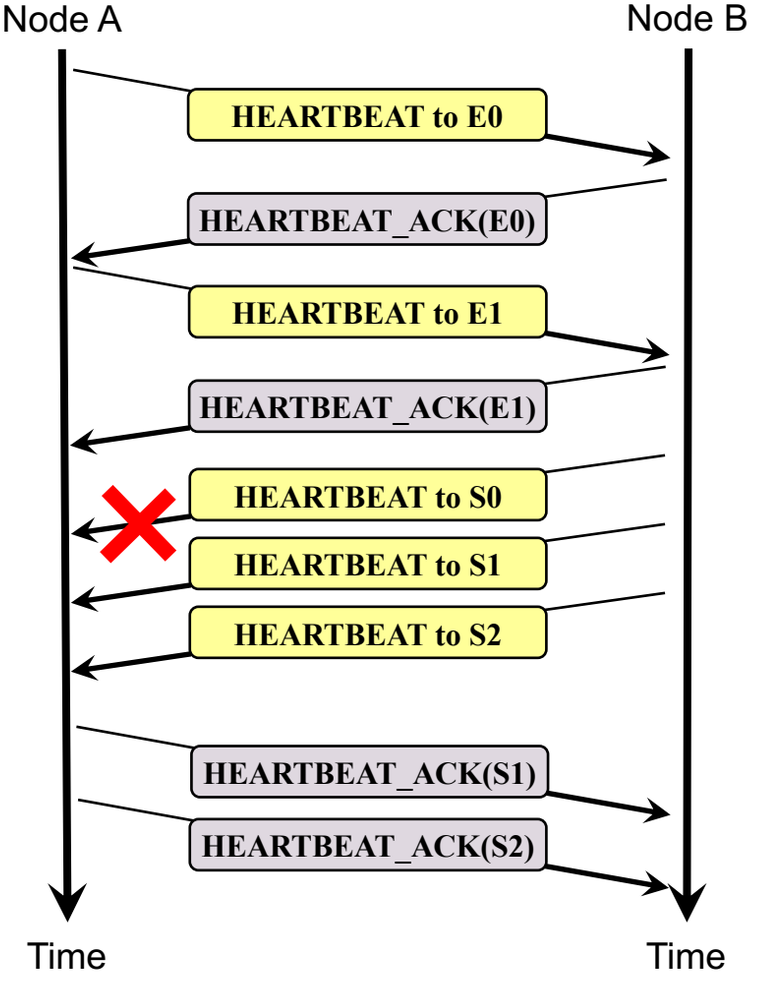
Selective Acknowledgement (SACK)

- SACK chunk
  - Contains acknowledgement for data chunks
- Important fields
  - Advertised receiver window (in bytes)
  - Cumulative TSN Ack
    - Up to this TSN, all chunks have been received
    - By definition: last TSN before a gap
  - Gap Ack blocks
    - Containing sequences (start TSN, end TSN) of chunk TSNs received following a break
  - Duplicate TSNs
    - Array of TSNs for which duplicates have been received
- Efficiency
  - Avoiding unnecessary retransmissions by gap reports
  - Detection of unnecessary retransmissions by duplicate TSNs

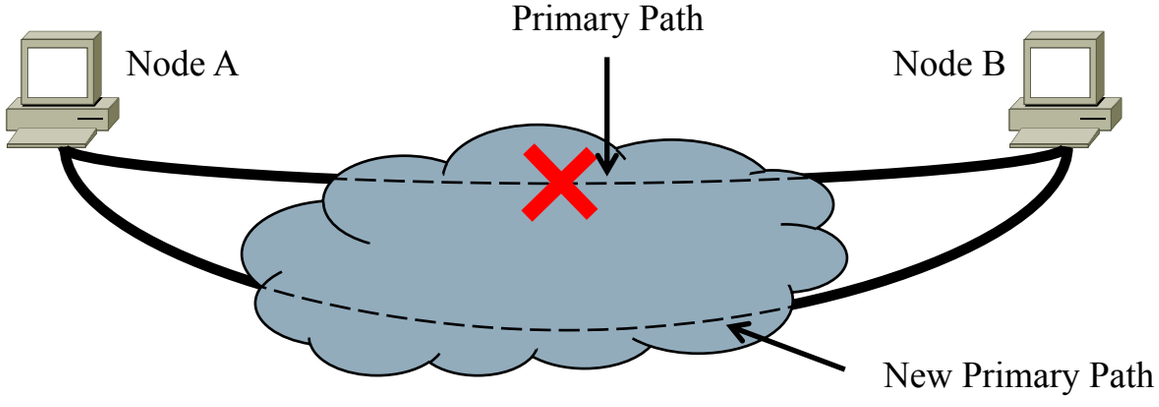
Path Monitoring using Heartbeats

- Path testing using Heartbeats
  - For each path (remote address)
    - Period HEARTBEAT messages
  - Answered by HEARTBEAT_ACK
  - Used to detect broken paths
  - No answer within timeout
    - Path is assumed to be broken
    - Broken paths will not be used as backup for primary



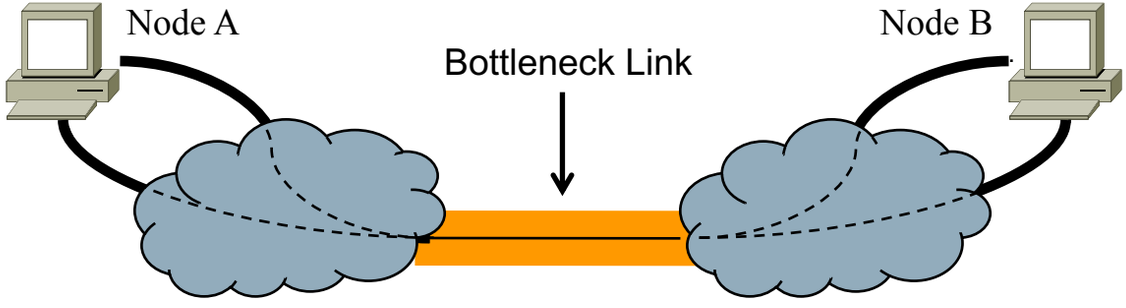


Changing the Primary Path



- Upon failure of the primary path
  - SCTP selects another path
- Upper layer is not involved
  - But can be notified about the path change (by option)
- Upper layer can explicitly request a change of the primary path
  - Locally
  - From the remove node for the reverse direction

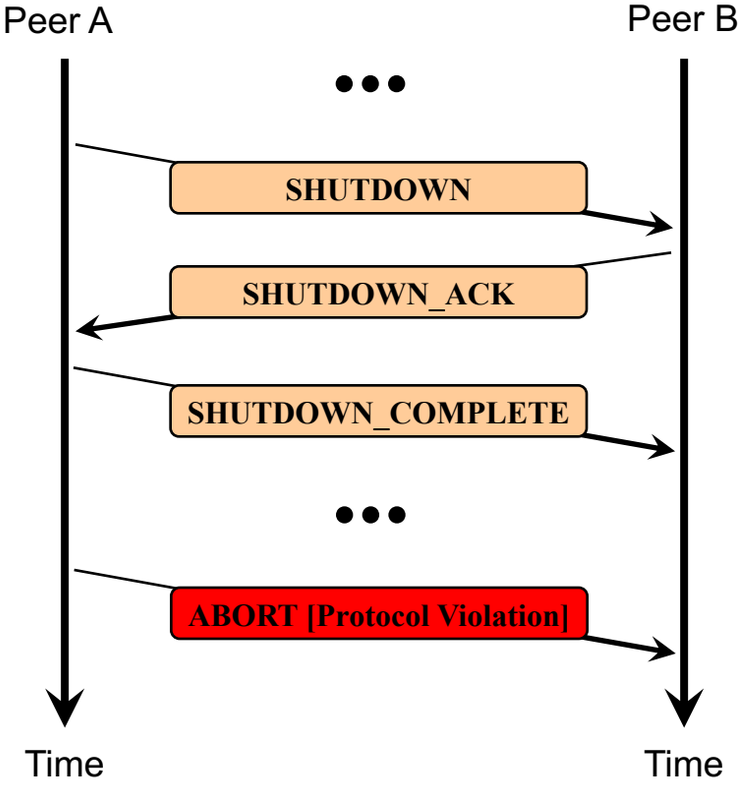
Why only one Primary Path?



- Example
  - Two paths between nodes A and B
  - Both paths share the same bottleneck link
- TCP fairness problem: Shared bottleneck
  - If both paths (or n paths) would be used for data transmission simultaneously
    - SCTP would get n times the bandwidth of a TCP connection if n paths use a shared bottleneck
- Network topology is unknown
  - There may be non-disjoint paths, i.e., shared bottlenecks
  - Reliable detection of shared bottlenecks not feasible in realistic networks

Association Teardown

- SHUTDOWN chunk
  - Initiates association shutdown
  - After all of Peer A's data has been acknowledged
  - Upon SHUTDOWN, Peer B stops accepting new data from upper layer (no half-open connections!)
- SHUTDOWN_ACK chunk
  - Acknowledges shutdown, after all data is acknowledged by Peer A
- SHUTDOWN_COMPLETE chunk
  - Finally acknowledges teardown of the association
- In case of any error
  - ABORT chunk for termination



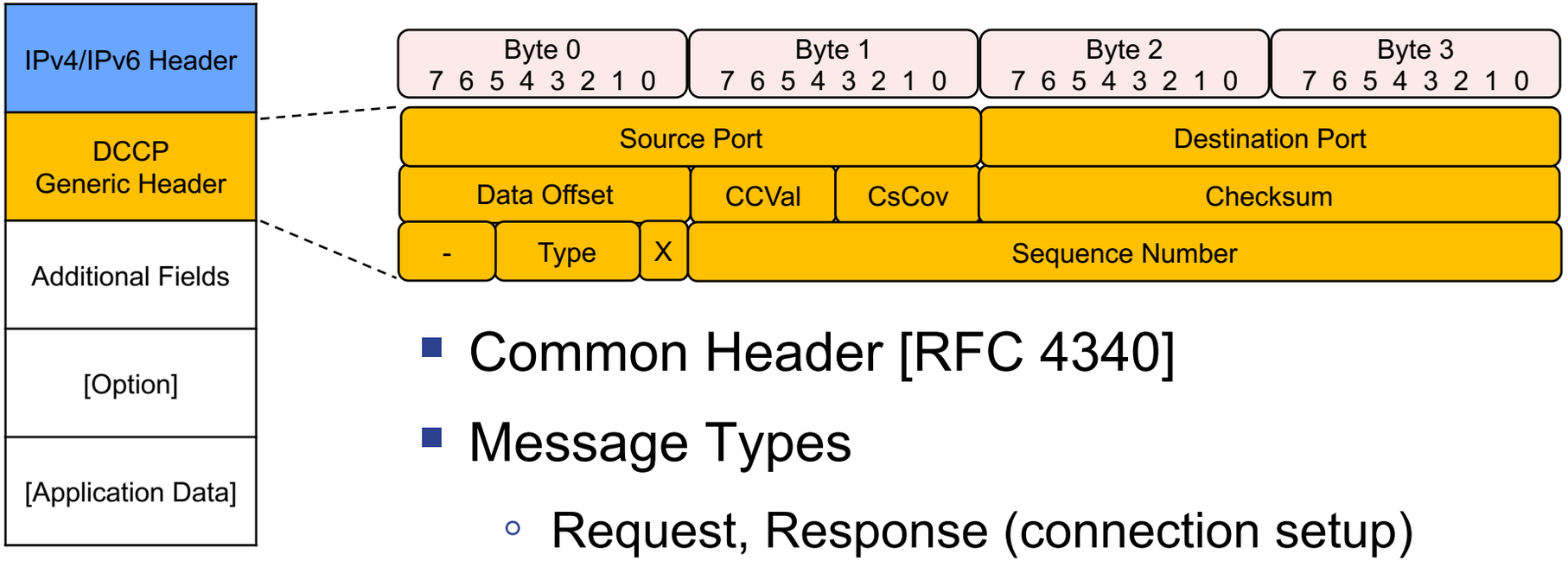
DCCP: Addressing UDP Limitations

- UDP has no congestion control
  - Sender may send as many UDP packets as possible
    - Limited only by its outgoing interface (e.g., Gigabit Ethernet)
  - On congested link
    - TCP-flows reduce bandwidth on congestion
    - UDP does not even notice congestion ➔ unfair to TCP flows!
- UDP does not ensure packet sequence
  - Packets may arrive out of sequence, e.g., due to
    - Route change
    - Load sharing
  - Receiver (upper layer) has to care for out-of-sequence packets

DCCP: Features

- Like UDP: message oriented, unreliable transport
- Like TCP: connection oriented, with congestion control
- Congestion control (per connection)
  - Congestion control mechanism negotiable
  - Mechanism identified by CCID (Congestion Control ID)
  - Current strategies defined
    - TCP-like flow control (window mechanism)
    - TCP-friendly rate control (try to provide given bandwidth)
- Sequence numbers
  - Numbering of packets (24-bit or optionally 48-bit)
  - Helps application to reorder packets
    - (but does not reorder!)
  - Detection of packet losses
    - (but does not retransmit!)

Packet Structure



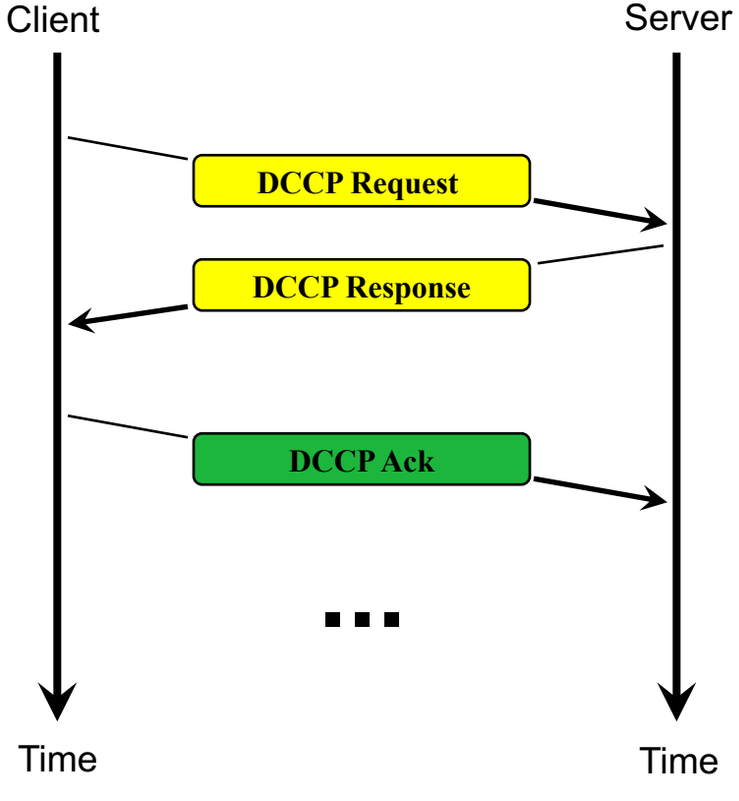
- Common Header [RFC 4340]
- Message Types
  - Request, Response (connection setup)
  - Data, Ack, DataAck (data transfer, acknowl.)
  - CloseReq, Close (connection shutdown)
  - Reset (connection termination)
  - Sync, SyncAck (resynch. seq. numbers)

Header Fields

- Source and Destination port
  - As for TCP and UDP
- Data offset
  - Start of data part in the message (in 32-bit units)
- Checksum
  - 16-bit Internet checksum
- CsCov – Checksum Coverage
  - Checksum can be turned off for part of data
  - Better efficiency for tolerant codec on error-prone channels
- CCVal – Congestion Control Value
  - Can be used for congestion control (CCID-specific)
- X – Extended Sequence Number
  - If set to 1, sequence numbers are 48 bits; 24 bits otherwise
  - Support for high-speed, long-delay transmission channels
  - Protection against blind Reset-attacks

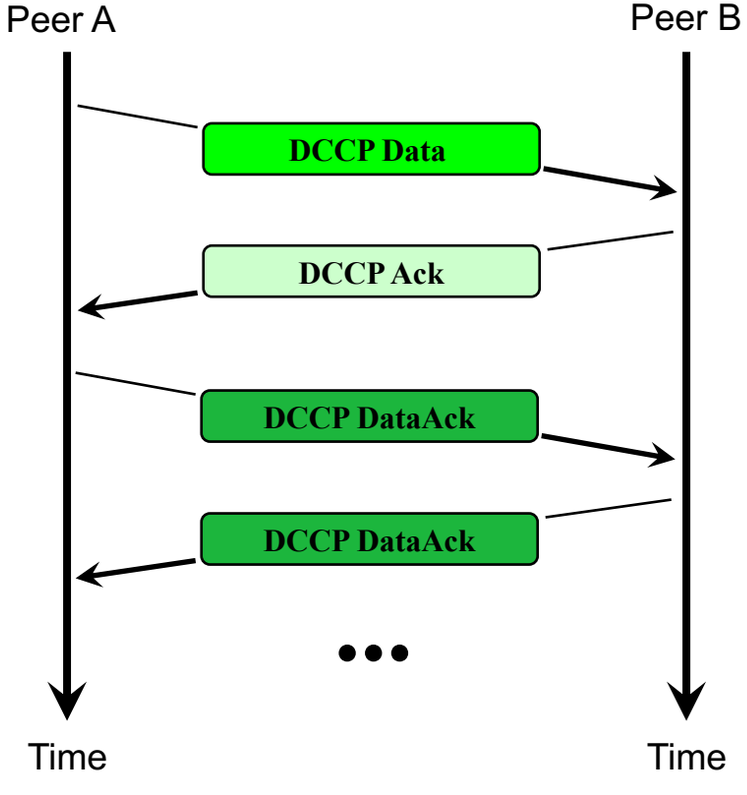
Connection Setup

- Request message
  - Initiates connection to server
- Response message
  - Confirms connection with client
- Init Cookie option for Response
  - Prevents “Request Flooding” attack
  - Signed connection parameters (server forgets them after sending!)
  - Must be returned by client in each message, until confirmed (3-way handshake)
  - Server retrieves parameters from cookie ➔ connection established



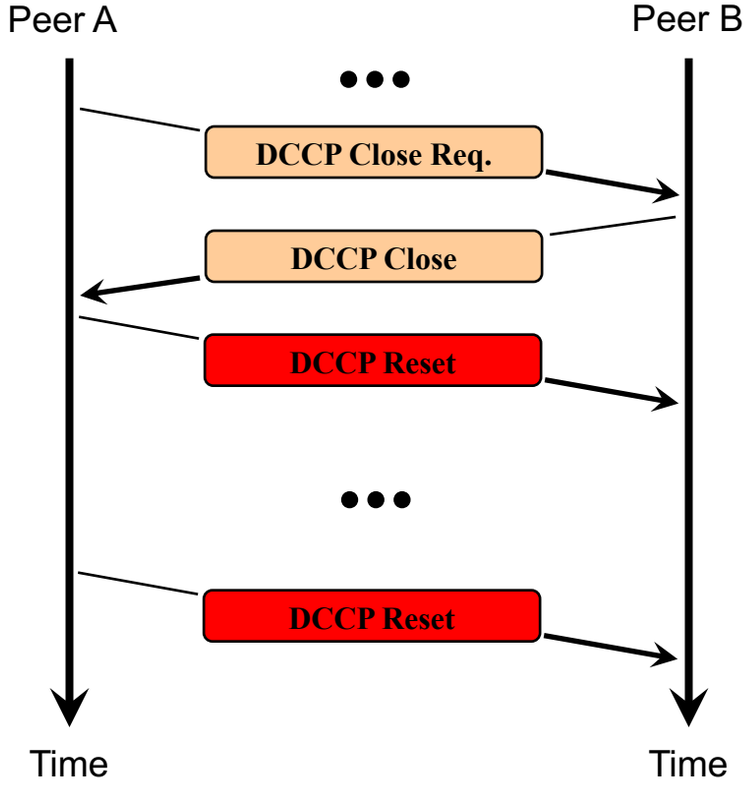
Data Transmission

- Data message
  - Transmission of user data
- Ack message
  - Acknowledgement for data
  - By sequence number
  - May acknowledge packet as “ECN marked”
- DataAck message
  - Piggybacking of acknowledgement with user data



Connection Teardown

- CloseReq message
  - Initiates connection teardown
  - Similar to TCP FIN flag
- Close message
  - Confirms connection teardown
- Reset message
  - Connection termination
  - Also used in case of errors
  - Similar to TCP RST flag



TCP-like Congestion Control

- CCID 2 [RFC 4341]
- Application cases
  - Achieve maximum bandwidth over the long term…
  - … when abrupt changes are no problem
- Provides congestion control similar to TCP
  - Slow start and slow start threshold
  - Congestion windows
    - Halved on congestion events
  - AIMD behaviour similar to TCP
    - Additive increase
    - Multiplicative decrease
  - Support for ECN

TCP-friendly Rate Control (TFRC)

- CCID 3 [RFC 4342]
- Application cases
  - Flows preferring to minimise the abrupt changes in the sending rate
  - Example: multimedia streams (with small buffering before playback)
- TCP-friendly Rate Control (TFRC)
  - Sender maintains transmit rate
  - Updated using receiver's estimate of packet loss
  - Response to congestion more smoothly
    - Different from TCP-like behaviour in short term…
    - … but operates fairly over the long term
  - Support for ECN
  - Assumptions
    - Fixed packet sizes
    - Application varies packet rate – not size – in response to congestion

Transport Layer Protocol Comparison

Feature	UDP	DCCP	TCP	SCTP
Connection-Oriented	no	yes	yes	yes
Message-Oriented	yes	yes	no	yes
Reliable Transport	no	no	yes	configurable
Unreliable Transport	yes	yes	no	configurable
Ordered Delivery	no	no	yes	configurable
Flow Control	no	no	yes	yes
Congestion Control	no	configurable	yes	yes
Multi-Homing	(Application)	no	no	yes
Multi-Streaming	(Application)	(Application)	no	yes

Literature (1)

- “TCP/IP Illustrated, Volume 1 The Protocols” by W. Richard Stevens
- “Internet Working with TCP/IP Volume 1” by Douglas E. Comer
- “Sams Teach Yourself TCP/IP in 24 Hours” by Joe Casad. Published by Sams
- “Network Congestion Control” by Michael Welzl.
- S. Floyd, J. Padhye, J. Widmer “Equation Based Congestion Control for Unicast Applications”, *Sigcomm 2000.*
- J. Padhye, V.Firoiu, D. Towsley, J. Kurose, “Modeling TCP Throughput: a Simple Model and its Empirical Validation”, *Sigcomm 1998.*
- M. Handley, S. Floyd, J. Padhye, J. Widmer, “TCP Friendly Rate Control (TFRC): Protocol Specification”, RFC 3448, January 2003

Literature (2)

- "Stream Control Transmission Protocol (SCTP)" by R. Stewart and Q. Xie, Addison-Wesley Longman, Amsterdam, 2002

- RFCs
  - [RFC 768]: "User Datagram Protocol (UDP)"
  - [RFC 793]: "Transmission Control Protocol (TCP)"
  - [RFC 4340]: "Datagram Congestion Control Protocol (DCCP)"
  - [RFC 4340]: "Datagram Congestion Control Protocol (DCCP)"
  - [RFC 4960]: "Stream Control Transmission Protocol" (New Version)