Master Course Computer Networks IN2097

Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.

Chair for Network Architectures and Services
Institut für Informatik
Technische Universität München
http://www.net.in.tum.de

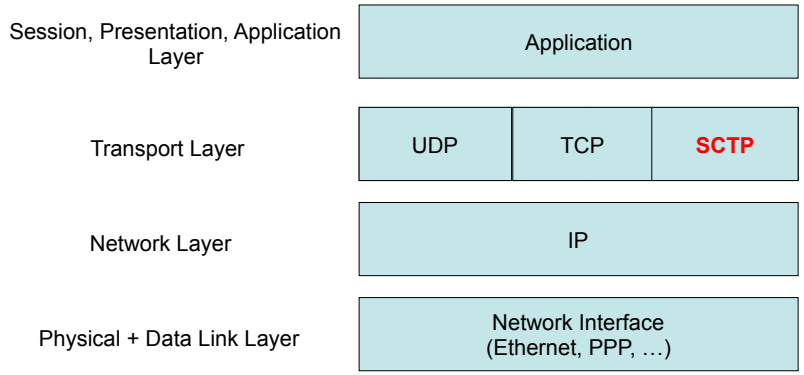
Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Stream Control Transmission Protocol (SCTP)

Technische Universität München

Internet Protocol Stack

- The Internet Protocol Stack



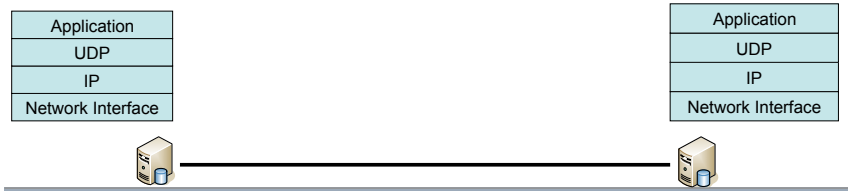
- Why another transport layer protocol?

Contents

- Limitations of UDP and TCP
- The Stream Control Transmission Protocol (SCTP)
  - Association setup / stream setup
  - Message types
  - Partial Reliability
  - Multi-Homing support
  - Congestion control

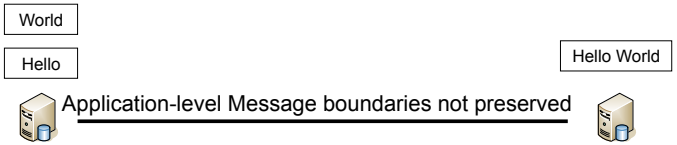
User Datagram Protocol

- Message oriented
  - Sending application writes a N byte message
  - Receiving application reads a N byte message
- Unreliable
  - Lost packets will not be retransmitted
- Unordered delivery
  - Packets may be re-ordered in the network



Transmission Control Protocol

- Connection/Stream oriented (Not message oriented)



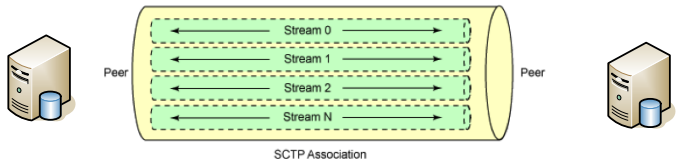
- Reliable transmission
  - Lost packets are retransmitted
  - Retransmission will be repeated until acknowledgment is received
- In-order delivery
  - Segments n + 1, n + 2, n + 3, will be delivered after segment n
- Congestion control
  - TCP tries to share bandwidth equally between all end-points

Problems

- Certain applications have problems with UDP and TCP
- TCP: Head-of-line blocking with video streaming
  - Frames 2,3,4 arrived but cannot be shown because frame 1 is missing
  - ⇨ Video will stop until frame 1 is delivered
- UDP:
  - Out-of-order delivery possible
  - Lost packets neither detected nor corrected
  - No congestion control
- Example: Internet-Telephony
  - Two types of traffic:
    - Signalling traffic: should be delivered reliable + in-order (TCP)
    - Voice traffic: should not suffer from head-of-line blocking (UDP)
  - Need to manage two sockets
- SCTP can deal with these problems

SCTP Features at a glance

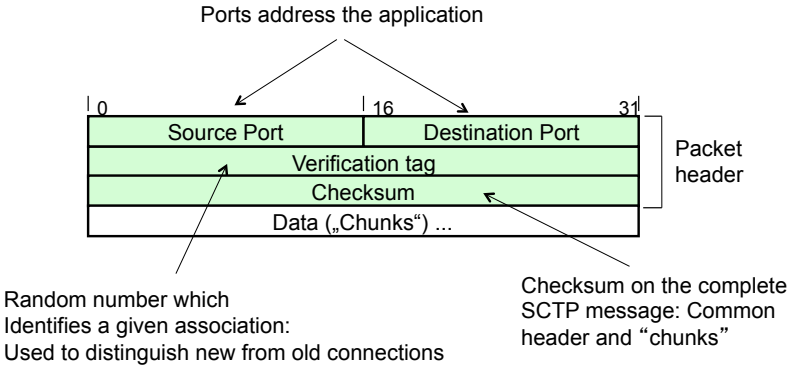
- **Connection and message oriented**
  - SCTP builds an “association” between two peers
  - Association can contain multiple “streams”
  - Messages are sent over one of the streams



- **Partial reliability**
  - “Lifetime” defined for each message
    - Retransmission of a message is performed during its lifetime
  - Messages delivery can be unreliable, fully reliable or partially reliable
- **Multi-Homing**
  - SCTP can use multiple IP addresses

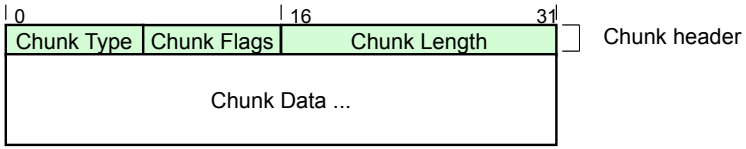
SCTP Message Format

- **Common header format**
  - 12 byte header
  - included in every SCTP message



SCTP Chunk Format

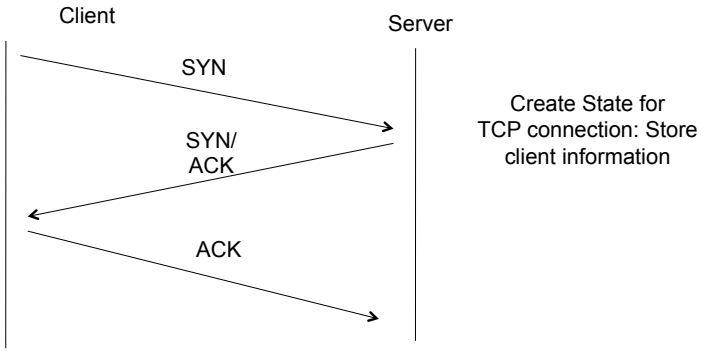
- Data and signaling information is transported in chunks
  - One or more chunks in a SCTP message
  - Each chunk type has a special meaning:
    - INIT, INIT-ACK, COOKIE, COOKIE-ACK ⇨ Connection setup
    - DATA ⇨ Transports user data
    - SACK ⇨ Acknowledge Data
- Common chunk format



- Additional formats are defined for specific chunk types

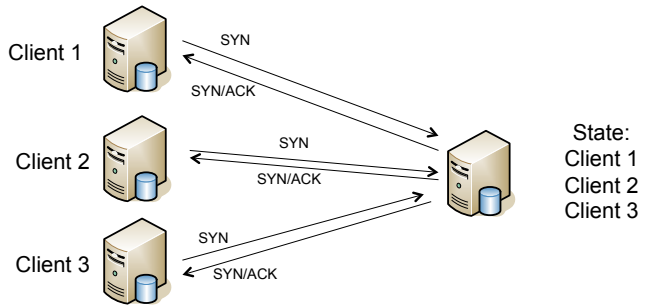
Connection Setup

- **TCP connection setup**



- **Known Problem: TCP SYN-Flooding**

SYN Flodding



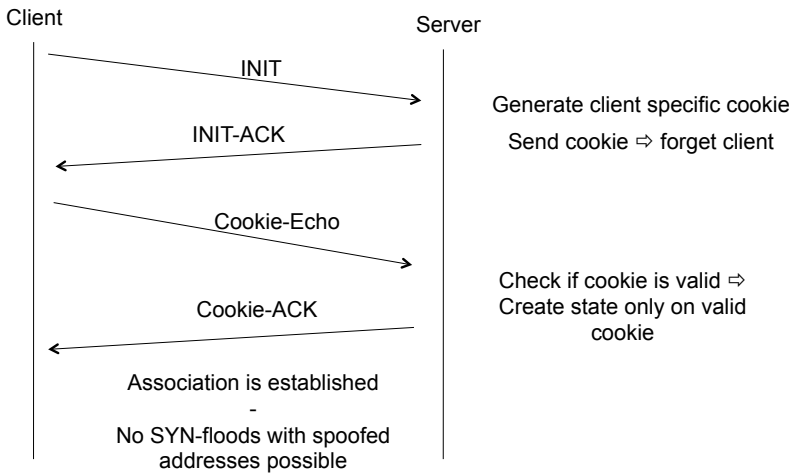
- Clients send SYN-Packets but do not respond to SYN-ACK
  - Usually done by a single client that performs IP address spoofing
  - Works because only a single forged packet is necessary

⇨ Server has to store state until a TCP timeout occurs

- May lead to resource exhaustion, during which server cannot accept new connections

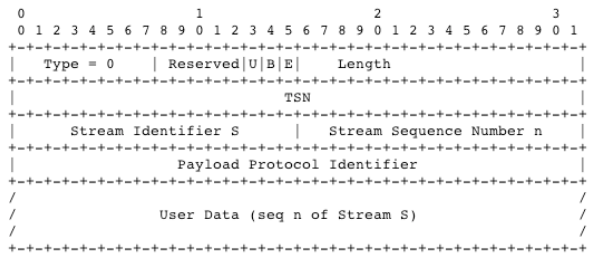
SCTP Association Setup

- **Solution to SYN-Flood problem: Cookies**



Data Transmission

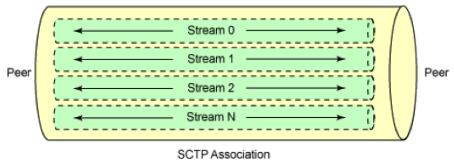
- Application data is transmitted in Data Chunks
  - A data chunk is associated to a stream (Stream Identifier S)



- TSN (Transport Sequence Number)
  - Global Sequence Number
  - Similar to TCP sequence number, used for retransmissions
- Stream sequence number
  - Necessary for per-stream transmission reliability

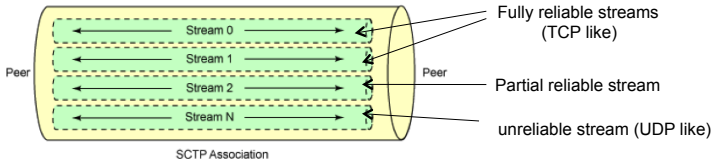
Transmission reliability (1)

- TCP
  - Segments are transmitted fully reliably
  - Segments are delivered in-order to the application
  - Slow start and congestion avoidance for congestion control
- UDP
  - Packets are transmitted fully unreliable ⇨ never retransmitted
  - No re-ordering ⇨ packet order may be changed at the receiver
  - No congestion control
- SCTP can do both and more, in a stream-specific way



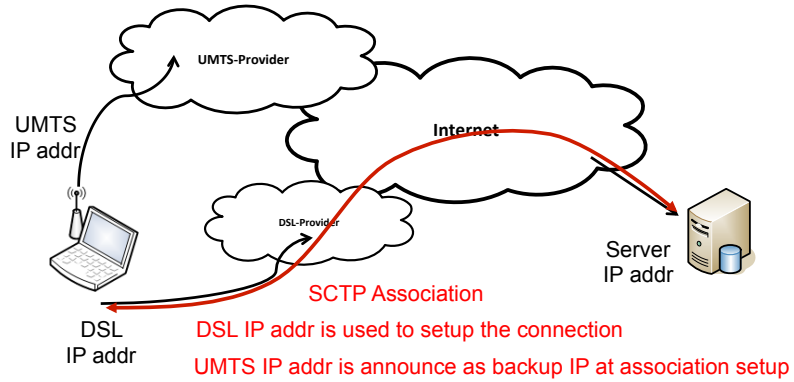
Transmission reliability (2)

- Why multiple streams?
  - Solves head of line blocking
  - Simpler firewall rules (only one port for several streams)
  - Partial Reliability Extension (PR-SCTP) for different reliability levels
- PR-SCTP
  - Allows to set a lifetime parameter for each stream
  - Lifetime specifies how long the sender should try to retransmit a packet
  - Allows to mix reliable and unreliable streams



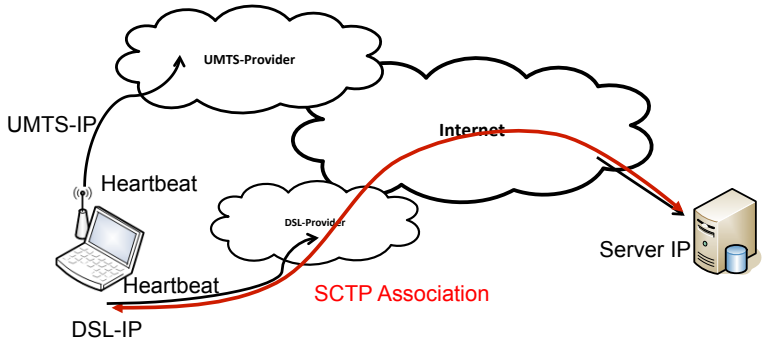
Multi-Homing: Association setup

- SCTP chooses one IP address at association setup
  - IP address can be specified by user



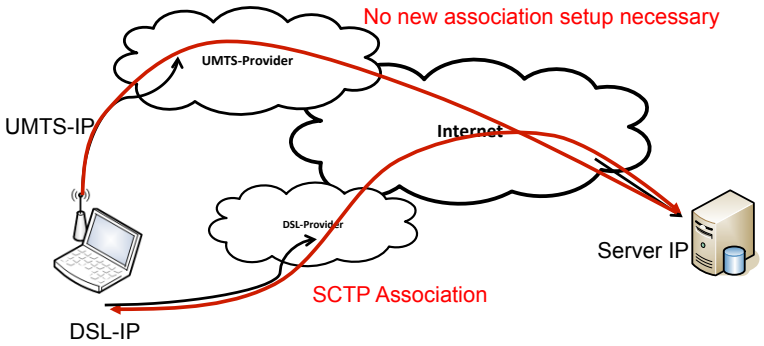
Multi-Homing

- Heartbeat messages are periodically sent to check link availability



Multi-Homing

- Changes occur when the default link is found to be broken
  - Is identified because of packet loss (data or heartbeat)
  - Consequence: SCTP will resume on the backup link



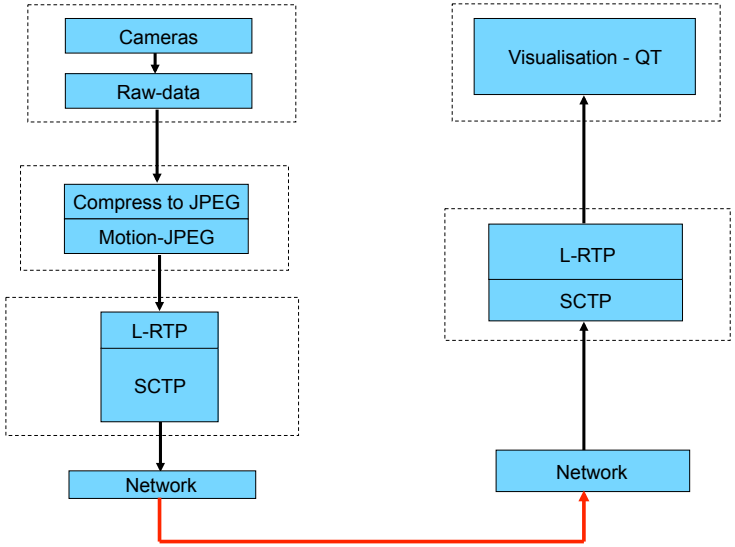
SCTP Example Scenario

- Real-time transmission of video streams and control data in vehicular scenario

Server: vehicle with embedded PC (Linux)

Client: Unix/Windows

Protocol Architecture



SCTP Deployment

- SCTP has attractive features
  - but to which extent is it used?
- Why do we use HTTP over TCP for Video Streaming?
- Firewall and NAT issues
  - Most home routers simply can‘t translate SCTP
- Implementations
  - not yet supported by all operating systems / hosts
- BUT: mandatory for some newly developed protocols such as IPFIX (IP Flow Information Export)

SCTP Standardisation

RFC 6458 Sockets API Extensions for the Stream Control Transmission Protocol (SCTP)
RFC 6096 Stream Control Transmission Protocol (SCTP) Chunk Flags Registration (updates RFC 4960)
RFC 5062 Security Attacks Found Against the Stream Control Transmission Protocol (SCTP) and Current Countermeasures
RFC 5061 Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration
RFC 5043 Stream Control Transmission Protocol (SCTP) Direct Data Placement (DDP) Adaptation
RFC 4960 Stream Control Transmission Protocol
RFC 4895 Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)
RFC 4820 Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)
RFC 4460 Stream Control Transmission Protocol (SCTP) Specification Errata and Issues
RFC 3873 Stream Control Transmission Protocol (SCTP) Management Information Base (MIB)
RFC 3758 Stream Control Transmission Protocol (SCTP) Partial Reliability Extension
RFC 3554 On the Use of Stream Control Transmission Protocol (SCTP) with IPsec
RFC 3436 Transport Layer Security over Stream Control Transmission Protocol
RFC 3309 Stream Control Transmission Protocol (SCTP) Checksum Change (obsoleted by RFC 4960)
RFC 3286 An Introduction to the Stream Control Transmission Protocol
RFC 3257 Stream Control Transmission Protocol Applicability Statement
RFC 2960 Stream Control Transmission Protocol (updated by RFC 3309 and obsoleted by RFC 4960)

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

Reliable Multicast Transport

Technische Universität München

Many Uses of Multicasting

- Teleconferencing
- Distributed Games
- Software/File Distribution
- Video Distribution
- Replicated Database Updates

⇨ multicast transport is done differently for each application

Multicast Application Modes

- Point-to-Multipoint:
  Single Source, Multiple Receivers
- Multipoint-to-Multipoint:
  Multiple Sources, Multiple Receivers
- Sources are receivers
- Sources are not receivers

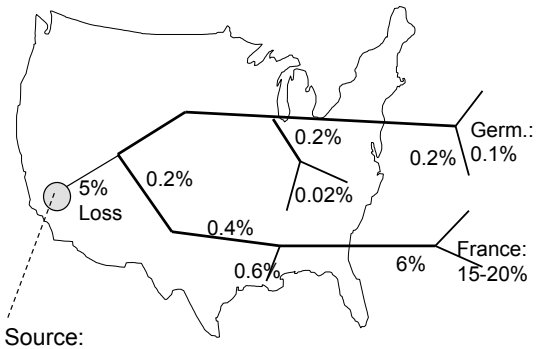
Classification of Multicast Applications

Transport service type	*Fully reliable multicast*	*Real-time multicast*
Single source: ***1:N***	Multicast-FTP; Software update	Audio-visual conference; Continuous Media Dissemination
Multiple Sources ***M:N***	CSCW; Distributed computing	DIS; VR

- CSCW: Computer Supported Cooperative Work
- DIS: Distributed Interactive Simulation
- VR: Virtual Reality

Where Does Multicast Loss Occur

- Example measurements
  (April 96, Yajnik, Kurose, Towsely, Univ. Mass., Amherst)



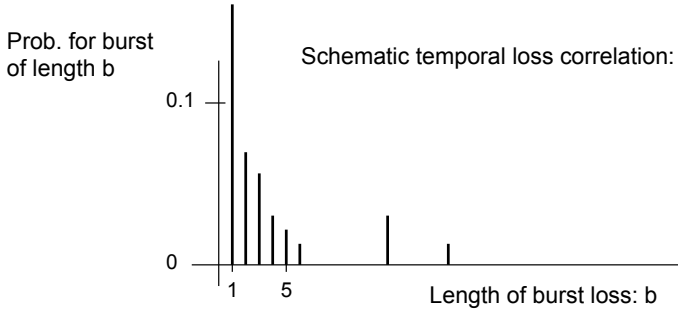
Simultaneous Packet Loss

- Q: distribution of number of receivers losing packet?
- Example dataset:
  47% packets lost somewhere
  5% shared loss
- Similar results across different datasets
- Models of packet loss (for protocol design, simulation, analysis):
  - star: end-end loss independently
  - full topology: measured per link loss independently
  - modified star: source-to-backbone plus star
    ⇨ good fit for example data set

Temporal Loss Correlation

Q: do losses occur individually or in “bursts”?

- occasional long periods of 100% loss
- generally isolated losses
- occasional longer bursts



Reliable Multicast Challenge

- How to transfer data reliably from source to R receivers
- scalability: 10s - 100s - 1000s - 10000s - 100000s of receivers
- heterogeneity
  - different capabilities of receivers (processing power, buffer, protocol capabilities)
  - different network conditions for receivers (bottleneck bandwidths, loss rates, delay)
- feedback implosion problem

ARQ: Alternatives for Basic Mechanisms

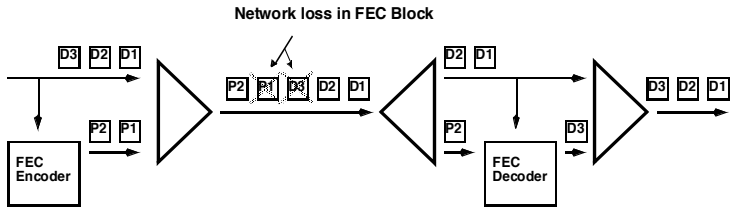
- Who retransmits
  - source
  - network / servers
  - other group member.
- Who detects loss
  - sender based: waiting for all ACKs
  - receiver based:
    NACK, more receivers ⇨ faster loss detection.
- How to retransmit
  - Unicast
  - Multicast
  - Subgroup-multicast

Approaches

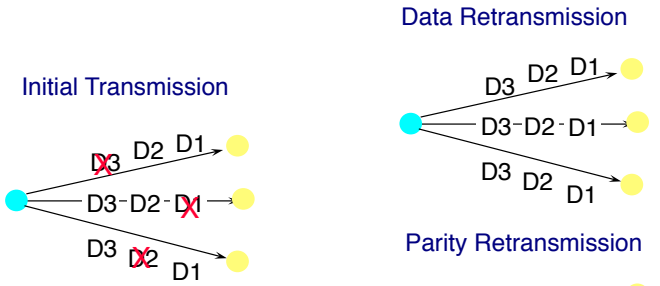
- shift responsibilities to receivers (in contrast to TCP: sender is responsible for large share of functionality)
- feedback suppression (some feedback is usually required)
- multiple multicast groups (e.g. for heterogeneity problems; can be used statically or dynamically)
- local recovery (can be used to reduce resource cost and latency)
- server-based recovery
- forward error correction (FEC)
  - FEC for unicast: frequently no particular gain
  - FEC for multicast: gain may be tremendous!

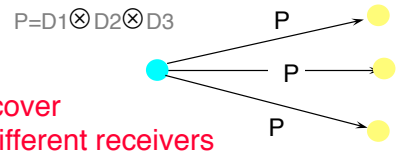
Forward Error Correction (FEC)

- k original data packets form a **Transmission Group (TG)**
- h parity packets derived from the k data packets
- any k received out of k+h are sufficient
- Assessment
  - \+ allows to recover lost packets
  - \- overhead at end-hosts
  - \- increased network load may increase loss probability



Potential Benefits of FEC





One parity packet can recover different data packets at different receivers