Advanced Networking ’2006

Jürgen Schönwälder

j.schoenwaelder@iu-bremen.de

International University Bremen

Campus Ring 1

28725 Bremen, Germany

http://www.faculty.iu-bremen.de/jschoenwae/an-2006/

0. Preface

Course Content

- Internet Multicasting (IGMP, DVMRP, MOSPF, PIM)
- New Internet Transport Protocols (SCTP, DCCP)
- Internet Quality of Service (IntServ, RSVP, DiffServ)
- Multimedia Transport and Signaling (RTP, RTCP, SIP)
- Voice over IP (codecs, metrics)
- Mobility (MIPv4, MIPv6, Link-Layer Handovers)
- Highspeed TCP Extensions (ECN, TCP improvements)
- Domain Name System Extensions (SRV, ENUM, DDDS)
- Label Switching (MPLS, LDP, RSVP-TE, GMPLS)
- Measurement, Modeling and Simulation (NETFLOW, NS-2)

? P2P Services (Skype, P2P SIP, Dundee, ...)

? Wireless Sensor Networks

Course Objective

- Introduce advanced networking concepts
- Combine theory with practical experiences (lab sessions)
- Prepare students for research in the networking field
  - Learn to read and review papers
  - Learn to present and evaluate ideas

Background Reading Material

- A.S. Tanenbaum, "Computer Networks", 4th Edition, Prentice Hall, 2002
- W. Stallings, "Data and Computer Communications", 6th Edition, Prentice Hall, 2000
- C. Huitema, "Routing in the Internet", 2nd Edition, Prentice Hall, 1999
- D. Comer, "Internetworking with TCP/IP Volume 1: Principles Protocols, and Architecture", 4th Edition, Prentice Hall, 2000
- J.F. Kurose, K.W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 3rd Edition, Addison-Wesley 2004.

Important Publications

- ACM/IEEE Transactions on Networking
- ACM Computer Communications Review (SIGCOMM)
- ACM Mobile Communications Review (MOBICOM)
- IEEE Journal on Selected Areas of Communications
- IEEE Transactions on Communications
- IEEE Transactions on Wireless Communications
- IEEE Transactions on Mobile Computing
- IEEE eTransactions on Network and Service Management
- IEEE Communications
- IEEE Surveys and Tutorials
- Elsevier Computer Networks
- Elsevier Computer Communications
- Elsevier Ad Hoc Networks
- . . .

Conferences and Workshops

- ACM SIGCOMM
- IEEE INFOCOM
- IEEE GLOBECOM
- IEEE/IFIP IM / NOMS / DSOM / MNMS / IPOM / ...
- IEEE ICC
- . . .

- It is essential to know where to submit a paper; not all events have the same quality.
- Kevin C. Almeroth maintains a nice useful web page with conference statistics.

Networking Challenges

- Some network research areas:
  - Routing and Convergence
  - Security, Trust and Key Management
  - Network Management and Operations
  - Ad-hoc networks and self-organizing networks
  - Scalable Inter-Domain Quality of Service (QoS)
  - Measurement and Modeling
  - Killer Applications
  - Disappearing (invisible) Networks
  - Interplanetary Internet

$\Longrightarrow$ See RFC 3869 for further information on Internet Research.

Prerequisites

- Protocol Layering (ISO/OSI, Internet)
- Names, Addresses, Services, Protocols
- Transmission Media and their Properties
- Data Encoding (NRZ, RZ, Manchester)
- Media Access (TDM, FDM, CSMA-CD, MACA)
- Error Detection (Checksums, CRC)
- Sequence Numbers, Acknowledgements, Timer
- Flow Control and Congestion
- Local Area Networks (IEEE 802.3, 802.11, 802.1D)

Prerequisites (cont.)

- Internet Network Protocols (IPv4, IPv6)
- Internet Forwarding and Routing (RIP, OSPF, BGP)
- Internet Transports (UDP, TCP)
- Internet Application Protocols
  - Notations (ASN.1, ABNF)
  - Domain Name System (DNS)
  - Email (SMTP, IMAP)
  - Document Transfer (HTTP, FTP)
  - Network Management/Monitoring (SNMP)
- Remote Procedure Calls (Stubs, Semantics)
- Socket Programming in C

Grading Scheme

- Homeworks (20%)
  - Individual submission of solutions
  - Control your continued learning success
- Projects (20%)
  - May include practical programming exercises
  - May include the study of some research papers
- Midterm examination (30%)
  - Covers the first part of the lecture
- Final examination (30%)
  - Covers the whole lecture

Reasons for not taking this course

- You do not have the time required for this course
- You do not have the required background
- You expected an introductionary course
- You find the topics covered by this course boring
- You are unable to do some programming in C/Unix
- You are not ready to take initiative
  - Reading research papers and specifications
  - Programming tasks
  - Software setup and installation

References

[1] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, 4 edition, 2002.

[2] W. Stallings. *Data and Computer Communications*. Prentice Hall, 7 edition, 2004.

[3] C. Huitema. *Routing in the Internet*. Prentice Hall, 2 edition, 1999.

[4] D. E. Comer. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*. Prentice Hall, 4 edition, 2000.

[5] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, 3 edition, 2004.

[6] R. Atkinson and S. Floyd. IAB Concerns and Recommendations Regarding Internet Research and Evolution. RFC 3869, Internet Architecture Board, August 2004.

1. Internet Multicasting

Terminology

- *Unicast*: Communication between a single sender and a single receiver (1:1).
- *Multicast*: Communication between a single sender and multiple receivers (1:n).
- *Concast*: Communication between multiple senders and a single receiver (m:1).
- *Multipeer*: Communication between multiple senders and multiple receivers (m:n).
- *Anycast*: Communication between a single sender and one selected receiver out of a group of receivers.
- *Broadcast*: Communication between a single sender and all receivers attached to a network segment.

Multicast Groups

- *Open* vs. *closed* multicast groups:
  Member of open groups accept messages from arbitrary senders while member of closed groups only accept messages from the group.
- *Dynamic* vs. *static*:
  Membership of static groups does not change during communication while membership of dynamic groups may change over time.
- *Transient* vs. *permanent*:
  Permanent groups exist permanently, even if the group has no members while transient groups only exist for a limited period of time.

Reliable Multicasts

- *Unreliable*:
  No guarantee that data is delivered to multicast group members.
- *Reliable*:
  Guarantee that data is delivered to all multicast group members.
- k-*Reliable*:
  Data is reliably delivered to at least k group members.
- p-*Reliable*:
  Data is reliably delivered to a certain percentage p of the group members.

Multicast Challenges

- Multicast flow and congestion control
  - Positive (ACK) vs. negative (NACK) acknowledgements
  - Maintenance of the sender's buffer
  - Rate-based instead of window-based flow control
  - Feedback implosion problems
- Multicast reliability
- Multicast routing
- Multicast security
- . . .

Group Addresses

- Group members can be identified by explicit member lists or group addresses.
- Group addresses may be assigned from a central authority or dynamically in a decentralized fashion.
- Group address assignments may be permanent or transient.
- Decentralized approaches usually require the introduction of mulicast group address management server for coordination.
- Transient group addresses may be announced in a shared directory.

Internet Multicast Addresses

- IPv4 (RFC 1112):

```
| 4  |          28 bits             |
+-----------------------------------+
|1110|         group ID             |
+----+------------------------------+
```

  - 224.0.0.1 (ip4-allnodes), 224.0.0.2 (ip4-allrouters)

- IPv6 (RFC 2375, RFC 3307):

```
|   8    |  4 |  4 |                   112 bits                    |
+------ -+----+----+-----------------------------------------------+
|11111111|flgs|scop|                   group ID                    |
+--------+----+----+-----------------------------------------------+
```

  - Permanent link-local multicast addresses:
    ff02::1 (ip6-allnodes), ff02::2 (ip6-allrouters)

Ethernet Mapping (IPv4)

- IEEE 802 MAC addresses support multicasts (lower two bits in the first octet indicate multicast)
- IANA owns a a block of Ethernet MAC addresses that start with the 23-bit prefix 01:00:5E.
- Half of this address block are multicast addresses (which means there is a fixed 25-bit prefix).
- The remaining 23 bits are filled by mapping the lower 23 bits of the IP multicast address into the MAC multicast address.
- As a consequence, $2^5 = 32$ IPv4 group addresses map to the same MAC address.

Ethernet Mapping (IPv6)

- IANA owns a a block of Ethernet MAC addresses that start with the 8-bit prefix 33:33.
- The lower 32 bits of the MAC address are filled by copying the lower 4 bytes of the IPv6 address into the MAC address.
- As a consequence, 2^{80} IPv6 group addresses map to the same MAC address.
- The question, of course, is how frequent collisions are in practice under normal conditions.
- To quote RFC 2464:

  "There is no protection from duplication through accident or forgery."

MBONE

- International project to establish a global multicast backbone
- Multicast is natively supported in the German research network (but currently not at IUB)
- Early experiements with audio and video conferencing (vat, rat, vic), shared whiteboards (wbd), shared editors, session directory (sdr), . . .
- Selected IETF meetings are broadcasted over the MBONE for several years now
- Not widely used outside of (multicast) research environments

Internet Multicast Services (RFC 3569)

- *Any-Source Multicast* (ASM):
  - IP datagrams are transmitted to a group G of nodes identified by a single IP multicast address.
  - Nodes may join and leave the group G any time, and there is no restriction on their location or number.
- *Source-Specific Multicast* (SSM):
  - IP datagrams are transmitted by a source S to an SSM destination address G, and receivers can receive this datagram by subscribing to channel (S, G).
- *Source-Filtered Multicast* (SFM):
  - ASM variant with filtered source addresses.
  - Supports whitelists (only a specific set of sources) and blacklists (all except a specific set of sources).

IPv4 Multicast Socket Extensions

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define IP_MULTICAST_IF     ...
#define IP_MULTICAST_TTL    ...
#define IP_MULTICAST_LOOP   ...
#define IP_ADD_MEMBERSHIP   ...
#define IP_DROP_MEMBERSHIP  ...

struct ip_mreq {
    struct in_addr imr_multiaddr;
    struct in_addr imr_interface;
};
```

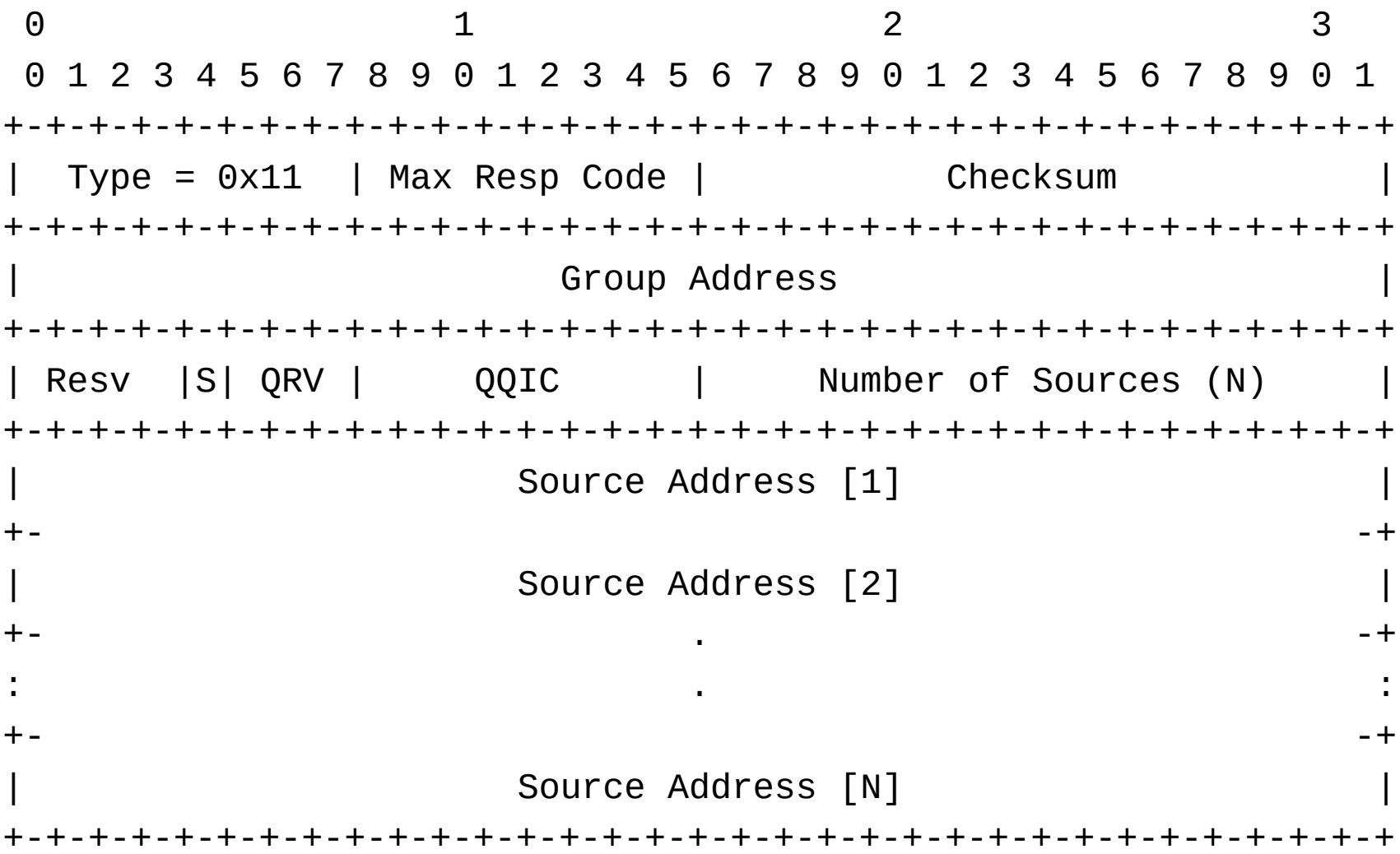
IPv6 Multicast Socket Extensions

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define IPV6_MULTICAST_IF   ...
#define IPV6_MULTICAST_HOPS ...
#define IPV6_MULTICAST_LOOP ...
#define IPV6_JOIN_GROUP     ...
#define IPV6_LEAVE_GROUP    ...

struct ipv6_mreq {
    struct in6_addr ipv6mr_multiaddr;
    unsigned int ipv6mr_interface;
};
```

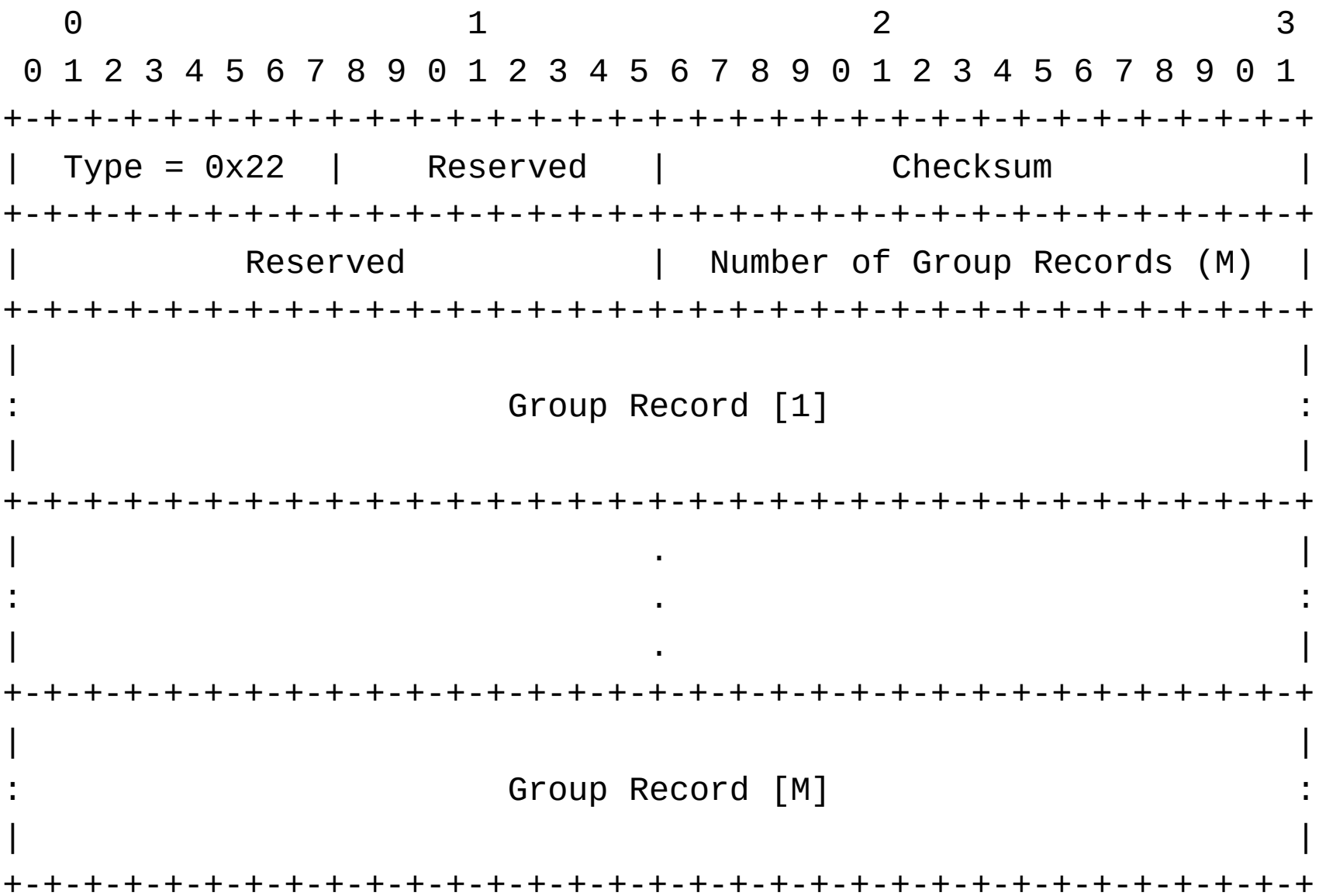
Internet Group Management Protocol

- IGMP version 3 (IGMPv3) is published in RFC 3376.
- IPv4 nodes use IGMP to report their group membership to the neighboring multicast router.
- A multicast router periodically sends a query message to 224.0.0.1 to check the group membership state.
- Group members respond with a membership report message.
- Multicast router maintain a per interface multicast group list.
- Group members can send unsolicited membership reports when they join/leave a group.

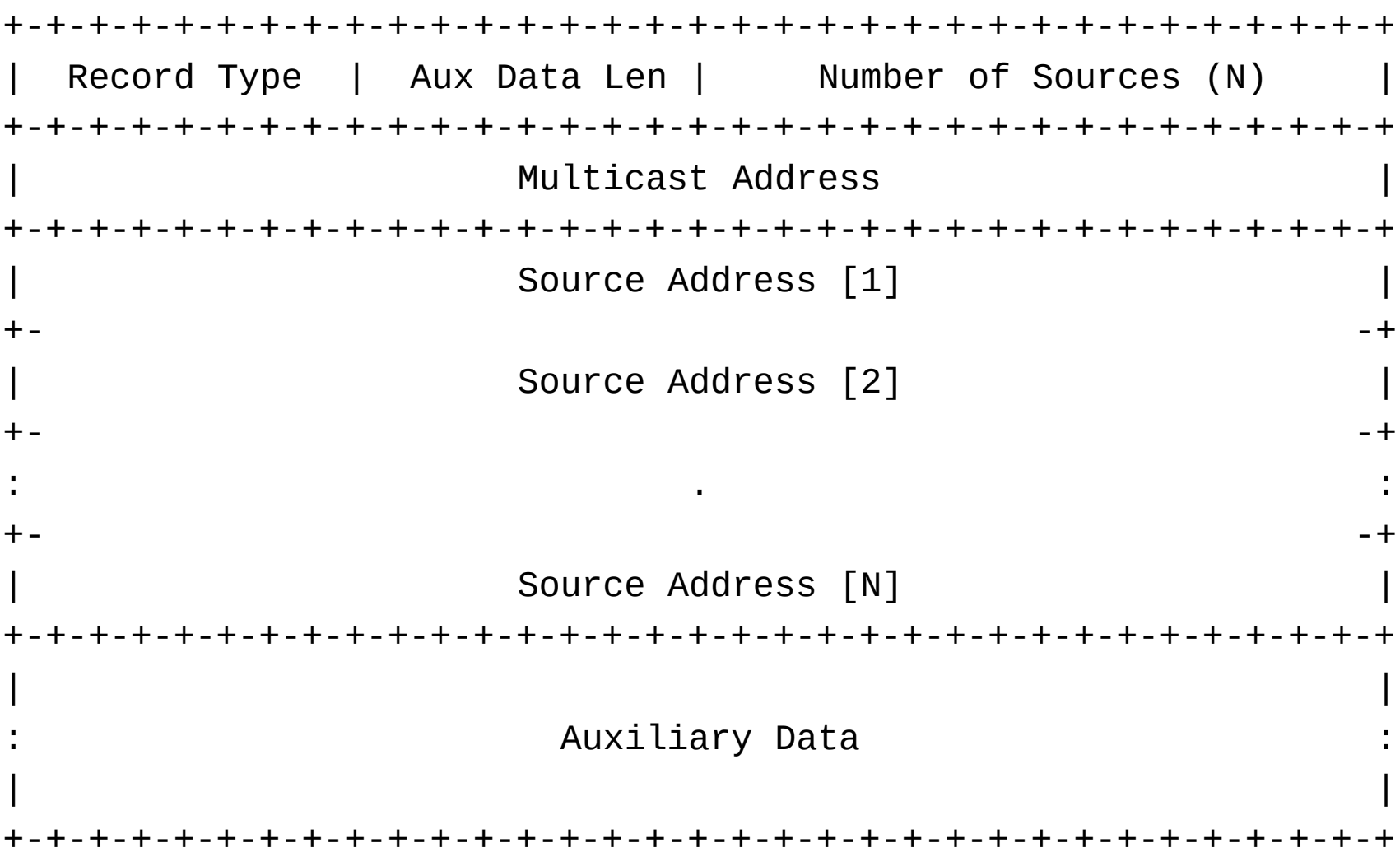
IGMPv3 Query Message



IGMPv3 Report Message



IGMPv3 Group Record



IGMP Snooping

- IEEE 802 bridges snoop IGMP packets to learn which port belongs to which IPv4 multicast group.
- Allows to suppress multicast traffic on segments / ports without group members.
- Widely supported by IEEE 802 bridges today.

Multicast Listener Discovery

- MLD version 2 (MLDv2) is published in RFC 3810
- Translation of the IGMPv3 protocol for IPv6 semantics

Multicast Routing

- Challenges:
  - Route data only to group members
  - Optimize routes from the source to the receivers
  - Maintain loop-free routes
  - Distribute multicast traffic over multiple links
  - Signalling (group membership) must scale well
- Several solutions have been tried...

Flooding and Spanning Trees

- Flooding
  - + Conceptually very simple
  - + Robust (all possible paths are explored)
  - - Requires to maintain history of last seen packets which can be memory intensive on high speed networks
- Spanning Trees:
  - + Robust and well understood technique
  - + Does not require much memory
  - - Group membership is not taken into account
  - - Concentrates multicast traffic on a subset of the links

Reverse Path Forwarding (RPF)

- Principle:
  1. When a multicast packet is received, note the source S and the interface I
  2. If I belongs to the shortest path to S, forward to all interfaces except I
  3. Otherwise, discard the packet

\+ Uses unicast routing tables to derive distribution trees

\+ Shortest path from source to destination

\+ Packets from different sources may use different links

\- Group membership is not taken into account

RPF Optimization

- Problem:
  - With plain RPF as described above, routers may receive packets via multiple paths
- Optimization:
  - Router determine whether they are on the shortest path between a neighbor and the multicast source before forwarding packets to a neighbor
  - The necessary information can be obtained from the link state database in OSPF

Flooding and Pruning

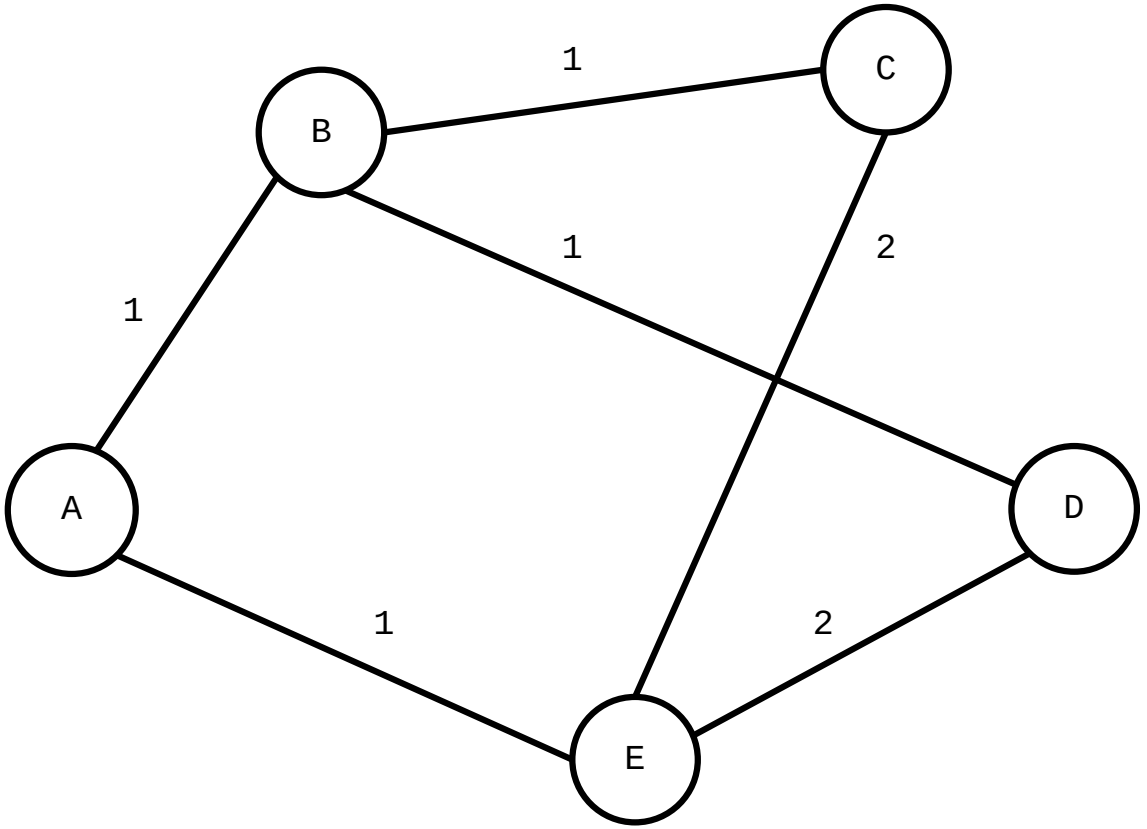
- Principle:
  1. Periodically, multicast packets are flooded
  2. Leaf routers who have no customers react by sending prune messages back towards the source
  3. Intermediate routers which do not have members on any interface will also send prune messages towards the source

\+ Computes minimal distribution trees

\- Requires periodic (global) flooding

\- Routers must keep state on a per-group and per-source basis

RPF Example



- Compute the reverse path forwarding (RPF) multicast trees for the sources A, B and D.

RPF Example: Unicast Routes

A	Dest.	Next
	B	B
	C	B
	D	B
	E	E

B	Dest.	Next
	A	A
	C	C
	D	D
	E	A

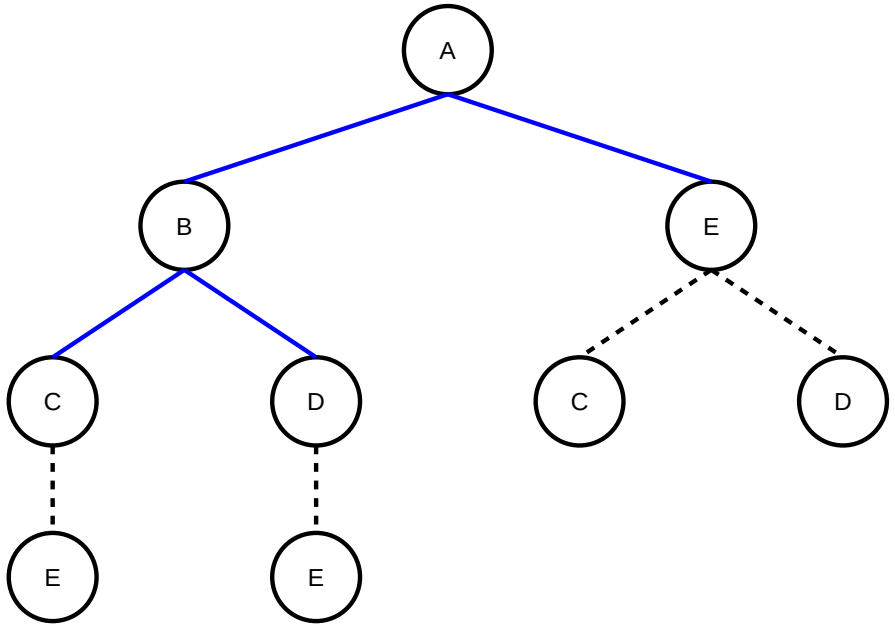
C	Dest.	Next
	A	B
	B	B
	D	B
	E	E

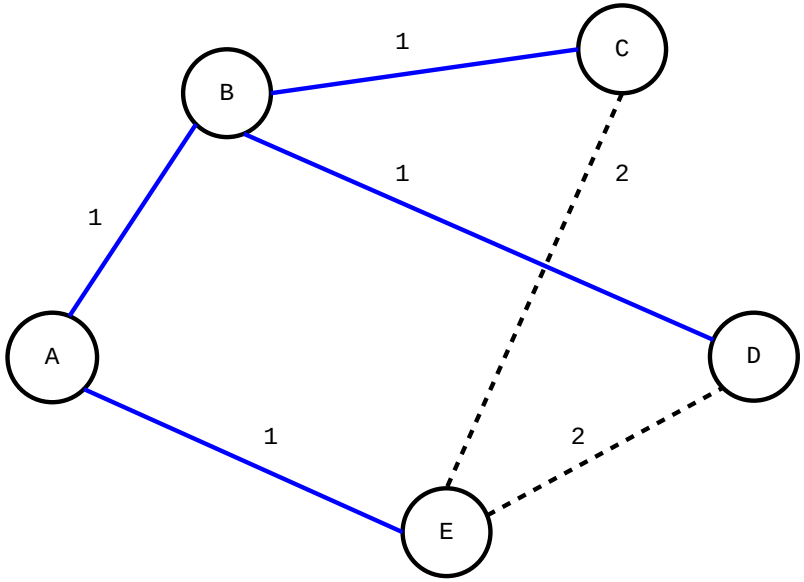
D	Dest.	Next
	A	B
	B	B
	C	B
	E	E

E	Dest.	Next
	A	A
	B	A
	C	C
	D	D

RPF Example: RPF Tree for A

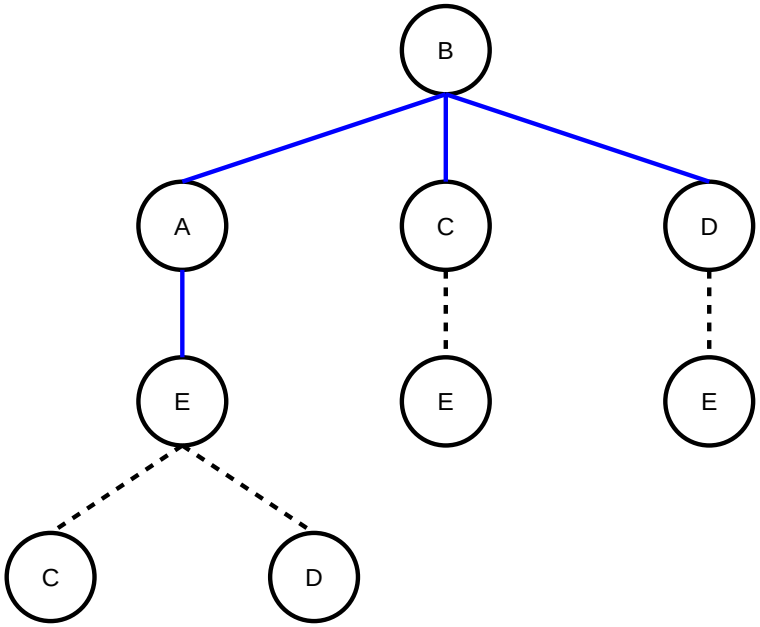
RPF tree for source A (dotted links may be pruned):

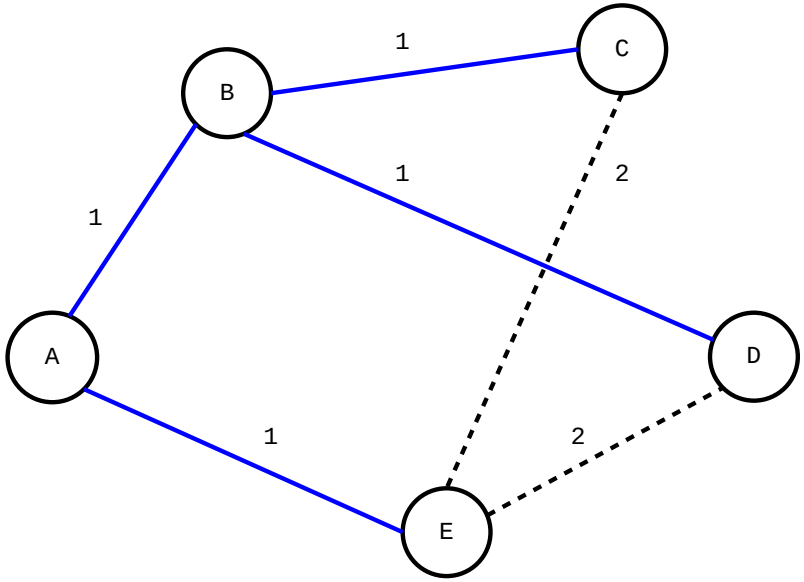




RPF Example: RPF Tree for B

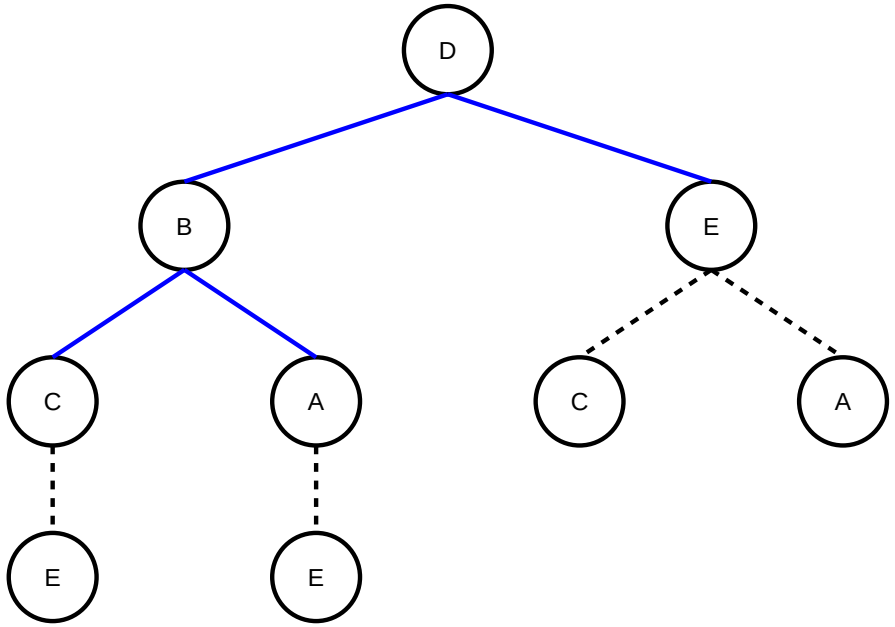
RPF tree for source B (dotted links may be pruned):

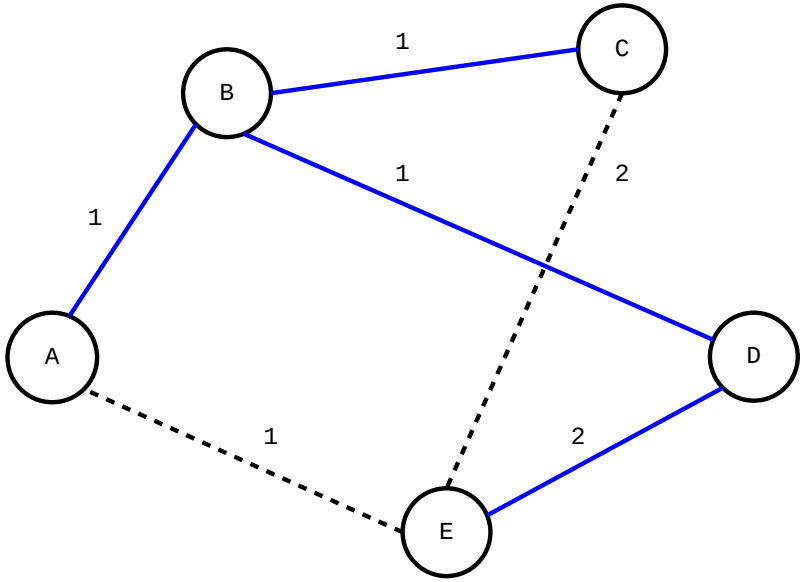




RPF Example: RPF Tree for D

RPF tree for source D (dotted links may be pruned):





Steiner Trees

- Steiner Trees:
  1. Given is a graph $G = (V, E)$ with a set of vertices V, a set of edges E and a cost function $c : E \mapsto \mathbb{N}$
  2. Let S be a subset of V. A steiner tree is a tree of G that spans S with a minimal total distance on its edges

+ Steiner trees minimize the number of links needed to connect the group members

- Steiner tree problem is known to be NP-hard
- Computation requires knowledge of the full topology
- Concentrates multicast traffic on a subset of the links
- Group membership changes lead to a recomputation and potentially massive changes of the distribution tree

Center-Based Trees

- Principle:
  1. Establish a well-known center for a multicast group
  2. Multicast receivers send join messages towards the center
  3. Intermediate routers keep state about the interfaces registered for a multicast group
- Center-based trees build a spanning tree for each group

\+ Limits the expansion of multicast traffic to the set of all group members

o Routers must keep state on a per-group basis

\- Choosing a center is difficult (optimal centers lead again to NP complete problems and instability)

Rendevous Points

- Instead of establishing a fixed center or core, it is possible to use rendevous points.
- The traffic originating from a new sender is encapsulated and unicast routed to a rendezvous point.
- Receivers send join messages towards the rendevous point, thereby establishing a path in the distribution tree.
- Distribution trees can be dynamically optimized by sending a source-specific join message towards the source (and subsequent pruning of the shared tree after establishing a shorter path).

Internet Multicast Routing

- Available Multicast Routing Protocols:
  - Distance Vector Multicast Routing Protocol (DVMRP)
  - Multicast Extensions to OSPF (MOSPF)
  - Protocol-Independent Multicast, Dense Mode (dense PIM)
  - Protocol-Independent Multicast, Sparse Mode (sparse PIM)
  - Border Gateway Multicast Protocol (BGMP)
- Today's protocol of choice is usually PIM in sparse mode.

DVMRP (RFC 1075)

- Oldest deployed multicast routing protocol based on RPF
- DVMRP routers exchange distance vectors that contain lists of potential sources and distances
- Popular `mrouted` implementation supports tunnels and was used to establish the multicast backbone (mbone).
- DVMRP has scaling problems because of the necessity to flood frequently.
- Early implementations did not implement pruning which made periodic flooding even more expensive

MOSPF (RFC 1584)

- Multicast OSPF (MOSPF) is a multicast extension of the OSPF routing protocol.
- MOSPF essentially computes multicast distribution trees from an augmented link state database.
- Augmented link state advertisements propagate information about the multicast groups active on each network.
- MOSPF computes distribution trees for each source/group pair.
- Distribution trees are cached, but they must be recomputed whenever a link state changes.

PIM Dense Mode (RFC 3973)

- Dense mode PIM uses RPF and is similar to DVMRP (but can use any unicast routing protocol).
- Dense mode works well in the following situations:
  - Senders and receivers are in close proximity to one another.
  - There are few senders and many receivers.
  - The volume of multicast traffic is high.
  - The stream of multicast traffic is constant.
- Dense mode PIM maintains distribution trees for sender/group pairs.
- Graft messages can be used to turn pruned branch back into a forwarding branch quickly.

PIM Sparse Mode (RFC 4601)

- Sarse mode PIM uses rendezvous points:
  - Senders initially send traffic to a rendevous point.
  - Receivers initially register with a rendevous point.
  - The initial forwarding path via a rendevous point can be optimized by the routers in the path.
- Sparse mode works well in the following situations:
  - There are few receivers in a group.
  - Senders and receivers are separated by WAN links.
  - The type of traffic is intermittent.

Border Gateway Multicast Protocol (RFC 3913)

- XXX

References

[1] S. Bhattacharyya. An Overview of Source-Specific Multicast (SSM). RFC 3569, Spring, July 2003.

[2] B. Cain, S. Deering, I Kouvelas, B. Fenner, and A Thyagarajan. Internet Group Management Protocol, Version 3. RFC 3376, Cereva Networks, Cisco Systems, AT&T Labs, Ericsson, October 2002.

[3] R. Vida and L. Costa. Multicast Listener Discovery Version 2 (MLDv2) for IPv6. RFC 3810, LIP6, June 2004.

[4] D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol. RFC 1075, BBN STC, Stanford University, November 1988.

[5] J. Moy. Multicast Extensions to OSPF. RFC 1584, Proteon, March 1994.

[6] A. Adams, J. Nicholas, and W. Siadak. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). RFC 3973, NextHop Technologies, ITT A/CD, January 2005.

[7] B. Fenner, M. Handley, H. Holbrook, and I Kouvelas. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 4601, AT&T Labs, UCL, Arastra, Cisco, August 2006.

[8] D. Thaler. Border Gateway Multicast Protocol (BGMP): Protocol Specification. RFC 3913, Microsoft, September 2004.

2. New Internet Transport Protocols

Classic Internet Transports

- Transmission Control Protocol (TCP):
  - stream oriented
  - reliable ordered delivery of data
  - connection oriented (establishment, delivery, teardown)
  - congestion aware (AIMD congestion control)
  - head of line blocking
- User Datagram Protocol (UDP):
  - packet oriented
  - unreliable, unordered delivery of data
  - connectionless (delivery)
  - congestion unaware
  - no blocking

Why Additional Transports?

- Would it not be nice to have . . .
  - a congestion aware datagram transport?
  - a reliable connection oriented transport which preserves message boundaries?
  - a reliable protocol which does not suffer from head of line blocking?
  - a transport protocol which can recover from failures in the underlying layers?
  - a transport which can bundle multiple independent streams?
  - a transport which starts quickly and recovers fast from packet loss?

Example #1: Loading a Web Page

- A web page usually consists of an HTML page which includes several embedded elements (e.g., graphics).
- Various ways to load a web page:
  - Load elements sequentially using separate TCP connections
    $\Rightarrow$ slow, high overhead for small elements (e.g., icons)
  - Load elements sequentially using a single TCP connection
    $\Rightarrow$ better, head-of-line blocking, network / server friendly
  - Load elements in parallel using multiple TCP connections
    $\Rightarrow$ faster, high overhead for small elements, no shared congestion state
- Ideally, one would like to use a single connection and multiple independent streams within this connection...

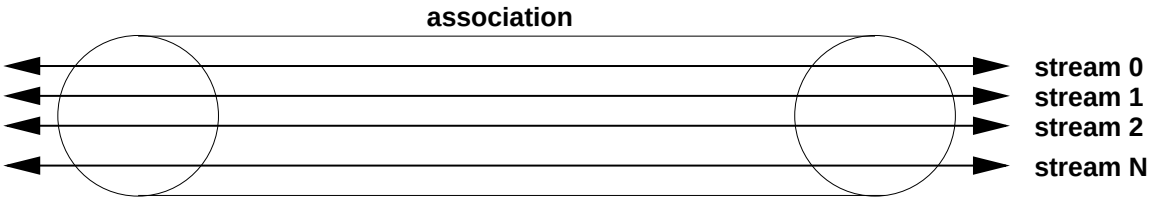
Example #2: Realtime Streams

- TCP's reliable ordered delivery is unsuited for real-time applications such as voice and video streaming.
- UDP is typically used, but . . .
  - UDP is not congestion aware
  - TCP streams may suffer from bad behaving UDP streams
  - voice and video may be treated differently as separate streams
- Ideally, one would like to use an unreliable congestion aware datagram transport.

Stream Control Transmission Protocol (SCTP)

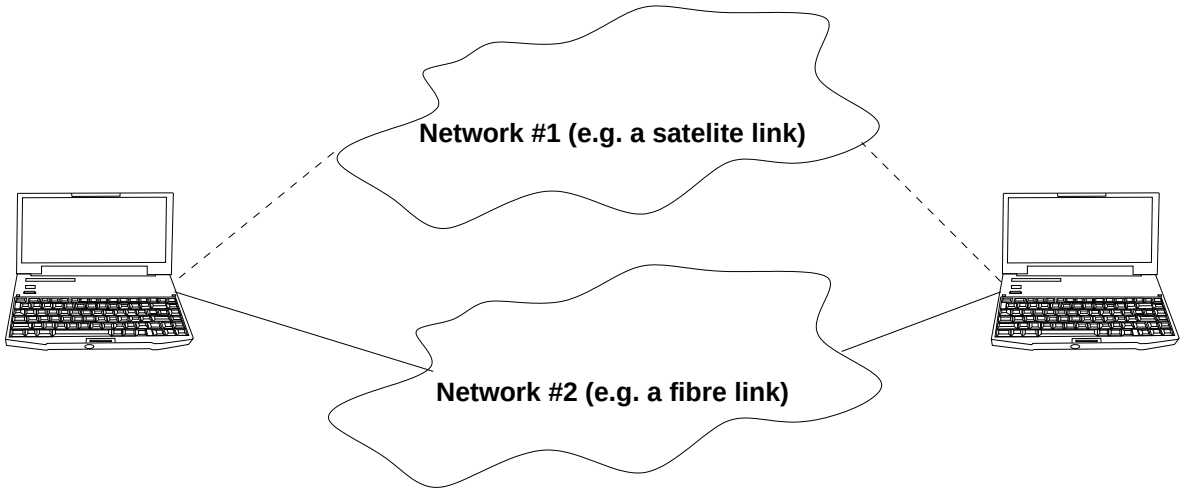
- The Stream Control Transmission Protocol (SCTP) provides the following services [1]:
  - Reliable, ordered and unordered delivery of data
  - Message oriented (preserves application layer framing)
  - Multiple independent streams bundled in an association
  - Multi-homing of association endpoints for fast failover
  - Initiation protection and graceful shutdown
- Initial version developed 1998-2000 in the IETF mainly as a transport for signaling protocols.
- SCTP has much wider applicability and is continuously extended.

SCTP Streams and Associations



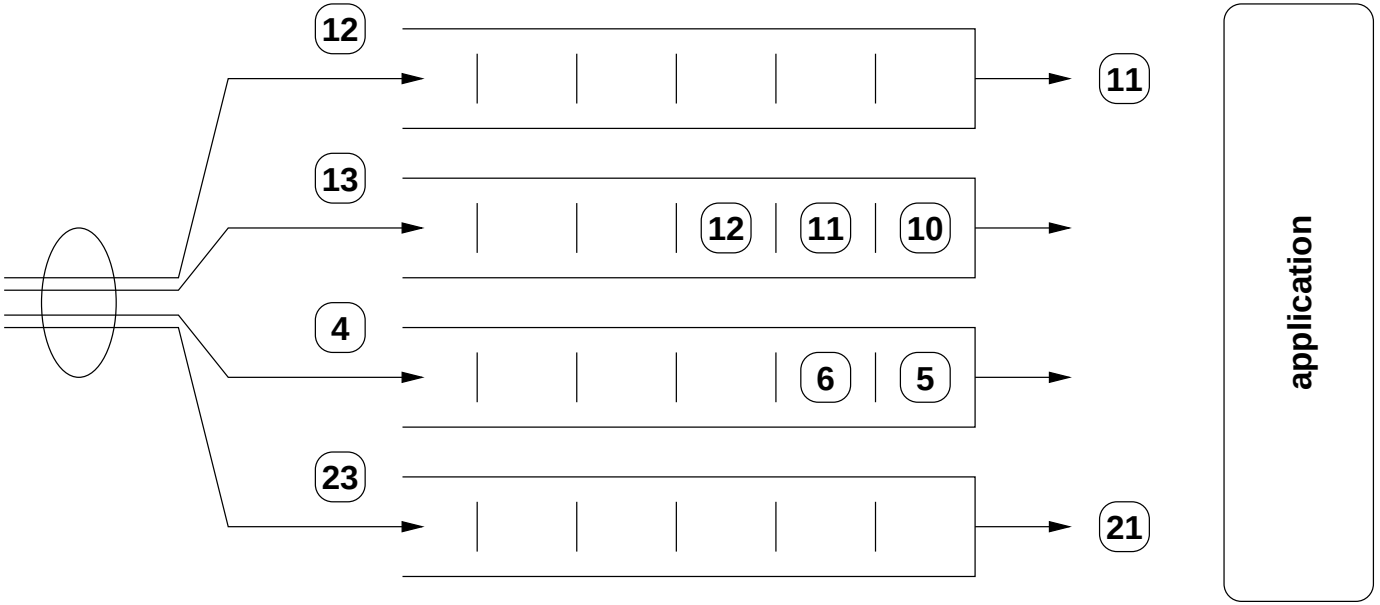
- Data transfer between two SCTP hosts takes place in the context of an association.
- An association may contain multiple data streams and each stream has the property of independently sequenced delivery.
- A message lost in one stream thus does not affect the delivery in other streams.
- SCTP accomplishes multi-streaming by creating independence between data transmission and data delivery.

SCTP Multi-homing



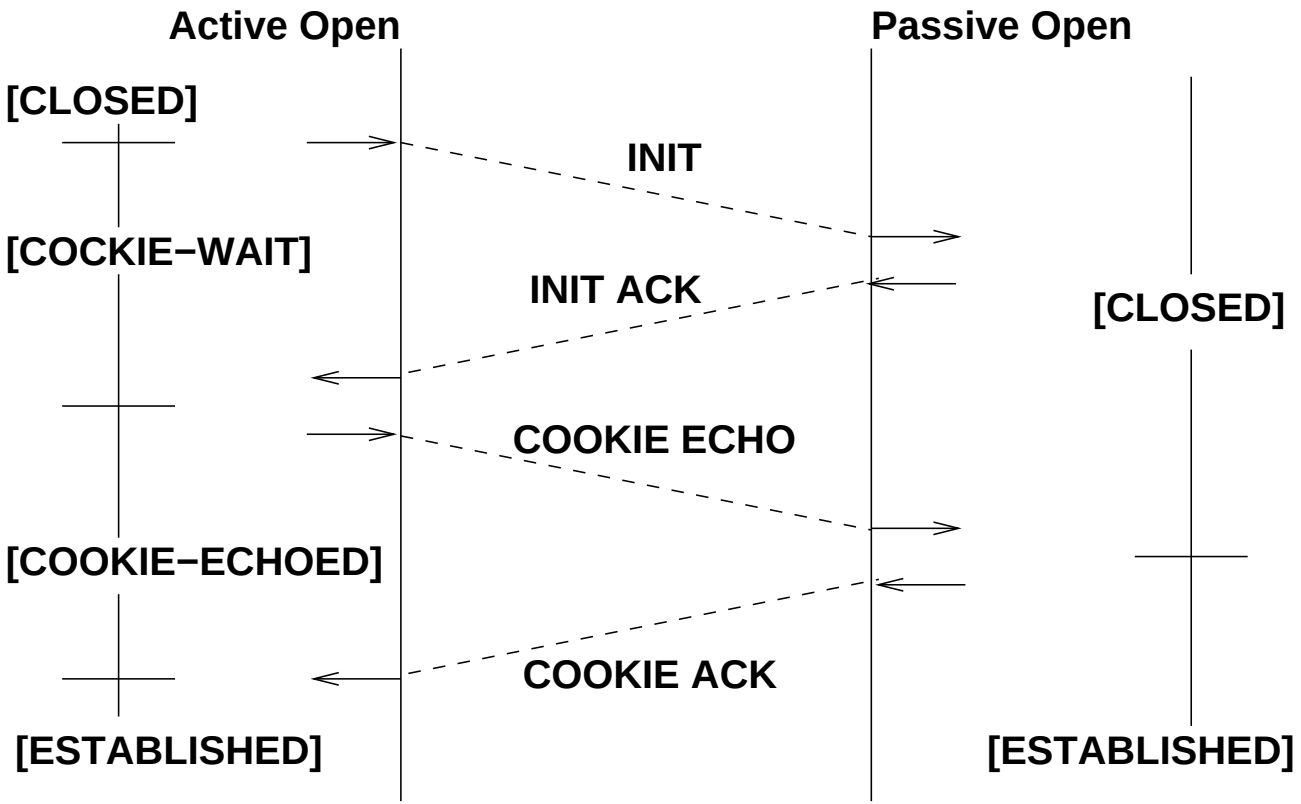
- SCTP allows SCTP endpoints to have multiple IP addresses.
- This multi-homing feature provides the benefit of potentially greater survivability of an SCTP association in the presence of network failures.
- Multi-homing is not designed as a load balancing mechanism.

SCTP Sequencing



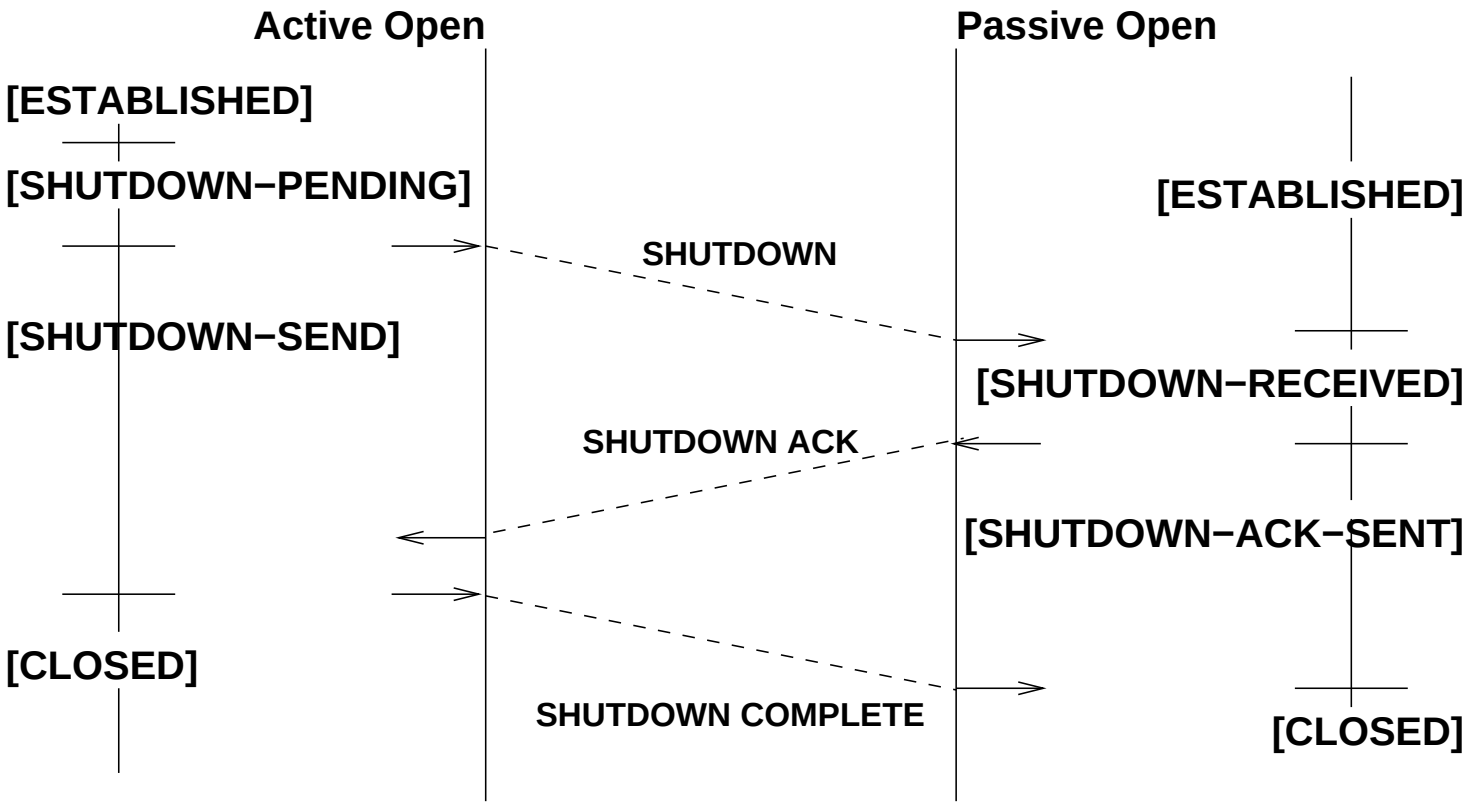
- Every stream has its own independent sequence number space
- Streams progress independently (no head-of-line blocking)

SCTP Association Establishment



- A TCB is not allocated on the server side before the `COOKIE-ECHO` has been received to avoid TCP's `SYN`-flooding problem

SCTP Association Teardown



- SCTP supports only a full teardown procedure (TCP's half closed connections do not exist in SCTP)

SCTP State Machine

State	Description
`CLOSED`	Initial and final state
`COOKIE-WAIT`	Waiting for a cookie
`COOKIE-ECHOED`	Cookied echoed to the server
`ESTABLISHED`	Association established
`SHUTDOWN-PENDING`	Shutdown requested by application
`SHUTDOWN-SENT`	Shutdown initiated
`SHUTDOWN-RECEIVED`	Shutdown request received
`SHUTDOWN-ACK-SENT`	Shutdown request acknowledged

- The SCTP state machine has the states shown above and is slightly simpler than TCP's state machine.
- For a description of the transitions, see RFC 2960.

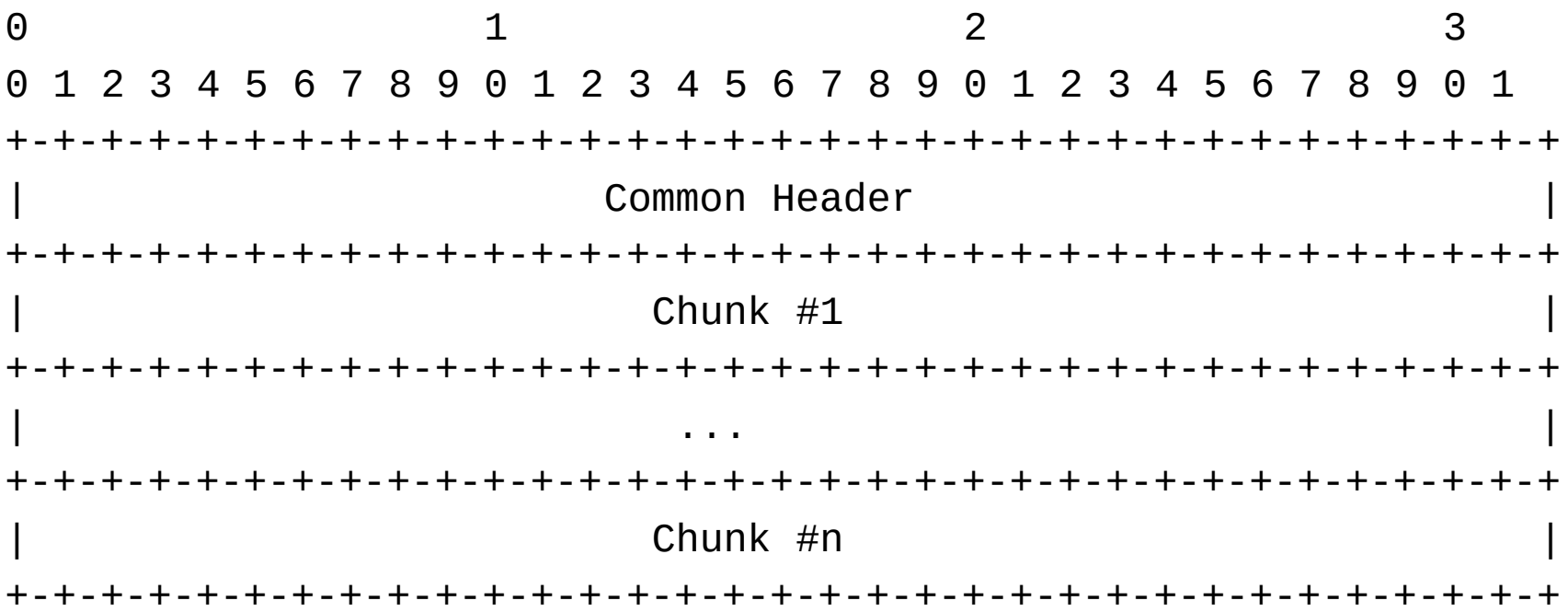
SCTP Fragmentation / Reassembly / Bundling

- In order to preserve application layer message boundaries, SCTP has to perform fragmentation and reassembly, since SCTP messages should not exceed the path MTU.
- Fragmentation / reassembly happens on the data chunk level and not on an message level.
- If a receiver runs out of buffer space while waiting for more fragments to arrive, it may pass the incomplete message to the application via a special API.
- Applications can also request the bundling of chunks so that they are shipped in a single SCTP message.

SCTP Congestion Control

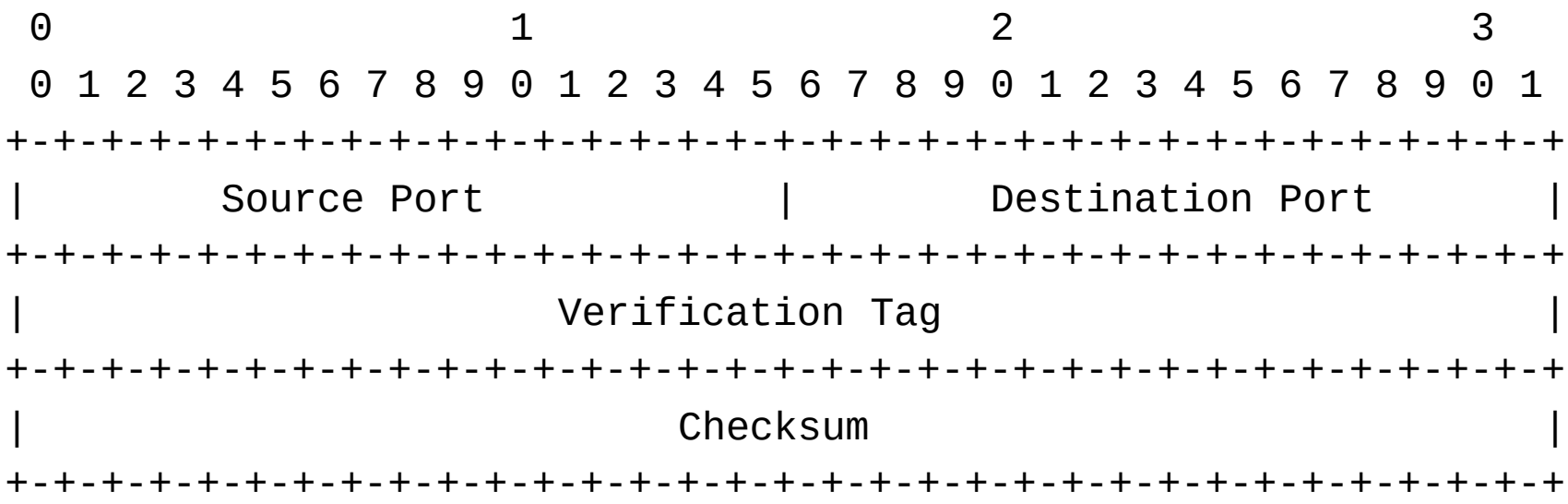
- Similar to TCP, but with a number of notable differences:
  - SCTP uses selective acknowledgements (SACKs) which speeds up the detection of loss and increases bandwidth utilization.
  - During congestion avoidance, `cwnd` is increased by the number of acknowledged bytes and not the number of segments.
  - During congestion avoidance, `cwnd` can only be increased when the full `cwnd` is utilized.
  - SCTP begins fast retransmission after receipt of four duplicate acknowledgedments (TCP after three).

SCTP Packet Format



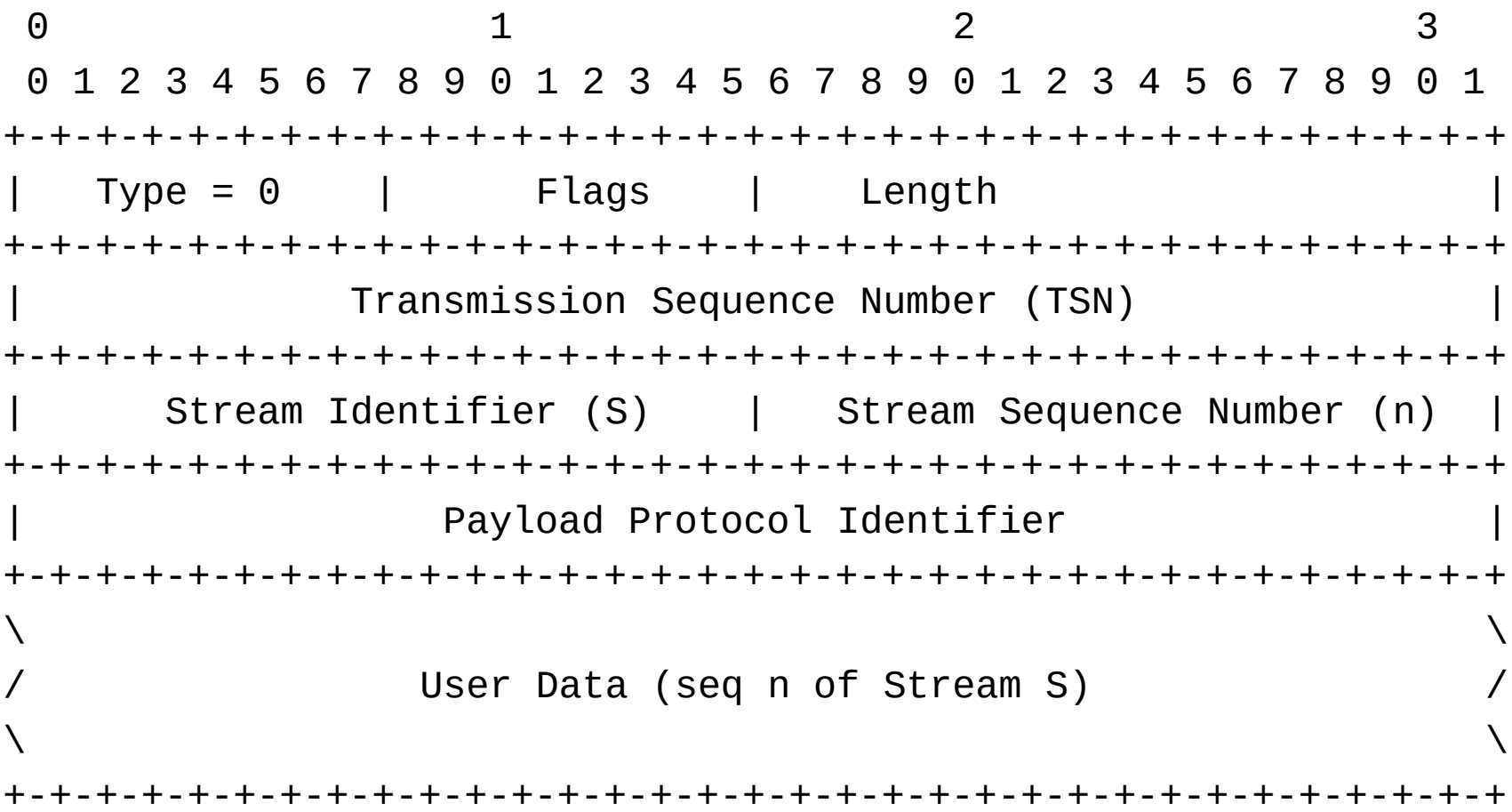
- An SCTP packet is composed of a common header and chunks.
- A chunk contains either control information or user data.
- Multiple chunks can be bundled into one SCTP packet up to the MTU size.

SCTP Common Header Format



- The `Source Port` and `Destination Port` fields contains the port number used by the sending / receiving application layer process.
- The `Verification Tag` field is used by the receiver to validate the sender of the SCTP packet.
- The `Checksum` field contains a 32-bit CRC checksum as specified in RFC 3309.

SCTP Data Chunk Format



SCTP Data Chunk Format

- The `Type` field indicates the chunk type. Data chunks use the type number `0`.
- The `Flags` field contains a set of binary flags:
  - `U`: The data chunk is unordered and there is no `Stream Sequence Number` assigned to the data chunk.
  - `B`: Beginning of a fragment of a user message.
  - `E`: Ending of a fragment (last fragment) of a user message.
- The `Length` field indicates the size of the chunk in bytes including the chunk header fields.

SCTP Data Chunk Format

- The `Transmission Sequence Number` field contains the transmission sequence number which is also used by the receiver to reassemble messages.
- The `Stream Identifier` identifies the stream to which the following data belongs.
- The `Stream Sequence Number` identifies the stream sequence number of the following user data within the stream identified by the `Stream Identifier`.
- The `Payload Protocol Identifier` identifies the upper layer application protocol and is opaque from the viewpoint of an SCTP protocol engine.
- The `User Data` is of variable length and contains the actual payload.

DCCP Motivation

- Multimedia streaming applications and online games often prefer timeliness over reliability.
- There is a potential that increasing non-congestion-controlled UDP traffic may lead to congestion collapse.
- Implementation of effective congestion control in application protocols is difficult.
- UDP flows are hard for firewalls to handle due to a lack of a setup and teardown exchange.
- See RFC 3714 for a discussion why DCCP is not just TCP with relaxed reliability.

DCCP Features

- Unreliable flows of datagrams.
- Reliable handshakes for connection setup and teardown.
- Reliable negotiation of options, including negotiation of a suitable congestion control mechanism.
- Mechanisms allowing servers to avoid holding state for unacknowledged connection attempts and already-finished connections.
- Support for Early Congestion Notification (ECN)
- Acknowledgement mechanisms communicating packet loss and ECN information. Acks are transmitted as reliably as the relevant congestion control mechanism requires, possibly completely reliably.
- Optional mechanisms that tell the sending application, with high reliability, which data packets reached the receiver, and

DCCP Congestion Control

- DCCP supports multiple congestion control mechanisms that are identified by so called congestion control identifiers (CCIDs).
- Two congestion control mechanism have been defined so far:
  - TCP-like Congestion Control (CCID-2) [RFC 4341]
  - TCP-Friendly Rate Control (TFRC) (CCID-3) [RFC 4342]
- The congestion control mechanism can be negotiated.
- Additional congestion control mechanism can be added in the future.

DCCP Messages

- DCCP-Request
- DCCP-Response
- DCCP-Data
- DCCP-Ack
- DCCP-DataAck
- DCCP-CloseReq
- DCCP-Close
- DCCP-Reset
- DCCP-Sync
- DCCP-SyncAck

DCCP Connection Establishment

```
  Client State                                    Server State
     CLOSED                                          LISTEN
1.   REQUEST    -->        Request          -->
2.              <--        Response         <--     RESPOND
3.   PARTOPEN   -->     Ack, DataAck        -->
4.              <--  Data, Ack, DataAck     <--     OPEN
5.   OPEN       <->  Data, Ack, DataAck     <->     OPEN
```

DCCP Connection Teardown (#1)

```
   Client State                                    Server State
      OPEN                                            OPEN
 1.               <--        CloseReq        <--    CLOSEREQ
 2.    CLOSING     -->         Close          -->
 3.               <--         Reset          <--    CLOSED (LISTEN)
 4.    TIMEWAIT
 5.    CLOSED
```

- Server initiates teardown procedure.
- Client takes the `TIMEWAIT` burden.

DCCP Connection Teardown (#2)

```
   Client State                                  Server State
      OPEN                                          OPEN
1.    CLOSING    -->          Close          -->
2.               <--          Reset          <--    CLOSED (LISTEN)
3.    TIMEWAIT
4.    CLOSED
```

- Client initiates teardown procedure.
- Client takes the `TIMEWAIT` burden.

DCCP Connection Teardown (#3)

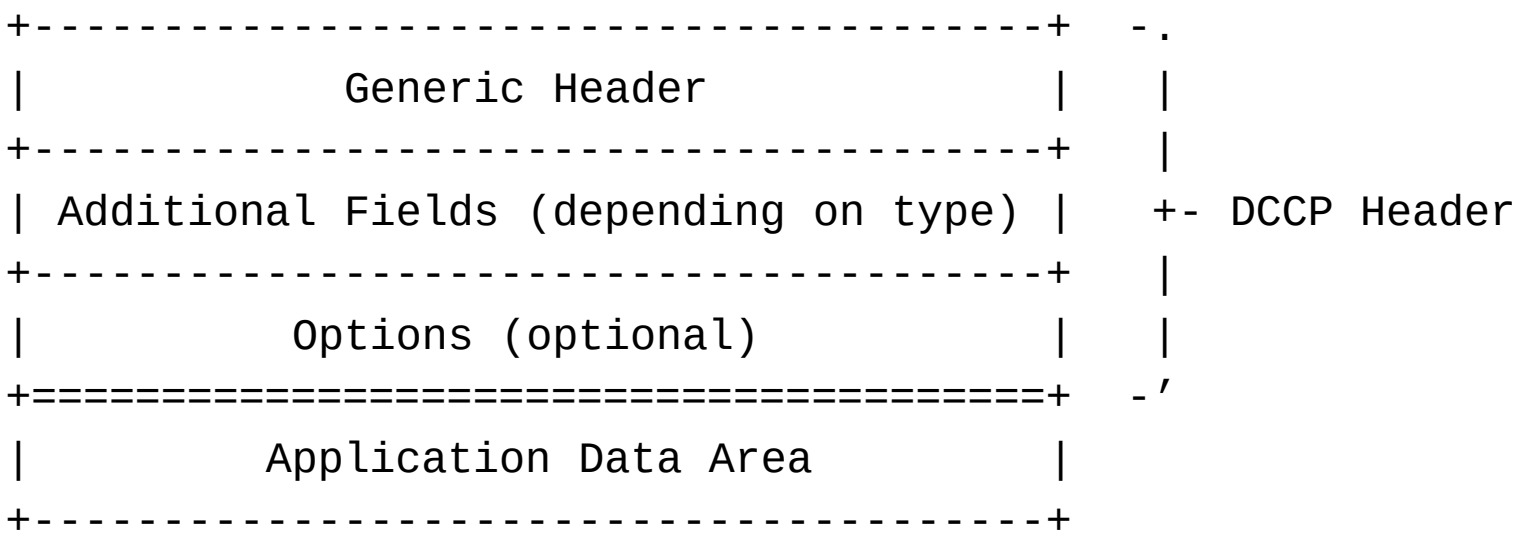
```
   Client State                                    Server State
      OPEN                                            OPEN
1.                 <--          Close          <--   CLOSING
2.    CLOSED       -->          Reset          -->
3.                                                    TIMEWAIT
4.                                                    CLOSED (LISTEN)
```

- Server initiates teardown procedure.
- Server takes the `TIMEWAIT` burden.

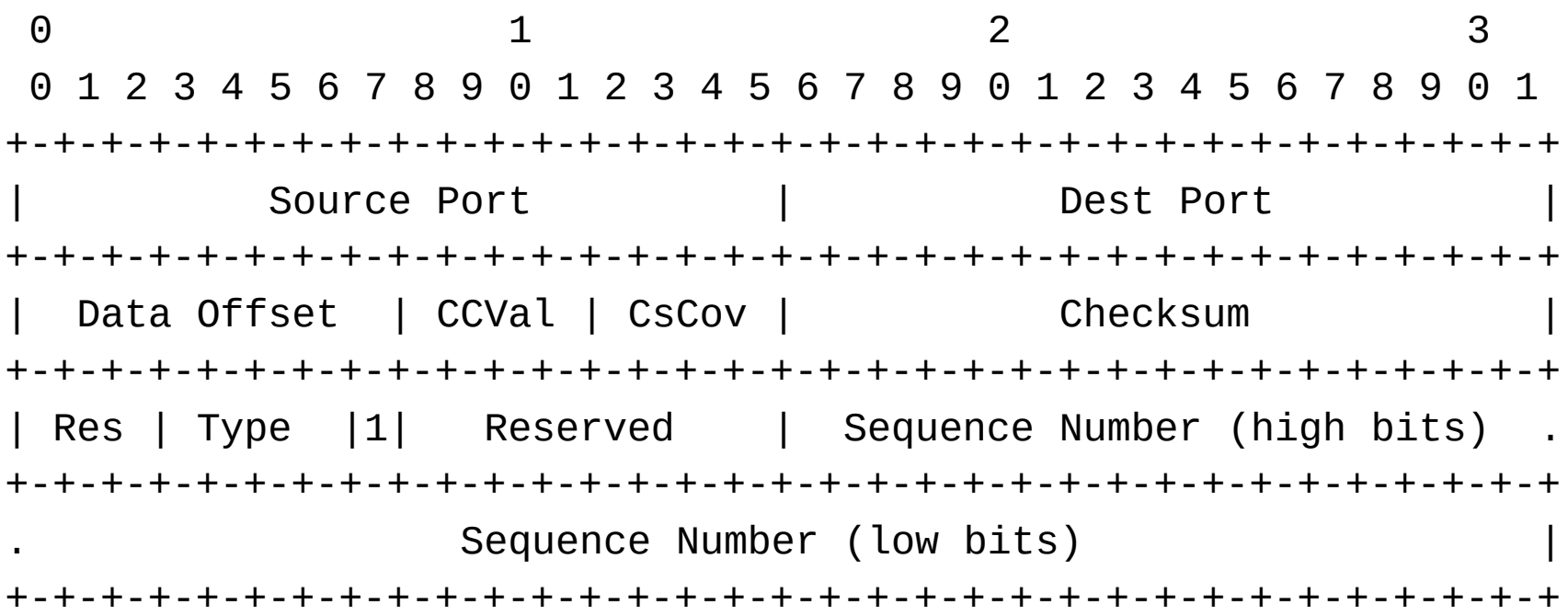
DCCP State Machine

- See RFC 4340 section 8.4. . .

DCCP Packet Formats

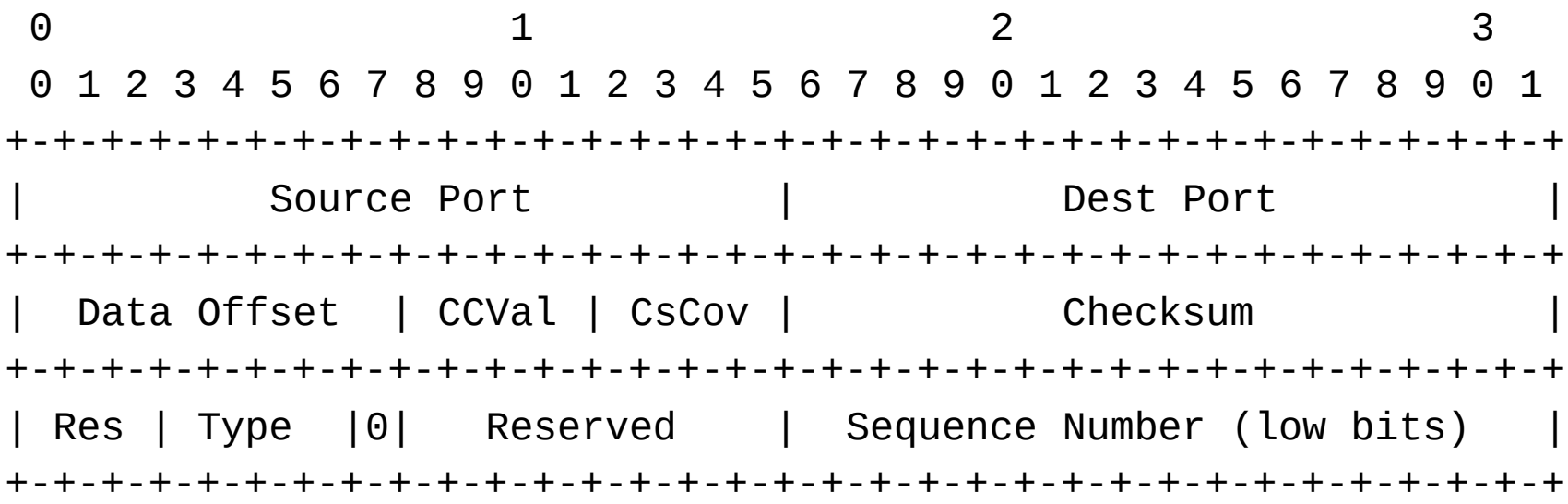


DCCP Generic Header



- The generic header with the “Extended Sequence Number” bit set to 1.

DCCP Generic Header



- The generic header with the “Extended Sequence Number” bit set to 0.

References

[1] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960, Motorola, Cisco, Siemens, Nortel Networks, Ericsson, Telcordia, UCLA, ACIRI, October 2000.

[2] L. Ong and J. Yoakum. An Introduction to the Stream Control Transmission Protocol (SCTP). RFC 3286, Ciena Corporation, Nortel Networks, May 2002.

[3] R. Stewart, Q. Xie, L. Yarroll, K. Poon, and M. Tuexen. Sockets API Extensions for Stream Control Transmission Protocol (SCTP). Internet Draft <draft-ietf-tsvwg-sctpsocket-13.txt>, Cisco Systems, Motorola, TimeSys Corp, Sun Microsystems, Univ. of Applied Sciences Muenster, June 2006.

[4] R. Stewart and C. Metz. SCTP: New Transport Protocol for TCP/IP. *IEEE Internet Computing*, November 2001.

[5] A.L. Caro, J.R. Iyengar, P.D. Amer, S. Ladha, G.J. Heinz, and K.C. Shah. SCTP: A Proposed Standard for Robust Internet Data Transport. *IEEE Computer*, 36(11), November 2003.

[6] S. Fu and M. Atiquzzaman. SCTP: State of the Art in Research, Products, and Technical Challenges. *IEEE Communications Magazine*, 42(4), April 2004.

[7] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). RFC 4340, UCLA, UCL, ICIR, March 2006.

3. Internet Quality of Service

Elastic vs. Inelastic Traffic

- Elastic Traffic:
  - Can adjust to changes in delay and throughput
  - Traditional type of traffic in the Internet
  - Examples: E-mail, file transfers
- Inelastic Traffic:
  - Does not easily, if at all, adapt to changes in delay and throughput
  - Examples: Video and audio streams, real-time stock trading

Quality of Service (QoS) Parameters

- Throughput:
  - Some applications require a minimum throughput.
- Delay:
  - Some applications require a minimum delay.
- Jitter:
  - Some applications do not tolerate arbitrary delay variations (jitter).
- Packet loss:
  - Real-time applications vary in the amount of packet loss, if any, they can sustain.

Flows

- Quality of Service (QoS) support requires to treat aggregations of packets that belong together.
- RFC 1633 introduces the concept of a flow as follows:
  - A flow is a distinguishable stream of related datagrams that results from a single user activity and requires the same QoS.
  - A flow has a single source but may have N destinations.
  - An N-way teleconference will generally require N flows, one originating at each site.
- A transport connection carrying a video stream is an example for a single flow.

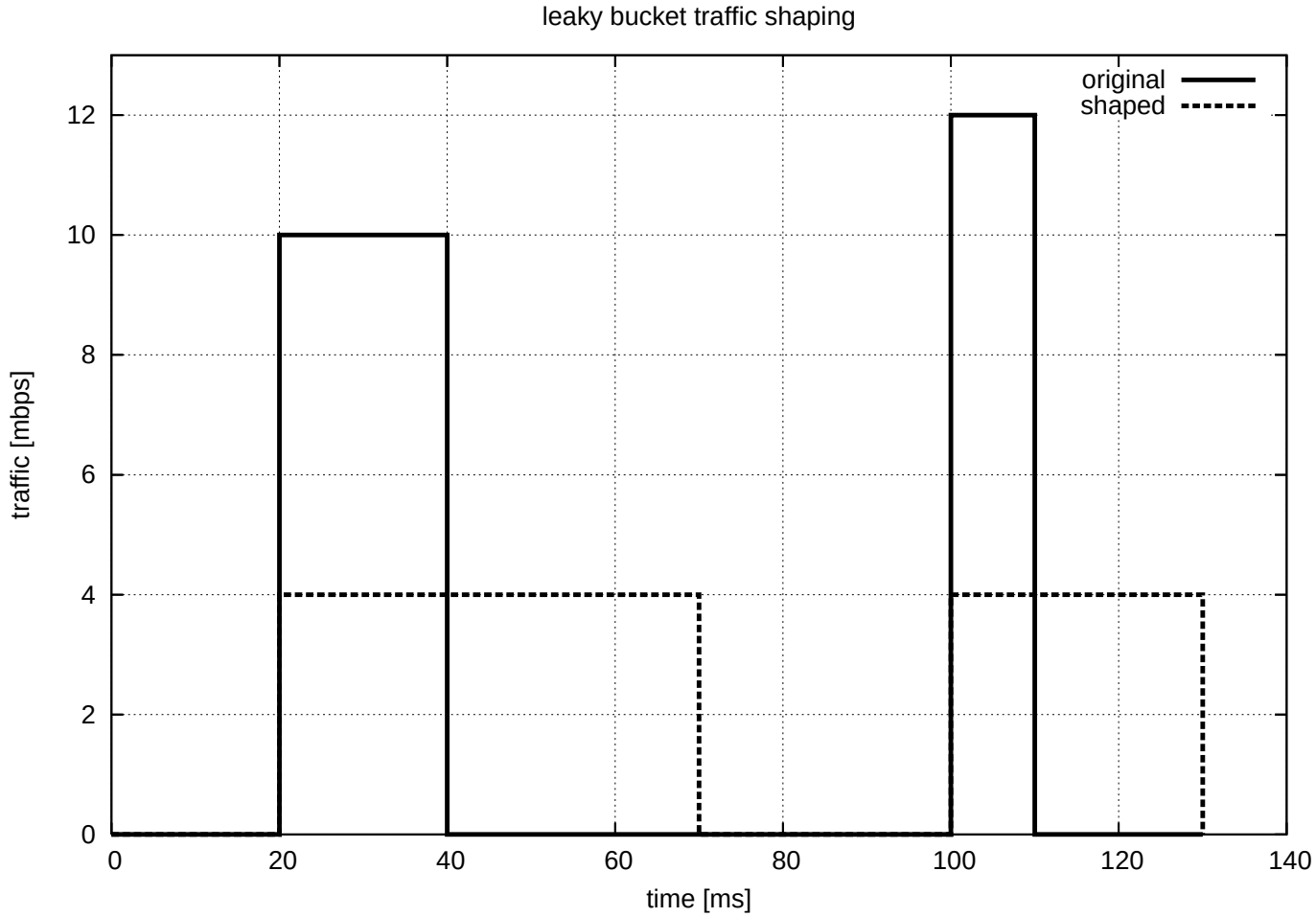
Controlling Quality of Service (QoS)

- Admission control:
  - New traffic flows are only admitted if there are enough resources to handle the flows.
  - Requires signaling phase before the data transfer.
- Routing algorithm:
  - Routing decisions may be based on a variety of QoS parameters, not just minimum delay.
- Queueing discipline:
  - Queue packets to meet QoS constraints (where necessary).
- Discard policy:
  - Discard packets (manage congestion) to meet QoS constraints.

Traffic Shaping: Leaky Bucket

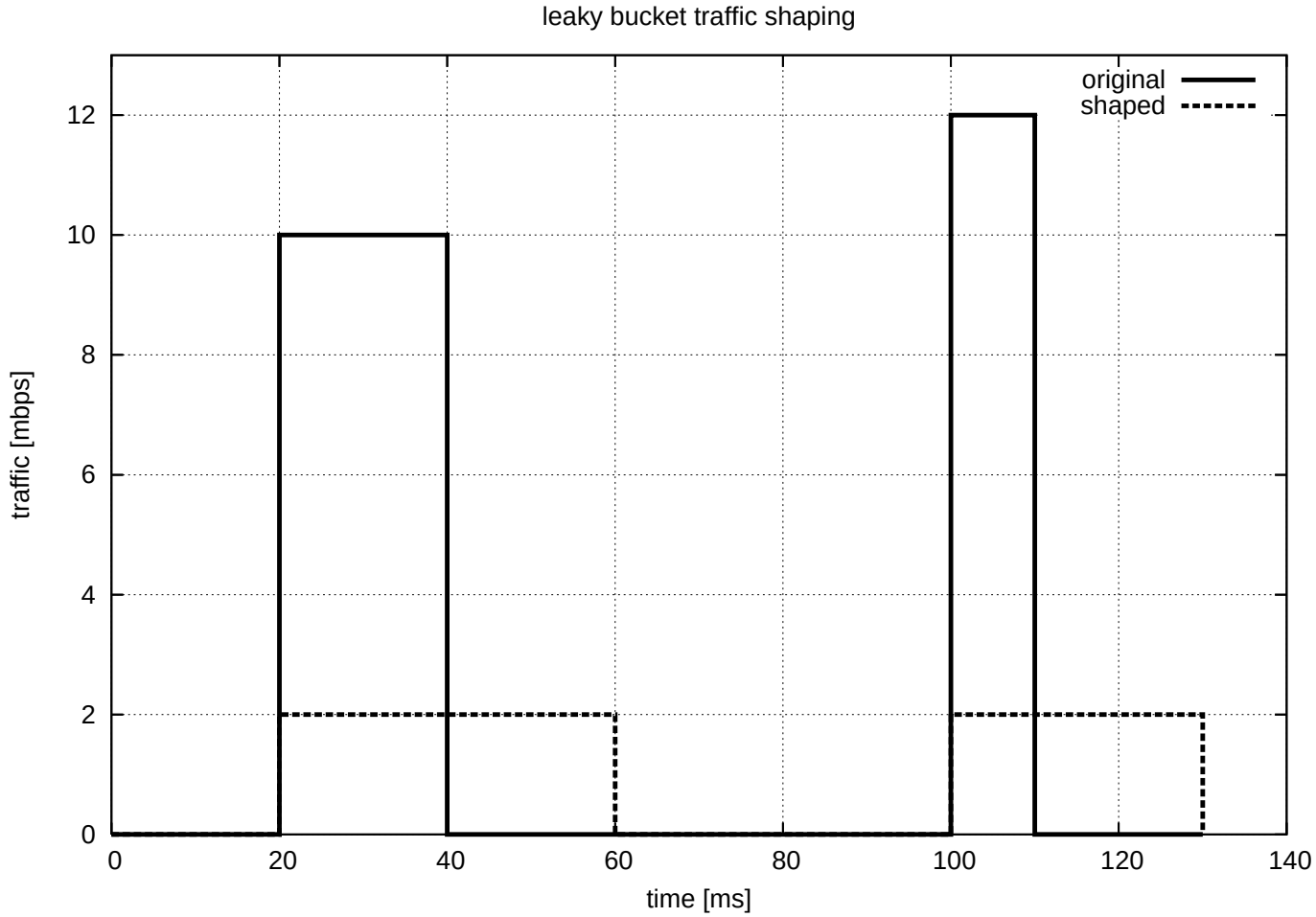
- Leaky buckets shape bursty traffic into a smooth traffic stream.
- The leaky bucket uses a conceptual bucket that can be filled with packets and which has a leak through which packets leave the bucket with a fixed rate.
- Arriving packets will be discarded if the bucket is full.
- Leaky bucket parameters:
  - bucket capacity C
  - departure rate R
- A leaky bucket is a single server queueing with constant service time R and limited queue size C.
- The bucket capacity can be counted in packets or bytes.

Leaky Bucket Example #1



- Leaky bucket with $C = 200kb$ and $R = 4mbps$.

Leaky Bucket Example #2



- Leaky bucket with $C = 40kb$ and $R = 2mbps$.

Traffic Shaping: Token Bucket

- Token buckets allow some burstiness if some capacity has not been used recently.
- The token bucket uses a conceptual bucket that is filled with tokens with a constant arrival time. Packets are allowed to leave the token bucket system once there are tokens available.
- Token bucket parameters:
  - token bucket capacity C
  - token arrival rate R
  - maximum output rate M
- Each token may represent a packet or a fixed number of bytes.

Token Bucket Burst Length

- Calculation of the token bucket burst length S must take into acount that token continue to arrive while the burst is being transmitted:
  - The output burst contains a maximum of $C + RS$ bytes.
  - The maximum number of bytes transmitted can also be written as MS.
  - Thus, we have $C + RS = MS$. Solving this equation to get S leads to:

$$S = \frac{C}{(M - R)}$$

- To fully understand how a token bucket behaves, it is also necessary to know how many tokens are initially in the token bucket.

Token Bucket Example

- Consider a token bucket with a token capacity of $C = 3$ tokens and a token arrival rate of $R = 1$ token per ms.
- The buffer used to hold packets can accommodate up to $B = 4$ packets.
- Assume that the system has been idle for some time before the following sequence of packets arrives.
- Computer for the packet arrival times $\{1, 1.1, 1.5, 2, 2.7, 2.9, 3, 3.1, 3.2, 6\}$
  - the departure times;
  - the number of tokens in the bucket;
  - the list of queued packets.

Token Bucket Example (cont.)

Packet	Arrival Time	Departure Time	Token Count	Queued Packets
1	1	1	2	$\{\}$
2	1.1	1.1	1	$\{\}$
3	1.5	1.5	0	$\{\}$
4	2	2	0	$\{\}$
5	2.7	3	0	$\{5\}$
6	2.9	4	0	$\{5, 6\}$
7	3	5	0	$\{6, 7\}$
8	3.1	6	0	$\{6, 7, 8\}$
9	3.2	7	0	$\{6, 7, 8, 9\}$
10	6	8	0	$\{9, 10\}$

Token and Leaky Bucket Combination

- Token buckets still allow some bursts, even though the maximum burst interval can be regulated by careful selection of the parameters.
- It is often desirable to further control the traffic peaks.
- Token buckets can be combined with leaky buckets to address this problem:
  - The token bucket does the primary traffic shaping.
  - The leaky bucket takes care of any remaining peaks.
- Policing such combined mechanisms can be tricky.
- Requires a good understanding of the actual traffic mix to be effective (requires ongoing measurements).
- Often considered to be part of *traffic engineering*.

Fair Queueing (FQ)

- Fair Queuing Idea:
  - Introduce a separate queue for each flow and each output interface.
  - Process queues in a round-robin fashion.
- Problem:
  - Packets have different sizes which can lead to unfairness.
- Solution:
  - Compute the time when a packet will be finished using byte-by-byte round robin.
  - Transmit packets in the order of their finishing times.

Fair Queuing Example (A. Tanenbaum)

- Assume the packets A (6 bytes), B (4 bytes), C (2 bytes), D (4 bytes), E (4 bytes) arrive simultaneously at a fair queuing interface.
- Compute the finish times for these five packets.
- Solution:

```
A -> 01 06 11 15 19 20
B -> 02 07 12 16
C -> 03 08
D -> 04 09 13 17
E -> 05 10 14 18
```

- Finish times: C (8), B (16), D(17), E(18), A (20)

Weighted Fair Queueing (WFQ)

- Weighted Fair Queueing Idea:
  - Introduce priorities so that some queues are processed more often.
  - Give some queues more than one byte per tick.
- Imlementation:
  - Compute the finish times and insert packets into a priority queue sorted by finish times.
  - Take the relative weight of the queues into account when computing the finish times.

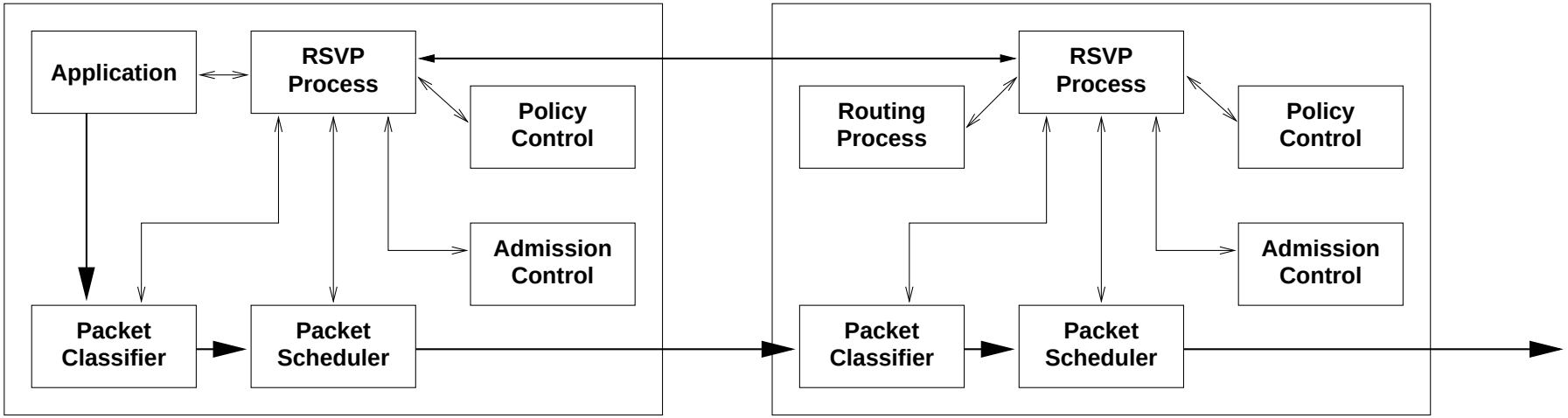
WFQ Example (H. Schulzrinne)

- A router with $N = 3$ queues uses weighted fair queueing (WFQ). The weights of the queues are $w_1 = 0.5$, $w_2 = 0.25$, $w_3 = 0.25$.
- Packets in the first queue are 100 byte long while packets in the second and third queue are 300 byte long.
- Assume that the buffer for each queue is full and the first packet in the second queue arrived shortly after the first packet of the first queue and the first packet of the third queue arrived shortly after the first packet of the second queue.
- In which order will the WFQ scheduler serve the packets?

WFQ Example (cont.)

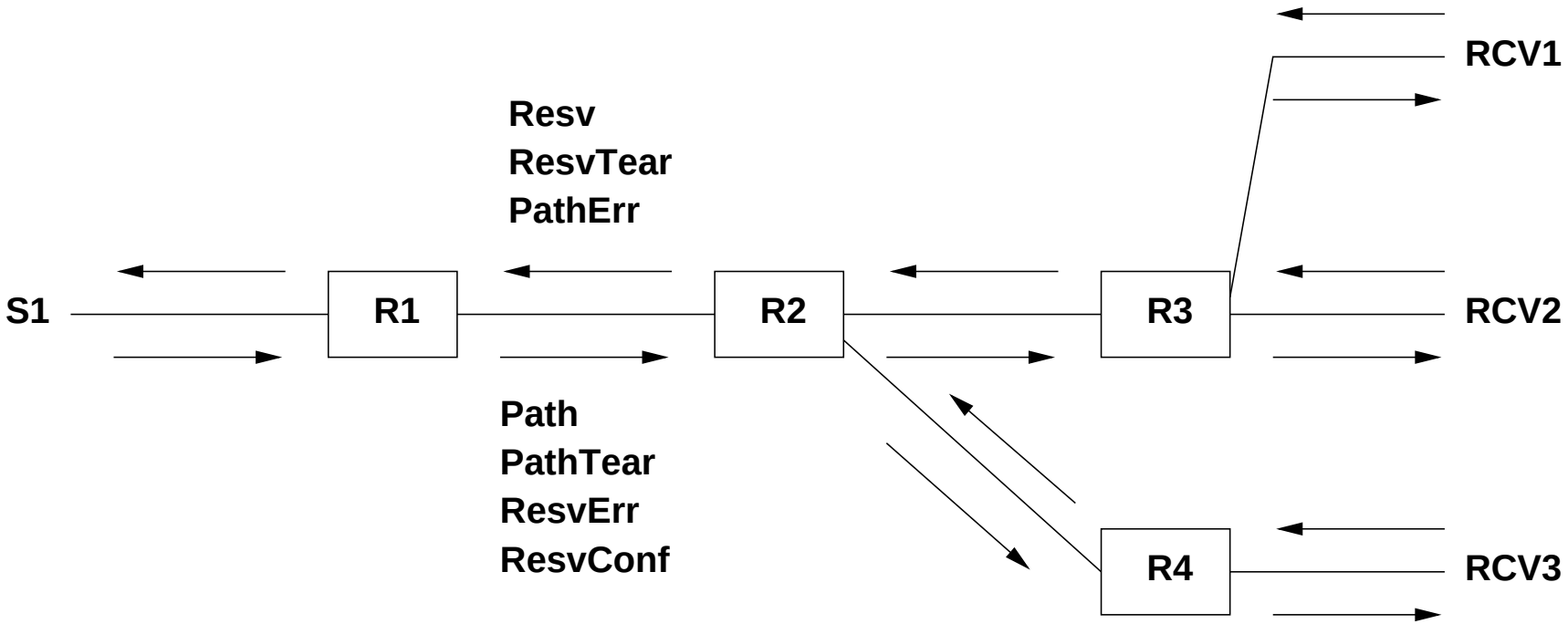
- The smallest common time slot is the time needed to transmit 100 bytes.
- The first stream needs to get half of the slots while the other two streams need to get 3 out of 12 slots each (note that packets in these two classes have a length of 3 slots).
- Thus, a virtual time sequence for the first stream could be $\{2, 4, 6, 8, \ldots\}$, for the second class $\{6.1, 18.1, 30.1, \ldots\}$, and for the third class $\{12.2, 24.2, 36.2, \ldots\}$.
- The resulting class sequence is $\{1, 1, 1, 2, 1, 1, 1, 3, \ldots\}$.

Integrated Services (RFC 1633)



- Motivation: Provide a framework for service guarantees to support applications which do not function with the Internet's best effort service model.
- Requires to introduce signalling and state in the core routing infrastructure.
- Policies are required to control the reservation and admission decisions.

Ressource Reservation Protocol (RSVP)



- QoS signalling protocol defined in RFC 2205.
- Flow-based Quality of Service (QoS).
- Routers may implement QoS by mapping to lower-layer QoS mechanisms.

RSVP Characteristics

- *Unicast and multicast*:
  RSVP is simplex, i.e., it makes reservations for unidirectional unicast or multicast data flows.
- *Soft state*:
  Reservation state is created and must be periodically refreshed. If routing changes, the RSVP state will timeout and new RSVP state will be installed on the new path.
- *Receiver initiated reservations*:
  The receiver of a data flow initiates and maintains the resource reservation used for that flow.
- *Policy control*:
  RSVP transports and maintains traffic control and policy control parameters that are opaque to RSVP.

Guaranteed Service (RFC 2212)

- The end-to-end delay bound is given by:

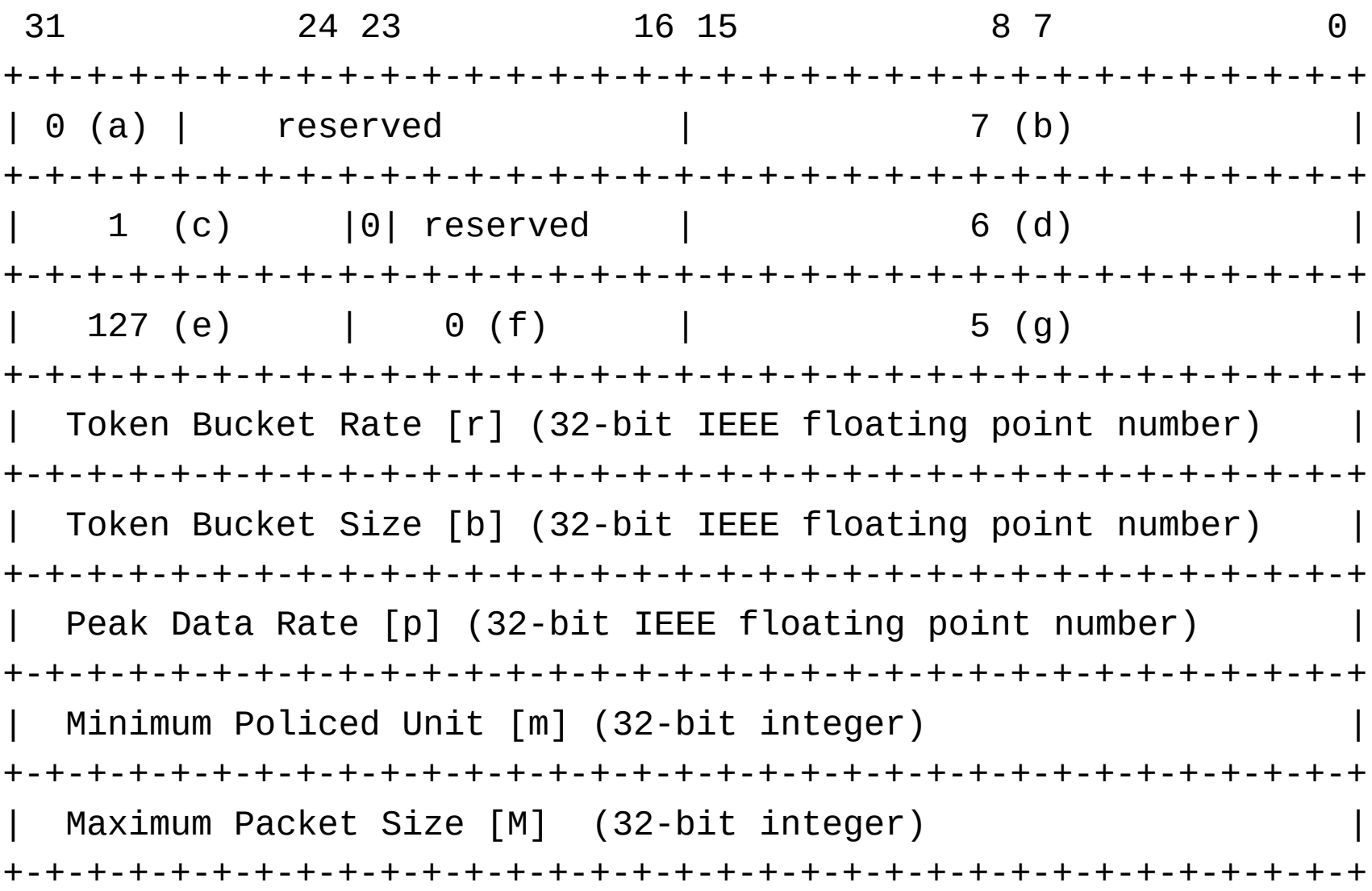
$$Q_{e2ed} = \begin{cases} \frac{b-M}{R}\frac{p-R}{p-r} + \frac{M+C_{tot}}{R+D_{tot}} & \text{for } p > R \geq r \\ \frac{M+C_{tot}}{R+D_{tot}} & \text{for } r \leq p \leq R \end{cases}$$

p	peak rate of flow (bytes/s)
b	bucket depth (bytes)
r	token bucket rate (bytes/s)
m	minimum policed unit (bytes)
M	maximum datagram size (bytes)
R	bandwidth (bytes/s)
S	slack term (s)
C_{tot}	cumulative sum of per hop error terms C
D_{tot}	cumulative sum of per hop error terms D

Controlled Load Service (RFC 2211)

- Controlled-load service provides the client data flow with a quality of service closely approximating the QoS that same flow would receive from an unloaded network element.
- Uses admission control to assure that this service is received even when the network element is overloaded.
- The important difference relative to best effort service is that controlled load service does not noticeable deteriorate as the network load increases.

Sender Traffic Specification (RFC 2210)



Sender Traffic Specification (RFC 2210)

(a) Message format version number (0)

(b) Overall length (7 words not including header)

(c) Service header, service number 1 (default/global information)

(d) Length of service 1 data, 6 words not including header

(e) Parameter ID, parameter 127 (Token_Bucket_TSpec)

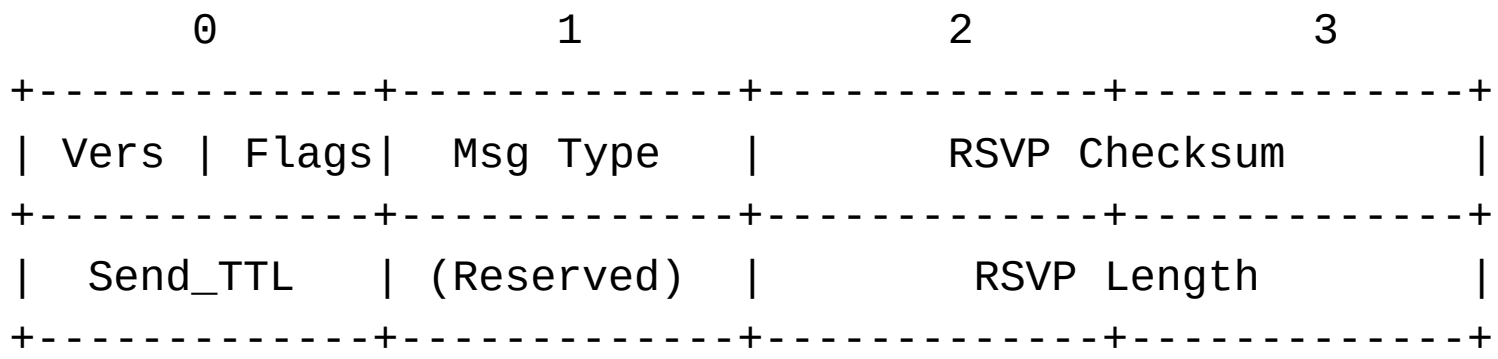
(f) Parameter 127 flags (none set)

(g) Parameter 127 length, 5 words not including header

RSVP Message Formats

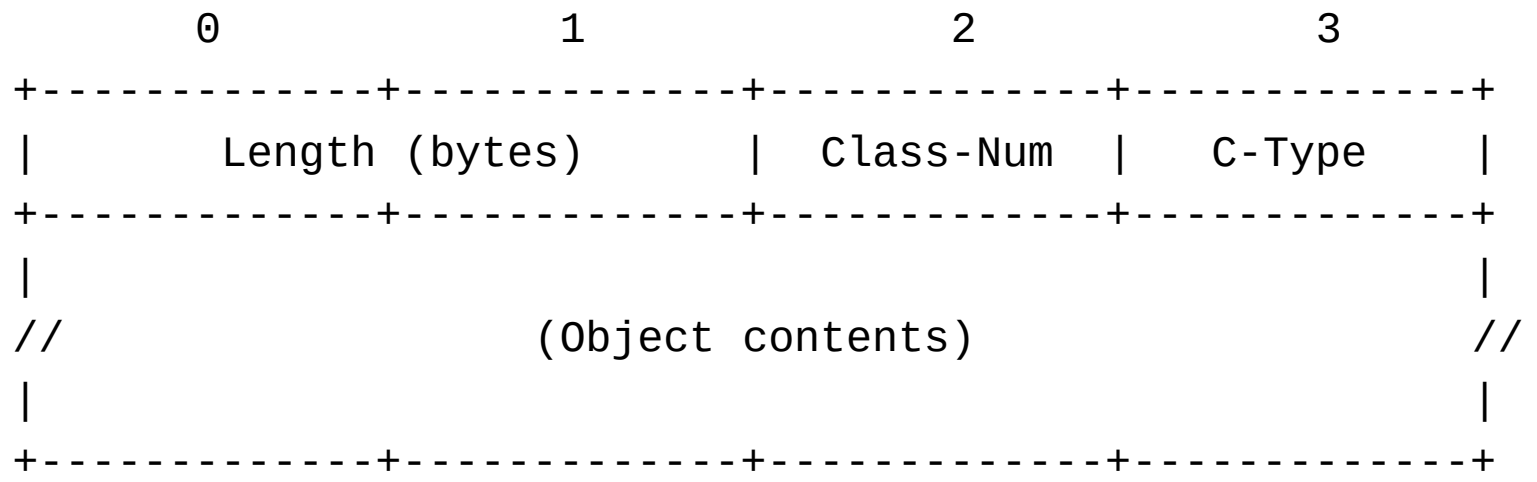
- An RSVP message consists of a common header.
- The body following the header consists of a variable number of variable-length, typed "objects".
- The permissible choice of object types is defined using an augmented Backus-Naur Form (BNF).

RSVP Common Header



- The `Vers` field contains the RSVP version (currently 1), the `Flags` field is currently unused, and the `Checksum` field contains the Internet checksum.
- The `Type` field identifies the RSVP message type and the `Length` field the overall length of the RSVP message.
- The `TTL` field contains the original TTL value.

RSVP Object Format



- Every object is encoded using one or more 32-bit words.
- The `Length` field contains the total length of an object.
- The `Class` field identifies the object's class while the `Type` field identifies the object's type within its class.

RSVP Messages

- `Path` message (periodically sent by sender):

```
<Path> ::= <Common Header> [ <INTEGRITY> ]
    <SESSION> <RSVP_HOP> <TIME_VALUES>
    [ <POLICY_DATA> ... ] [ <sender descriptor> ]

<sender descriptor> ::= <SENDER_TEMPLATE>
    <SENDER_TSPEC> [ <ADSPEC> ]
```

- `PathTear` message (sent by sender or router):

```
<PathTear> ::= <Common Header> [ <INTEGRITY> ]
    <SESSION> <RSVP_HOP> [ <sender descriptor> ]

<sender descriptor> ::= (see earlier definition)
```

RSVP Messages

- `Resv` message (sent by receivers)

```
<Resv> ::= <Common Header> [ <INTEGRITY> ]
    <SESSION>  <RSVP_HOP> <TIME_VALUES>
    [ <RESV_CONFIRM> ]  [ <SCOPE> ]
    [ <POLICY_DATA> ... ] <STYLE>
    <flow descriptor list>

<flow descriptor list> ::= <empty> |
    <flow descriptor list> <flow descriptor>
```

  - The content of the `flow descriptor` depends on the reservation style.

RSVP Messages

- `ResvTear` message (sent by receiver or router):

```
<ResvTear> ::= <Common Header> [<INTEGRITY>]
    <SESSION> <RSVP_HOP> [ <SCOPE> ] <STYLE>
    <flow descriptor list>

<flow descriptor list> ::= (see earlier definition)
```

- `ResvConf` message (sent by sender)

```
<ResvConf> ::= <Common Header> [ <INTEGRITY> ]
    <SESSION> <ERROR_SPEC> <RESV_CONFIRM>
    <STYLE> <flow descriptor list>

<flow descriptor list> ::= (see earlier definition)
```

RSVP Messages

- `PathErr` (sent by router):

```
<PathErr> ::= <Common Header> [ <INTEGRITY> ]
    <SESSION> <ERROR_SPEC>
    [ <POLICY_DATA> ...] [ <sender descriptor> ]

<sender descriptor> ::= (see earlier definition)
```

- `ResvErr` (sent by router):

```
<ResvErr> ::= <Common Header> [ <INTEGRITY> ]
    <SESSION>  <RSVP_HOP> <ERROR_SPEC>  [ <SCOPE> ]
    [ <POLICY_DATA> ...] <STYLE> [ <error flow descripto
```

  - The content of the `error flow descriptor` depends on the reservation style.

Integrated Services over IEEE 802

- Explains how RSVP reservations can be mapped to 802 link layer technologies.
- Defined in RFC 2815 and RFC 2816.

RSVP Critique

- RSVP requires that routers maintain state for every flow.
- Hence, RSVP does not scale with the number of flows.
- Are per-flow reservations practical in the Internet?
- RSVP only supports QoS signalling.
- Additional signalling needed (drilling holes into firewalls).
- IETF currently works on NSIS which builds upon RSVP and tries to provide a more general and lighter signalling mechanism.

Differentiated Services (RFC 2475)

- Goal: Scalability by aggregating traffic classification state which is conveyed by means of IP-layer packet marking.
- Packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path.
- Sophisticated classification, marking, policing, and shaping operations need only be implemented at network boundaries or hosts.
- Network resources are allocated to traffic streams by service provisioning policies which govern
  - how traffic is marked and conditioned upon entry to a differentiated services-capable network, and
  - how that traffic is forwarded within that network.

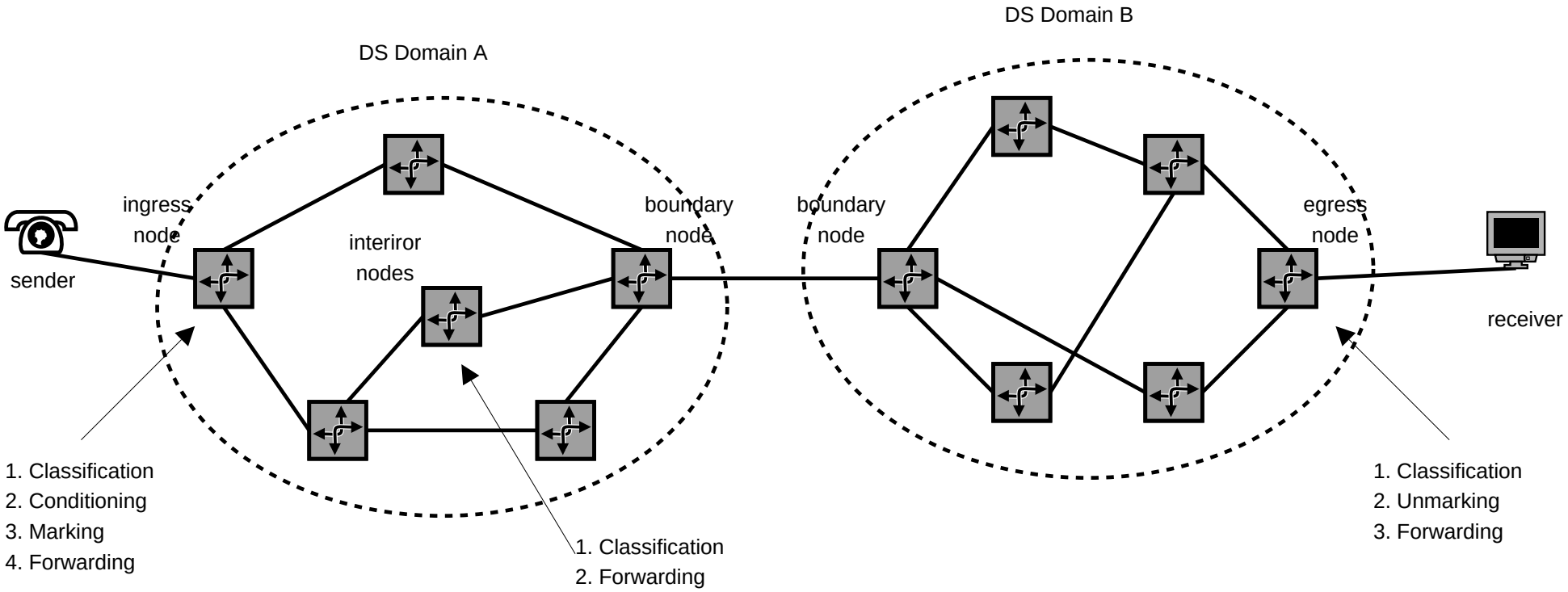
DS Code Point (RFC 2474)

- Two mechanisms are used to achieve scalability:
  1. Aggregation of related flows into service classes
  2. Reservations are provisioned for a longer period
- Packets are tagged when they enter a DiffServ domain using a 6-bit Differentiated Services Code Point (DSCP).
- The DSCP is a value carried in a DS field, which is either in
  - the Type of Service field of an IPv4 packet, or
  - the Traffic Class field of an IPv6 packet.
- Some of the 2^6 possible DSCP values have special usages (see RFC 2474).

Terminology

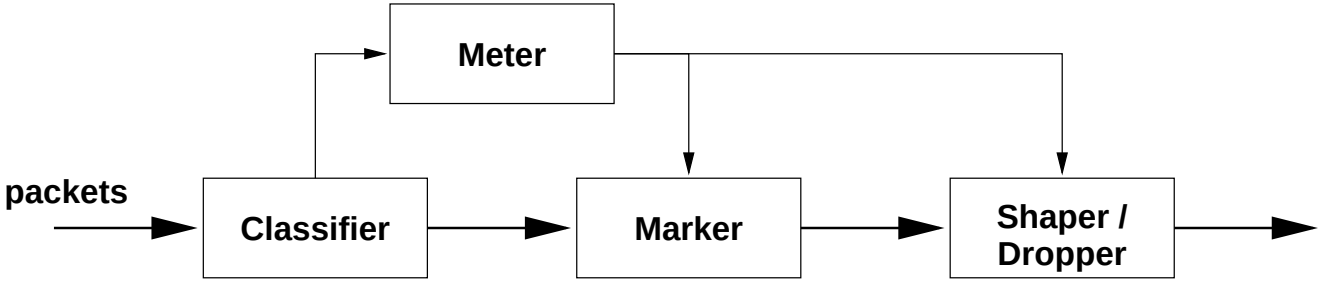
- *DS Domain*: a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions
- *DS Ingress Node*: a node handling traffic as it enters a DS domain
- *DS Egress Node*: a node handling traffic as it leaves a DS domain
- *DS Behavior Aggregate*: a collection of packets with the same DS codepoint crossing a link in a particular direction.
- *Per-Hop-Behavior (PHB)*: the externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate.

Differentiated Services Domains



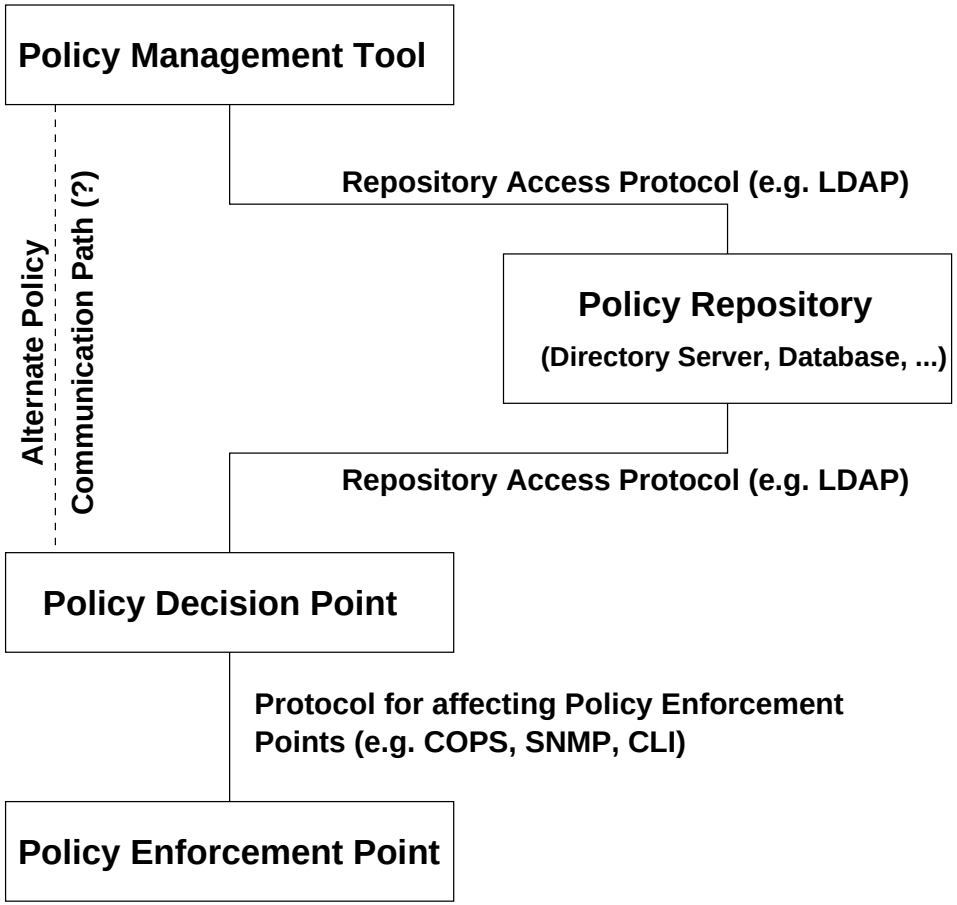
- Most of the complexity is moved to ingress nodes.
- Cooperating DS domains can form a DS region by establishing on Traffic Conditioning Agreements (TCAs).

Traffic Classifier and Conditioner



- *Classifier*: selects packets based on the content of packet headers according to defined rules.
- *Marker*: a device that sets DS codepoints in packets
- *Meter*: a device that performs metering
- *Shaper/Dropper*: a device that shaves and/or drops packets

Policy Management Framework

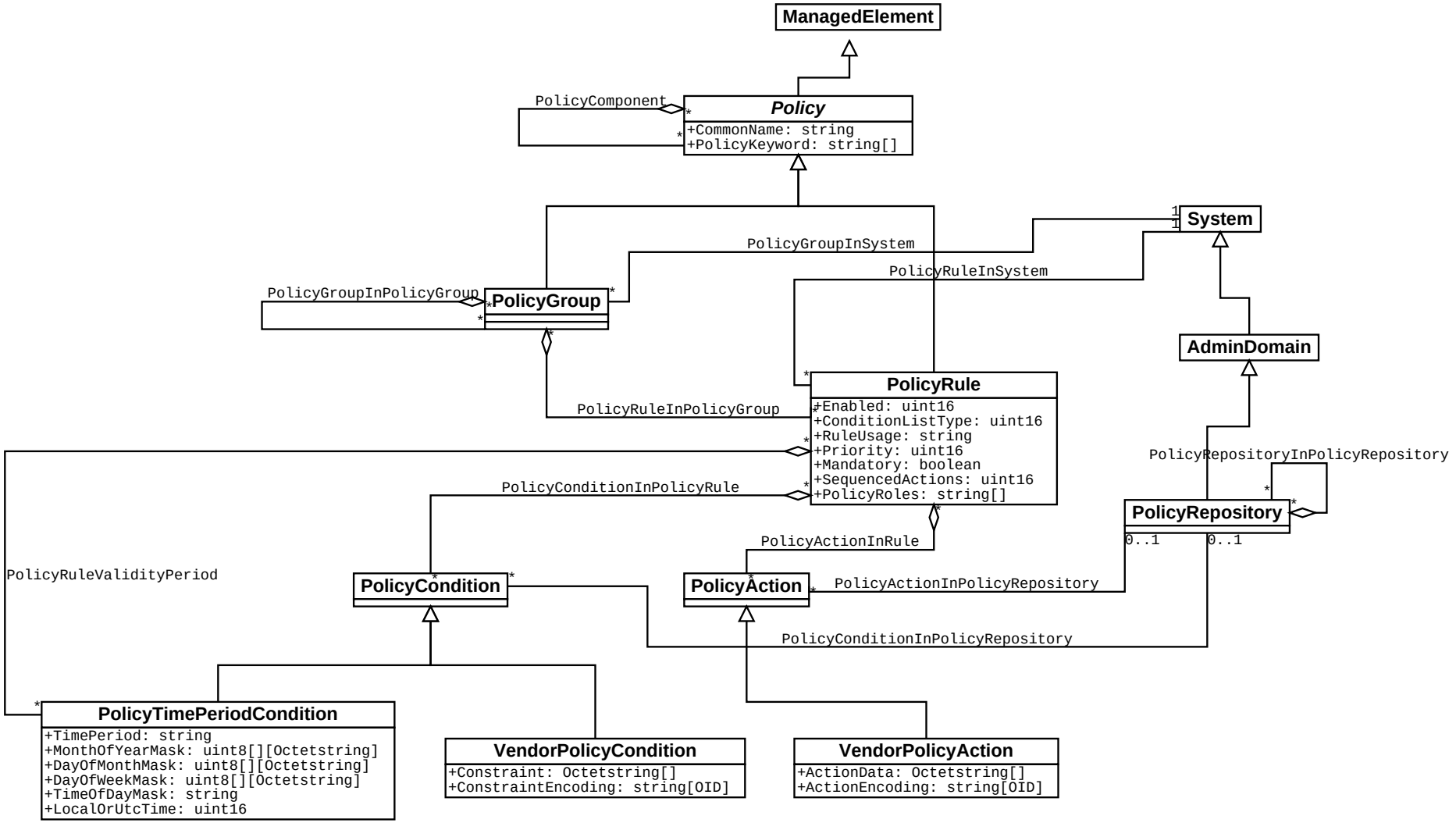


- Policy terminology is defined in RFC 3198.

Policy Core Information Model (PCIM)

- Policy Core Information Model (RFC 3460) is intended to serve as an extensible class hierarchy for defining policy objects represent policies of different types.
- Design of the Policy Core Information Model is influenced by a declarative, not procedural approach.
- Each policy rule consists of a set of conditions and a set of actions.
- Set of conditions associated with a policy rule can be in Disjunctive Normal Form (DNF) or Conjunctive Normal Form (CNF).
- For the set of actions associated with a policy rule, it is possible to specify an order of execution.
- Policy rules can be prioritized and aggregated into policy groups.

UML Diagram of the PCIM



PCIM Mapping to LDAP

- Structural classes:
  - Represent policy information and control of policies.
- Association classes:
  - Indicate how instances of the structural classes are related to each other.
- Mapping of structural classes:
  - PCIM classes are mapped to LDAP classes.
  - PCIM properties map to LDAP attributes.
- Mapping of association classes:
  - Partly mapped to LDAP
    - auxiliary classes,
    - attributes representing DN pointers,
    - containment in the DIT.

QoS PCIM Extensions

- Work continues on generic QoS extensions of the PCIM.
- Work continues on network device specific extensions of the QoS extension of PCIM.
- All the PCIM extensions are mapped (similar to PCIM itself) to LDAP schemas.

$\Rightarrow$ Strict top-down approach to define a policy class hierarchy.

$\Rightarrow$ Sometimes conflicts with the usual IETF way of working bottom-up.

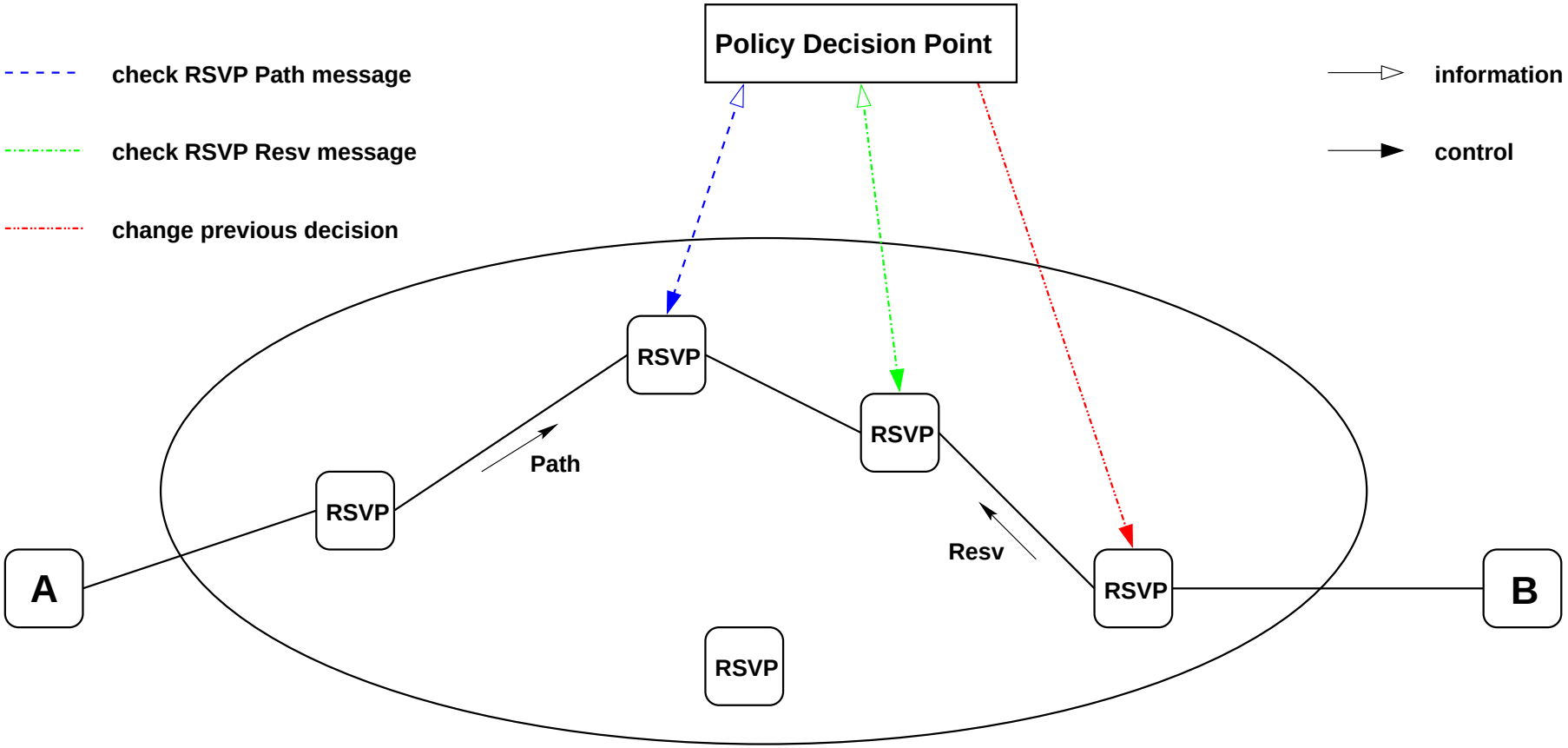
Common Open Policy Service (COPS)

- COPS (RFC 2748) is a client/server protocol between a policy decision point (PDP) and policy enforcement points (PEPs).
- Persistent TCP connections (well known port 3288).
- COPS messages contain sequences of COPS objects.
- Extensibility through self-identifying COPS objects.
- PDPs and PEPs can share state as long as the underlying TCP connection exists.
- PEP is responsible to establish a connection to its PDP.
- Optional message level security for authentication, replay protection, and message integrity through integrity objects.
- IPsec or TLS may be used for encryption.

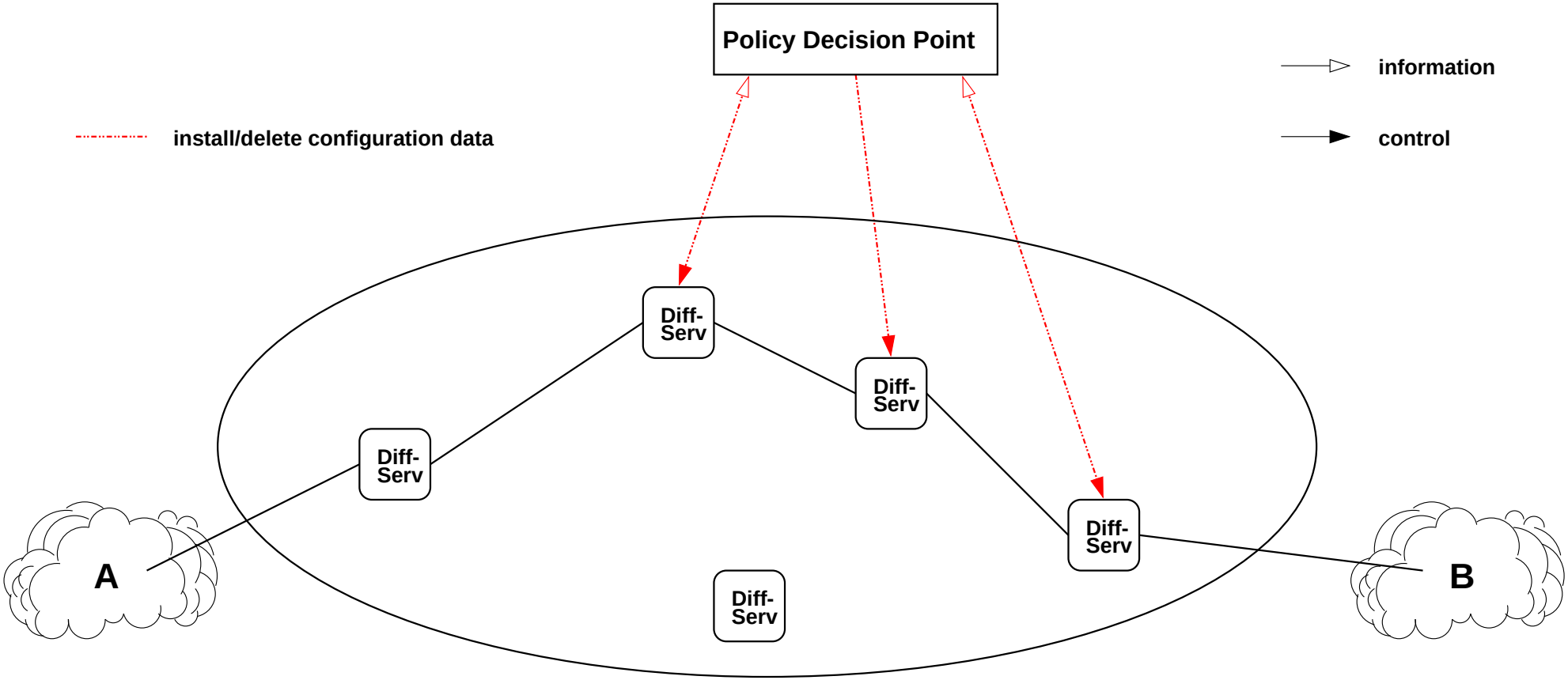
Outsourcing versus Provisioning

- Outsourcing Model:
  - Decisions are made event-driven during the signaling phase.
  - Additional asynchronous decisions from the policy-based management system.
  - Applicable where scalability is not a big issue.
- Provisioning Model:
  - Provisioning of all necessary configuration information to enforce policies locally.
  - Provisioning information is defined in a Policy Information Base (PIB).
  - PIBs are defined using the Structure of Policy Provisioning Information (SPPI).
  - No real-time policy interactions, highly scalable.

Outsourcing Model (RSVP)



Provisioning Model (DiffServ)



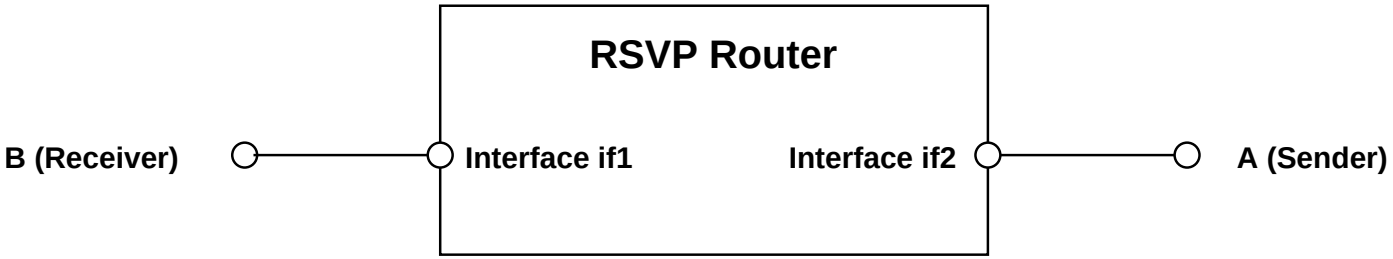
COPS Operation Types (RFC 2748)

Operation	Description	Direction
REQ	Request	PEP $\rightarrow$ PDP
DEC	Decision	PDP $\rightarrow$ PEP
RPT	Report State	PEP $\rightarrow$ PDP
DRQ	Delete Request State	PEP $\rightarrow$ PDP
SSQ	Synchronize State Request	PDP $\rightarrow$ PEP
SSC	Synchronize State Complete	PEP $\rightarrow$ PDP
OPN	Client-Open	PEP $\rightarrow$ PDP
CAT	Client-Accept	PDP $\rightarrow$ PEP
CC	Client-Close	PEP $\rightarrow$ PDP, PDP $\rightarrow$ PEP
KA	Keep-Alive	PEP $\rightarrow$ PDP, PDP $\rightarrow$ PEP

COPS for RSVP (RFC 2749)

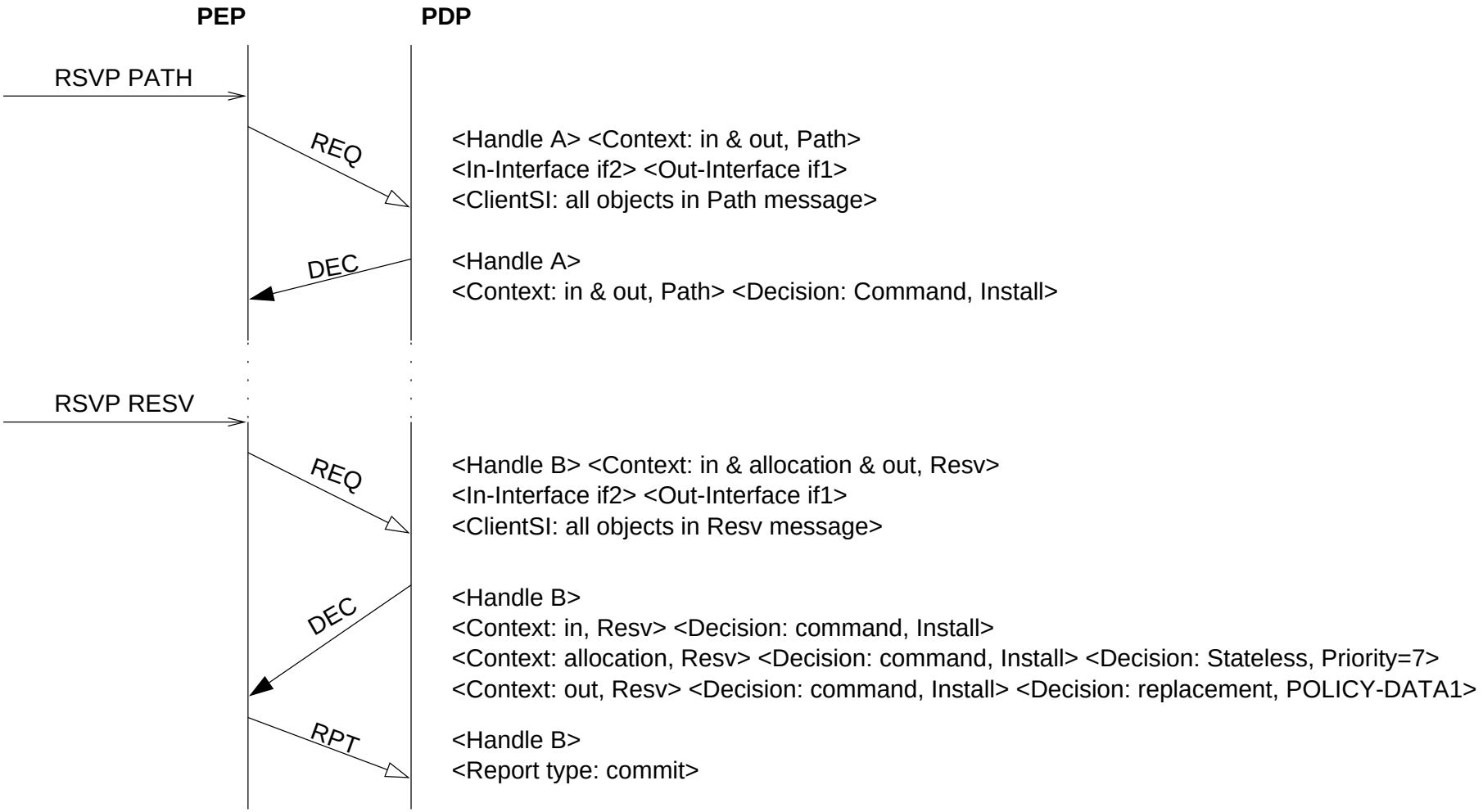
- All objects received in an RSVP message are encapsulated inside the COPS Client Specific Information Object (ClientSI) send from the PEP to the PDP.
- The PEP and PDP share RSVP state.
- Install decision command:
  - Accept/Allow/Admit an RSVP message or local resource allocation.
- Remove decision command:
  - Deny/Reject/Remove an RSVP message or local resource allocation.
- PEP may cache decisions in order to use them for a given time interval while the connection between the PEP and its PDP is lost.

COPS-RSVP Protocol Example

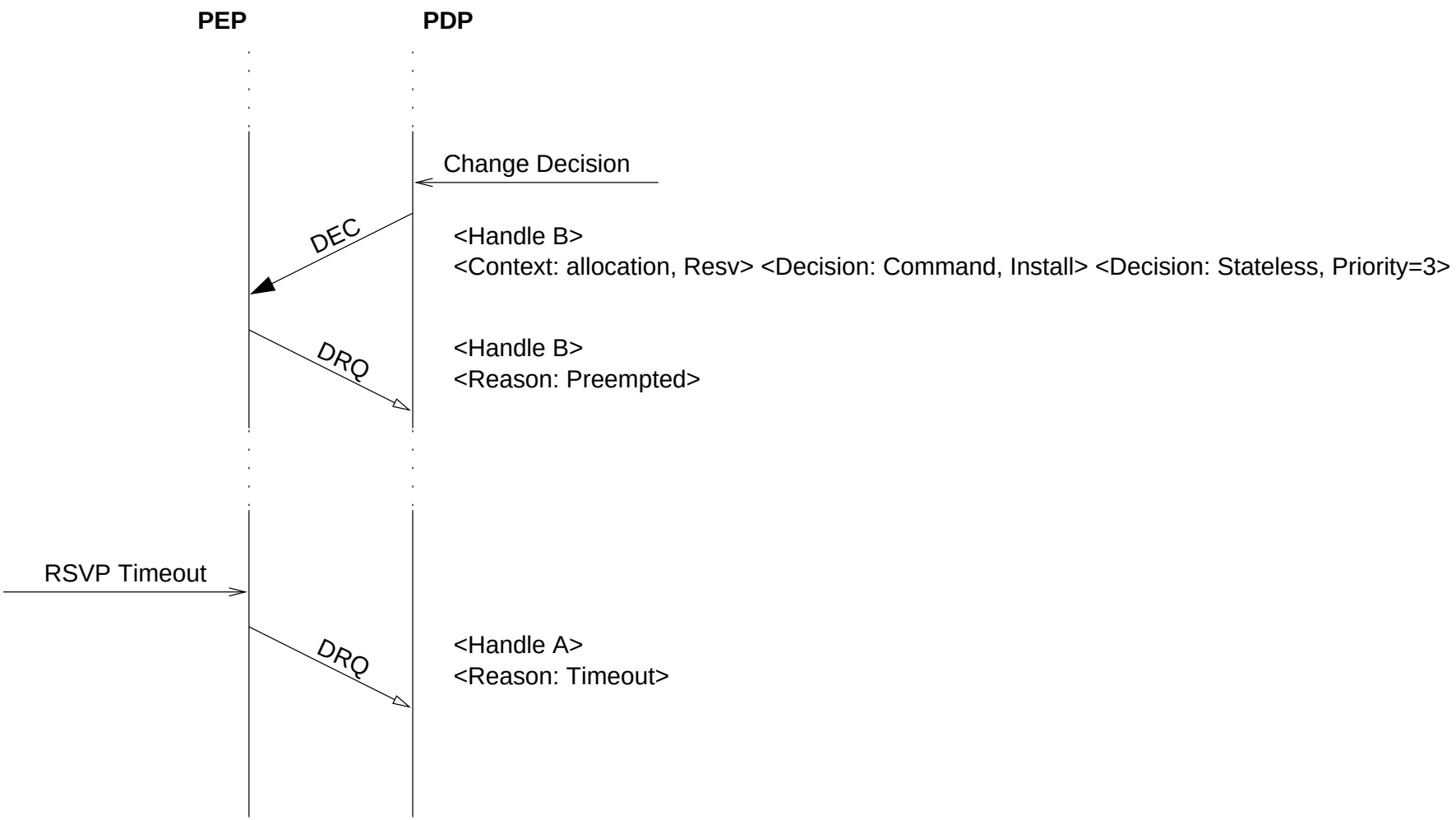


- The PEP router has two network interfaces (if1, if2).
- Sender A sends to receiver B.
- COPS RSVP is used to control the unicast RSVP flow between A and B.

COPS-RSVP Protocol Example



COPS-RSVP Protocol Example



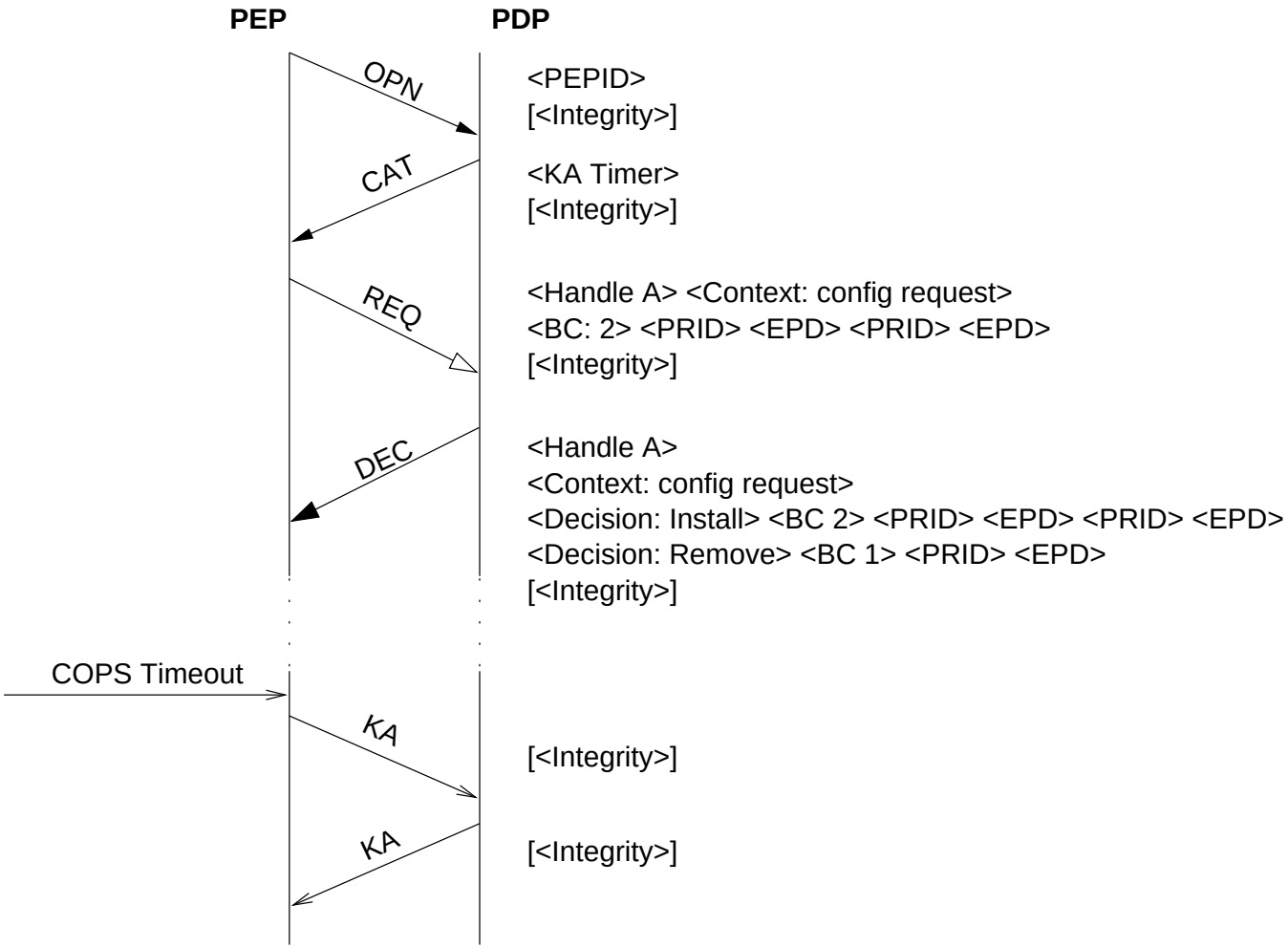
COPS for Provisioning

- Provisioning of policy configurations (a set of policy class instances) at PEPs.
- COPS-PR specific terminology:
  - Policy Rule Class (PRC):
    An ordered set of attributes. PRCs are defined in PIB modules and registered in the Object Identifier tree.
  - Policy Rule Instance (PRI):
    An instantiation of a PRC.
  - Policy Rule Instance Identifier (PRID):
    A positive integer which identifies a PRI of a given PRC.
  - Encoded Policy Instance Data (EPD):
    BER encoded representation of a PRI.

Policy Information Base (PIB)

- A Policy Information Base (PIB) defines a set PRCs for use with COPS-PR.
- PIB modules are written using the Structure of Policy Provisioning Information (SPPI).
- The SPPI is an adapted superset of the SNMP's SMIv2.
- Hooks in the SPPI can be used to generate MIBs from PIBs for usage with SNMP.

COPS-PR Protocol Example



References

[1] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633, ISI, MIT, Xerox PARC, June 1994.

[2] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification. RFC 2205, ISI, UCLA, IBM Research, Univ. of Michigan, September 1997.

[3] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC 2210, MIT LCS, September 1997.

[4] J. Wroclawski. Specification of the Controlled-Load Network Element Service. RFC 2211, MIT LCS, September 1997.

[5] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. RFC 2212, Xerox, BBN, IBM, September 1997.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475, Torrent Networking Technologies, EMC Corporation, Sun Microsystems, Nortel UK, Bell Labs Lucent Technologies, Lucent Technologies, December 1998.

[7] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474, Cisco Systems, Torrent Networking Technologies, EMC Corporation, December 1998.

[8] Y. Bernet, S. Blake, D. Grossman, and A. Smith. An Informal Management Model for Diffserv Routers. RFC 3290, Microsoft, Ericsson, Motorola, Harbour Networks, May 2002.

3. Multimedia Transport and Signaling

Overview

- Transport of multimedia streams (RTP)
- Description of multimedia sessions (SDP)
- Signalling of multimedia sessions (SIP)

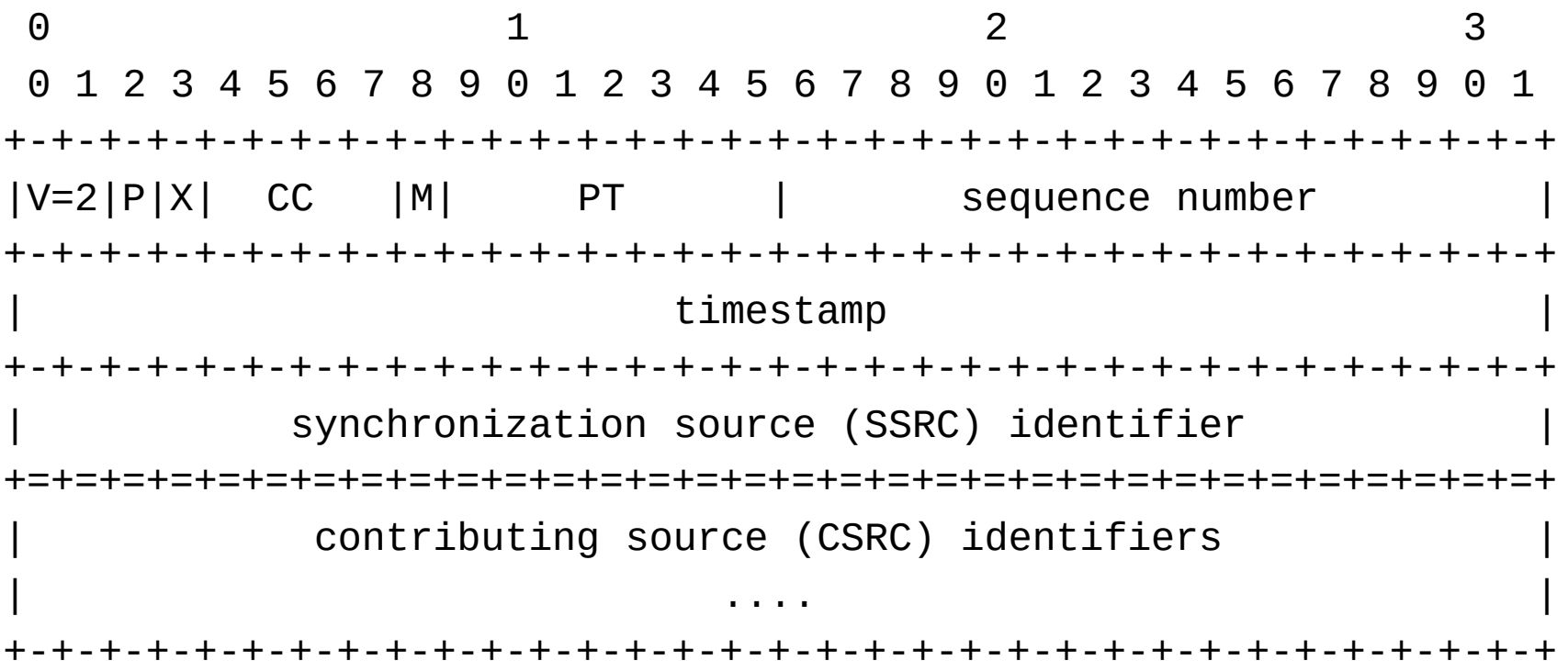
Real-time Transport Protocol (RTP)

- RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services.
- RTP and RTCP are designed to be independent of the underlying transport and network layers (but commonly used over UDP).
- The protocol supports the use of RTP-level translators and mixers.

RTP Elements

- *Synchronisation Source*: A source identified by a 32-bit number which is originating data streams.
- *Mixer*: A component capable of resynchronizing streams, mixing reconstructed streams into a single stream, or translating the encoding of a stream (e.g., from a high-quality encoding to a low-bandwidth encoding).
- *Translator*: A component capable to translate streams in order to cross firewalls or different transports.
- *Receiver*: A component resynchronizing received streams, usually using playout buffers.

RTP Message Format



- RTP is defined in RFC 3550.
- RTP supports multicasting of multimedia streams since it is running over UDP

RTP Message Header

- The `V` fied contains the RTP version number (current version is 2).
- The `X` bit indicates whether there are any extension headers
- The `P` bit indicates that there are padding bytes at the end of the packet. (The last padding byte contains the number of padding bytes.)
- The `CC` field contains the number of CSRC identifiers.
- The `M` bit may be used by profiles that mark certain bytes in the packets.
- The `PT` identifies the format of the RTP payload.

RTP Message Header

- The `sequence number` field contains a sequence number for each packet.
- The `timestamp` field is use indicate the relative time of the packet in the overall media stream (media timestamp).
- The `SSRC` field identifies the synchronization source.
- The `CSRC` identifiers (if present) identify the synchronization sources in cases where multiple media streams have been mixed into a single stream.
- RTP profiles define how RTP is used to transport specific codecs.

RTP Control Protocol (RTCP)

- RTCP allows senders and receivers to transmit a series of reports to one another that contain additional information about
  - the data being transmitted and
  - the performance of the network.
- RTCP packet types:
  - SR: Sender report, for transmission and reception statistics from participants that are active senders
  - RR: Receiver report, for reception statistics from participants that are not active senders
  - SDES: Source description items, including CNAME
  - BYE: Indicates end of participation
  - APP: Application-specific functions

RTCP Extended Reports (XR)

- XR packets convey information beyond that already contained in the reception report blocks of RTCP's sender report (SR) or Receiver Report (RR) packets.
- Packet-by-packet report blocks:
  - *Loss RLE Report Block*: Run length encoding of reports concerning the losses and receipts of RTP packets.
  - *Duplicate RLE Report Block*: Run length encoding of reports concerning duplicates of received RTP packets.
  - *Packet Receipt Times Report Block*: A list of reception timestamps of RTP packets.

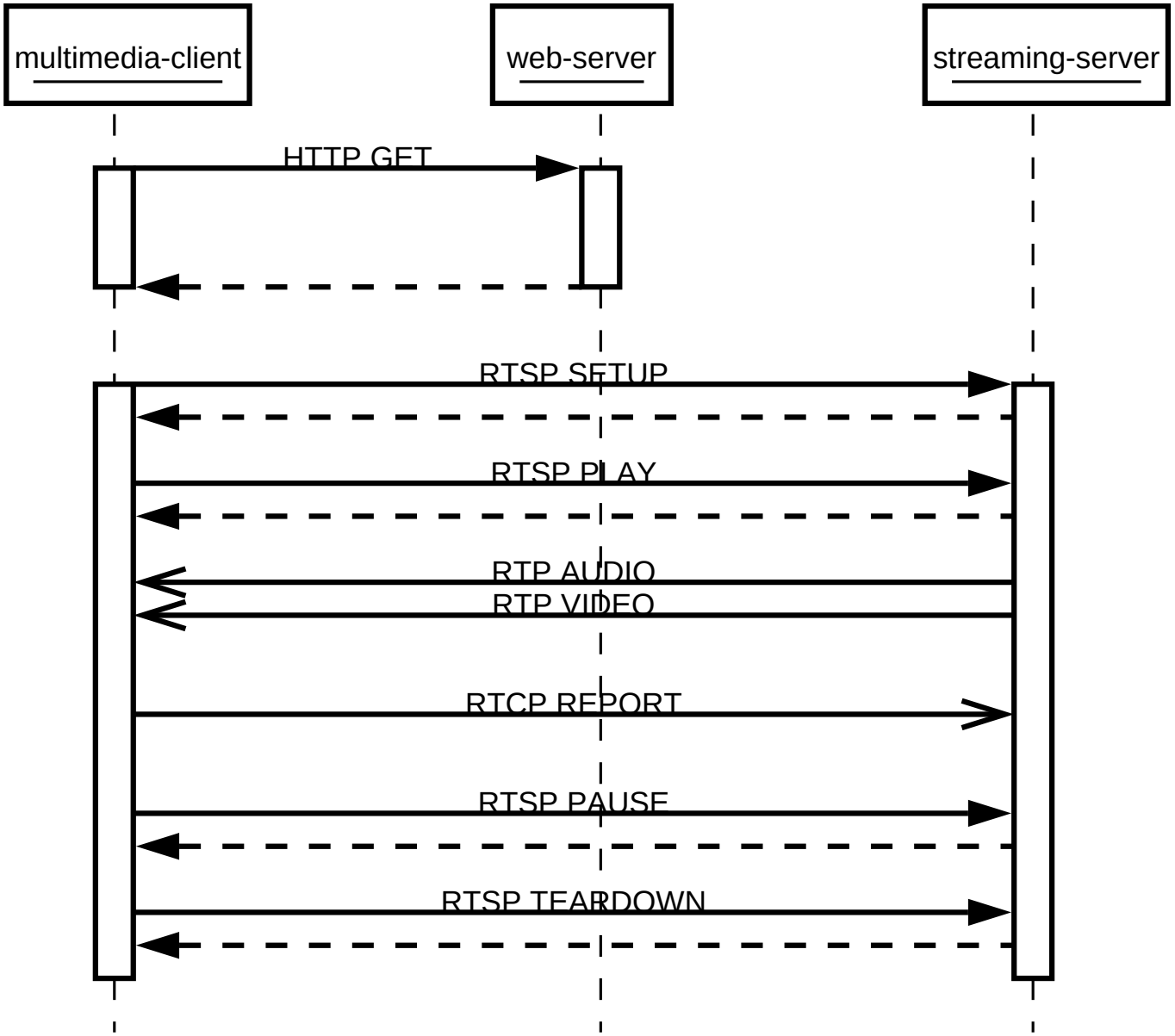
RTCP Extended Reports (XR)

- Reference time report blocks:
  - *Receiver Reference Time Report Block*: Receiver-end wallclock timestamps. Together with the DLRR Report Block mentioned next, these allow non-senders to calculate round-trip times.
  - *DLRR Report Block*: The delay since the last Receiver Reference Time Report Block was received.
- Metric report blocks:
  - *Statistics Summary Report Block*: Statistics on RTP packet sequence numbers, losses, duplicates, jitter, and TTL or Hop Limit values.
  - *VoIP Metrics Report Block*: Metrics for monitoring Voice over IP (VoIP) calls.

Real-Time Streaming Protocol (RTSP)

- The Real-Time Streaming Protocol (RTSP) defined in RFC 2326 establishes and controls either a single or multiple time-synchronized streams of continuous media.
- RTSP acts as a “network remote control” for multimedia servers.
- The RTSP protocol is similar in syntax and operation to HTTP version 1.1.
- RTSP, however, differs fundamentally from HTTP in that data delivery takes place out-of-band in a different protocol.
- While RTSP was writting to support RTP, it is not tied to RTP as the real-time media transport protocol.

RTP + RTCP + RTSP + HTTP



RSTP Options

- `OPTIONS` get available methods
- `SETUP` establish transport
- `ANNOUNCE` change description of media object
- `DESCRIBE` get (low-level) description of media object
- `PLAY` start playback, reposition
- `RECORD` start recording
- `REDIRECT` redirect client to new server
- `PAUSE` halt delivery, but keep state
- `SET PARAMETER` device or encoding control
- `GET PARAMETER` device or encoding control
- `TEARDOWN` remove state

Session Description Protocol (SDP)

- SDP as defined in RFC 4566 is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation.
- SDP description format is used by other protocols (such as RSTP).
- A session description includes:
  - Session name and purpose
  - Time(s) the session is active
  - The media comprising the session
  - Information to receive those media (addresses, ports, formats and so on)

Sample Session Description

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

- Description of a session called "SDP Seminar" which is sent to the multicast group 224.2.17.12 and contains three channels (audio, video, whiteboard).
- Start and stop times are indicated in the `t=` field.

Session Initiation Protocol (SIP)

- An application layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants.
- Sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.
- SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users.
- SIP runs on top of several different transport protocols.
- SIP is defined in RFC 3261.

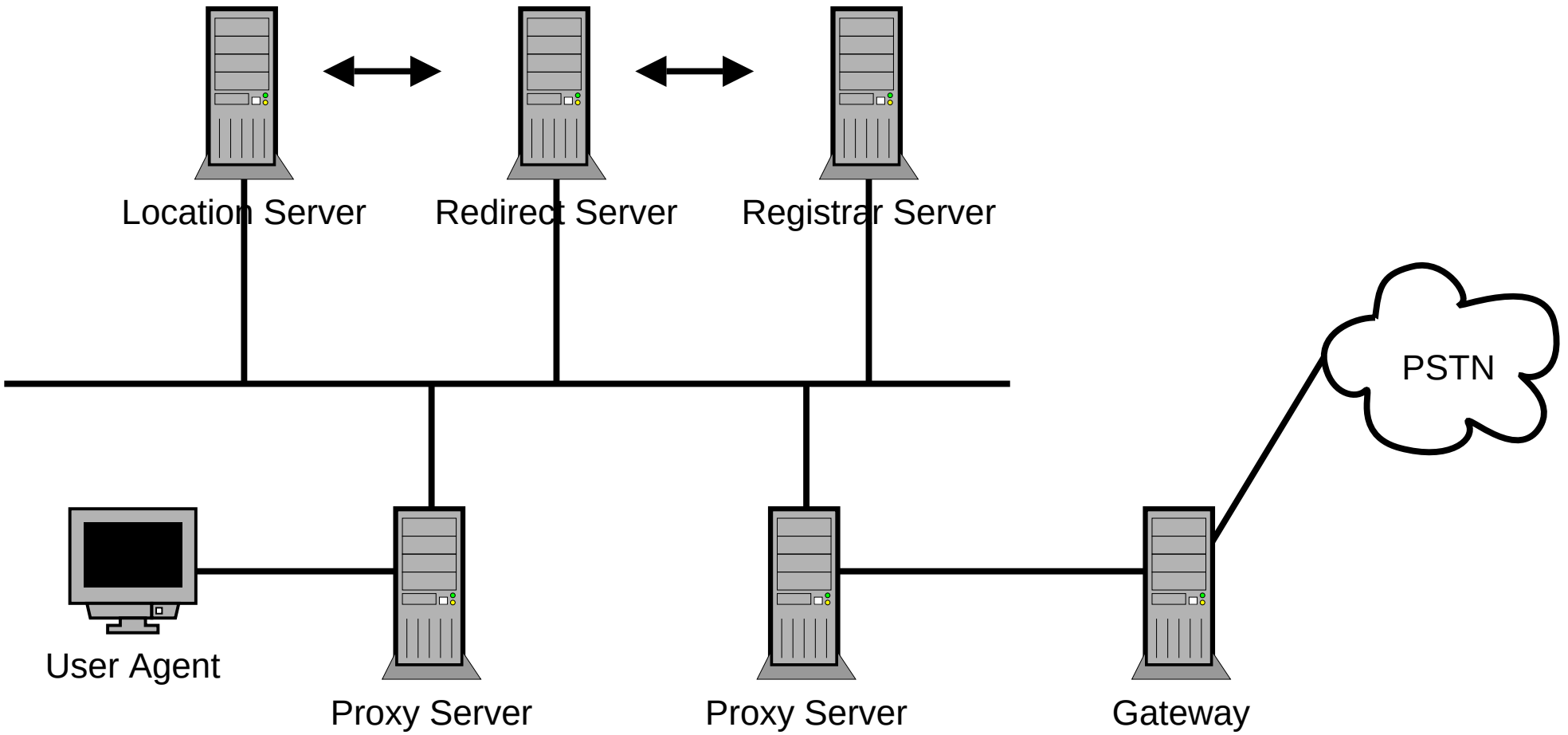
SIP Facets

- *User location*: determination of the end system to be used for communication
- *User availability*: determination of the willingness of the called party to engage in communications
- *User capabilities*: determination of the media and media parameters to be used
- *Session setup*: "ringing", establishment of session parameters at both called and calling party
- *Session management*: including transfer and termination of sessions, modifying session parameters, and invoking services

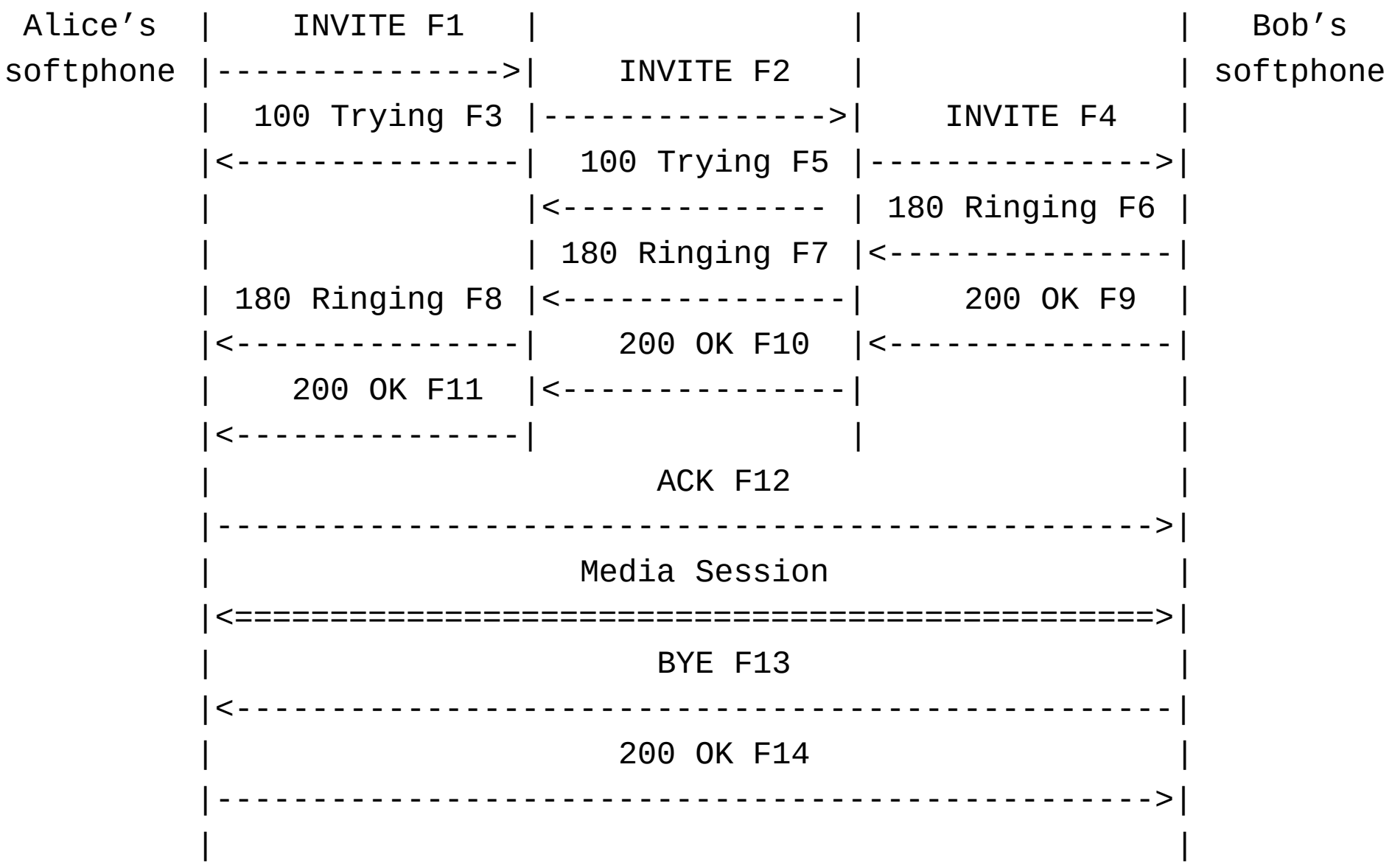
SIP Properties

- HTTP-like textual message format
- HTTP-like method calls and status responses
- Utilizes the Session Description Protocol (SDP) for the description of multimedia sessions
- User agents can initiate and receive calls (session endpoints)
- Proxy server provide an infrastructure to help user agents to establish sessions
- . . .

SIP Interworking



SIP Session Setup Example



Alice's INVITE Message

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
```

Bob's 200 Response

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
   ;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
   ;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com
   ;branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131

(Bob's SDP not shown)
```

SIP URIs

`sip:user:password@host:port;uri-parameters?headers`

- The `user` token identifies a particular resource at the host being addressed.
- The `password` token is a password associated with `user` (usage not recommended).
- The `host` token identifies the host providing SIP resources.
- The `port` number is the port to which a request is to be sent.
- The `uri-parameters` affect the request constructed from the URI.
- The `headers` token specifies the header fields to be included in a request constructed from a URI.

SIP URI Examples

- Typical SIP URI for user alice:

```
sip:alice@atlanta.com
```

- The same with an explicit IP address:

```
sip:alice@192.0.2.4
```

- The same with a password and an explicti transport:

```
sip:alice:secretword@atlanta.com;transport=tcp
```

- SIP URI with an embedded PSTN phone number:

```
sip:+1-212-555-1212:1234@gateway.com;user=phone
```

- SIP URI with an explicit method call:

```
sip:atlanta.com;method=REGISTER?to=alice%40atlanta.com
```

Locating SIP Servers (RFC 3263)

- Given a SIP URI, how do you find the responsible SIP server?
- Need to determine
  - the transport protocol and
  - the IP address and port number

  of the SIP server or proxy.
- In the general case, it is preferrable to have SIP URIs that belong to a domain rather than a specific host:

  `sip:j.schoenwaelder@iu-bremen.de`
- A mechanism similar to `MX` records for email is needed.
- Could there be a generalized solution?

Transport Selection

```
if <SIP URI specifies transport> {
    <use specified transport>
} else {
    if <host part of the URI is numeric> {
        <use udp for sip: and tcp for sips:>
    } else {
        <lookup a NAPTR DNS record using transport
         selection fields SIP+D2U, SIP+D2T, SIP+D2S>
        if <no NAPTR records available> {
            <construct and perform SRV queries>
        }
        if <still not successfull> {
            <use udp for sip: and tcp for sips:>
        }
    }
}
```

Port and IP Address Selection

```
if <host part of the URI is numeric> {
    <use IP address>
    if <port part of the URI is not empty> {
        <use the port number contained in the URI>
    } else {
        <use the default port number>
    }
} else {
    if <port part of the URI is not empty> {
        <lookup IP addresses using A or AAAA queries>
        <try the addresses with the port number until success>
    } else {
        <lookup a SRV record for the determined transport>
        if <no SRV record available> {
            <lookup IP addresses using A or AAAA queries>
            <try the addresses with the port number until success>
        } else {
            <try the locations specified by the SRV record until success>
        }
    }
}
```

DNS NAPTR and SRV Resource Records

```
;          order pref flags service      regexp  replacement
   IN NAPTR 50   50  "s"  "SIPS+D2T"     ""  _sips._tcp.example.com.
   IN NAPTR 90   50  "s"  "SIP+D2T"      ""  _sip._tcp.example.com.
   IN NAPTR 100  50  "s"  "SIP+D2U"      ""  _sip._udp.example.com.
;           Priority Weight Port   Target
    IN SRV  0        1      5060   server1.example.com
    IN SRV  0        2      5060   server2.example.com
```

- NAPTR records provide a mapping from a domain to the SRV record for contacting a server with the specific transport protocol in the NAPTR services field (RFC 2915).
- SRV records specify the location of server(s) for a specific protocol and domain (RFC 2782).

SIP Methods

- `INVITE`
  - Invites a user to participate in a session (call).
- `ACK`
  - Confirms final response to an `INVITE` request.
- `OPTIONS`
  - Used to query the capabilities of a server.
- `BYE`
  - Indicates termination of a call.
- `CANCEL`
  - Cancels a pending request.
- `REGISTER`
  - Registers a user agent at a proxy.

SIP Status Codes

1xx Provisional – request received, continuing to process the request

2xx Success – the action was successfully received, understood, and accepted

3xx Redirection – further action needs to be taken in order to complete the request

4xx Client Error – the request contains bad syntax or cannot be fulfilled at this server

5xx Server Error – the server failed to fulfill an apparently valid request

6xx Global Failure – the request cannot be fulfilled at any server

SIP Communication Establishment

- Communication establishment is done in six steps:
  1. Registering, initiating and locating the user.
  2. Determine the media to use – involves delivering a description of the session that the user is invited to.
  3. Determine the willingness of the called party to communicate.
  4. Call setup.
  5. Call modification of handling – example, call transfer (opional)
  6. Call termination

SIP Registration

- During startup, a SIP user agent registers with its proxy/registration server.
- Registration can also occur when the SIP user agent moves and the new location needs to be communicated.
- The registration information is periodically refreshed (each SIP user agent has to re-register).
- Typically, the proxy/registration server will forward the location information to the location/redirect server.
- In many cases, the different servers might be co-located.

ENUM and DDDS (RFC 3761)

- ENUM defined in RFC 3761 provides a mechanism to lookup information associated with telephone numbers in the DNS.
- Number conversion:
  1. Remove all characters with the exception of the digits. Example: "+442079460148" -> "442079460148"
  2. Put dots (".") between each digit. Example: 4.4.2.0.7.9.4.6.0.1.4.8
  3. Reverse the order of the digits. Example: 8.4.1.0.6.4.9.7.0.2.4.4
  4. Append the string ".e164.arpa" to the end. Example: 8.4.1.0.6.4.9.7.0.2.4.4.e164.arpa
- Lookup a NAPTR record for the resulting DNS name.

DDDS

- The Dynamic Delegation Discovery System (DDDS) defined in RFC 3403 is used to implement lazy binding of strings to data, in order to support dynamically configured delegation systems.
- The DDDS functions by mapping some unique string to data stored within a DDDS Database by iteratively applying string transformation rules until a terminal condition is reached.
- The core of DDDS are NAPTR records.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, Columbia University, Packet Design, Blue Coat Systems Inc., Packet Design, July 2003.

[2] A. Clark T. Friedman, R. Caceres. RTP Control Protocol Extended Reports (RTCP XR). RFC 3611, Paris 6, IBM Research, A. Clark, November 2003.

[3] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. RFC 4566, UCL, Packet Design, University of Glasgow, July 2006.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, dynamicsoft, Columbia University, Ericsson, WorldCom, Neustar, ICIR, ATT, June 2002.

[5] J. Rosenberg and H. Schulzrinne. Session Initiation Protocol (SIP): Locating SIP Servers. RFC 3263, dynamicsoft, Columbia University, June 2002.

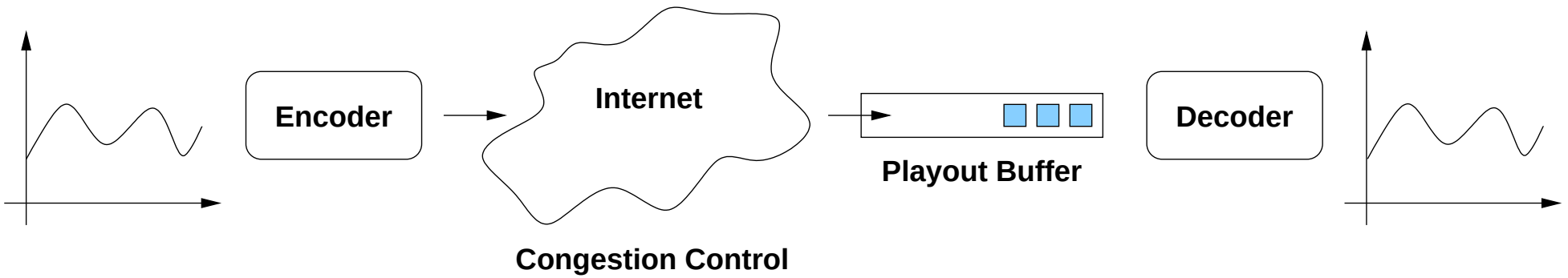
[6] H. Schulzrinne and J. Rosenberg. The Session Initiation Protocol: Internet-Centric Signaling. *IEEE Communications Magazine*, 38(10), October 2000.

4. Voice over IP

Voice over IP

- Idea: Send voice over the packet switched Internet
  - Digitizing and encoding voice signals
  - Transmission over the Internet
  - Decoding and generating the analog voice signal
- RTP/RTCP (UDP/IP) can be used for the transmission
- Need a common signalling protocol (ringing the phone)
- Need an infrastructure to locate users and phones

Voice over IP



- Voice quality is impacted by
  - encoding of the digitized analog signal
  - transmission impairments (delay, jitter, loss)
- Playout buffers can mitigate some of the effects
  - Playout buffers should be adaptive
  - For bidirectional voice conversations, there is an upper limit of delay

Pulse Code Modulation (PCM)

- Voice bandwidth is 4 kHz, so sampling bandwidth has to be 8 kHz (for Nyquist).
- Represent each sample with 8 bit (having 256 possible values).
- Throughput is 8000 Hz * 8 bit = 64 kbit/s, as a typical digital phone line.
- In real applications mu-law (North America) and a-law (Europe) variants are used which code the analog signal on a logarithmic scale using 12 or 13 bits instead of 8 bits (see Standard ITU-T G.711).

Other Codecs

- Adaptive differential PCM (ADPCM), ITU-T G.726
  - Encode the difference between the actual and the previous voice packet, requiring 32 kbps.
- LD-CELP, Standard ITU-T G.728
- CS-ACELP, Standard ITU-T G.729 and G.729a
- MP-MLQ, Standard ITU-T G.723.1, 6.3kbps, Truespeech
- ACELP, Standard ITU-T G.723.1, 5.3kbps, Truespeech
- LPC-10, able to reach 2.5 kbps
- iLBC, low bit-rate, able to deal with packet loss
- speex, free codec, 8/16/32 kHZ sampling

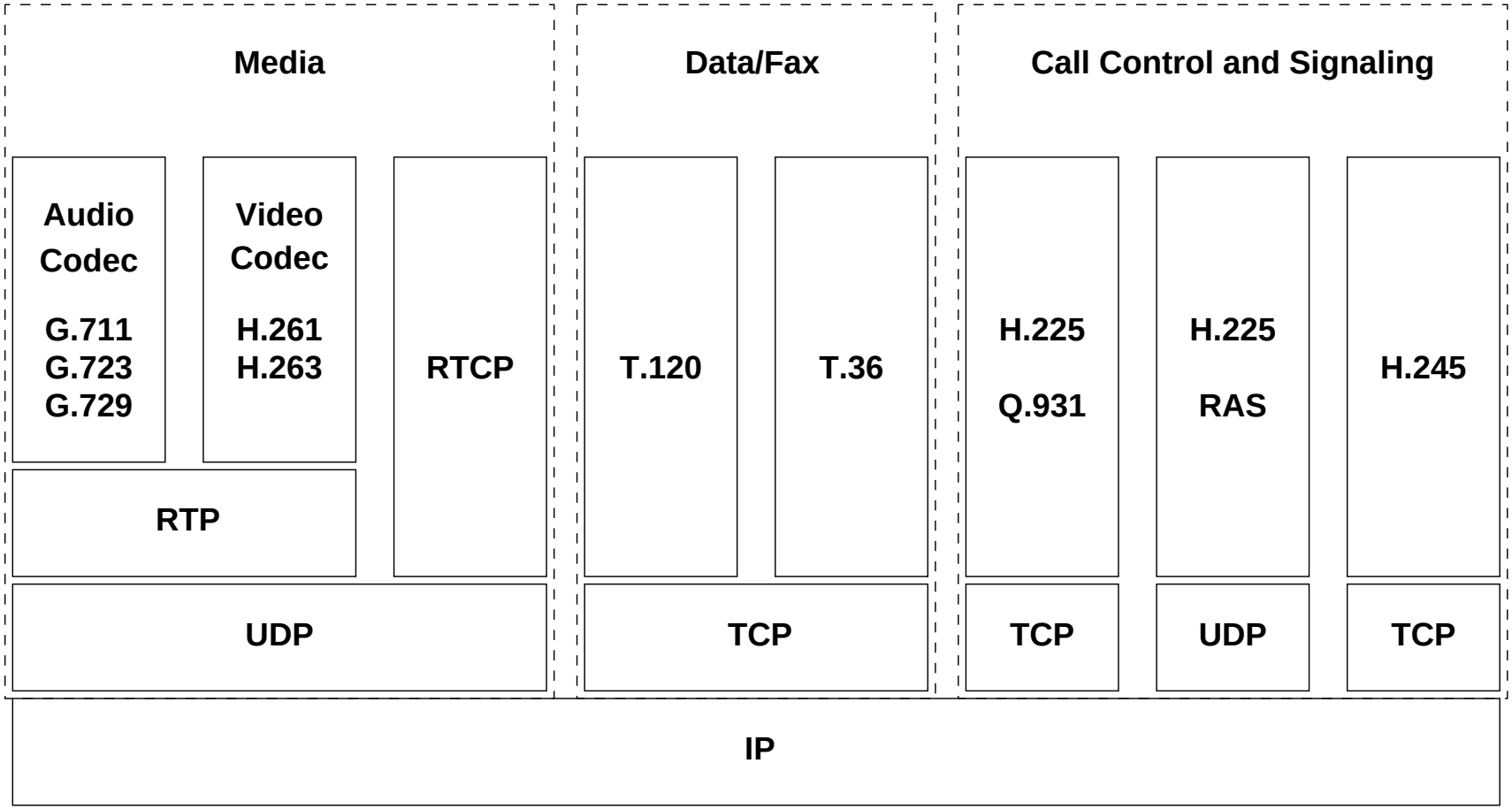
Interworking with the PSTN

- PSTN: Public Switched Telephone Network
- Like to call from a VoIP phone a PSTN phone
- Like to call from a PSTN phone a VoIP phone
- Like to be reachable via the same "call number" regardless where I attach to the network
- Like to be mobile while placing voice calls
- Two big standardization bodies involved:
  - ITU-T (International Telecommunication Union)
  - IETF (Internet Engineering Task Force)

H.323 (ITU-T)

- The H.323 standard provides a foundation for audio, video, and data communications across IP-based networks, including the Internet.
- H.323 is an umbrella recommendation setting standards for multimedia communications over packet switched networks.
- H.323 includes parts of H.225.0 - RAS, Q.931, H.245 RTP/RTCP and audio/video codecs, such as the audio codecs (G.711, G.723.1, G.728, etc.) and video codecs (H.261, H.263) that compress and decompress media streams.
- Media streams are transported on RTP/RTCP.
- The signaling is transported reliably over TCP.

H.323 Protocols



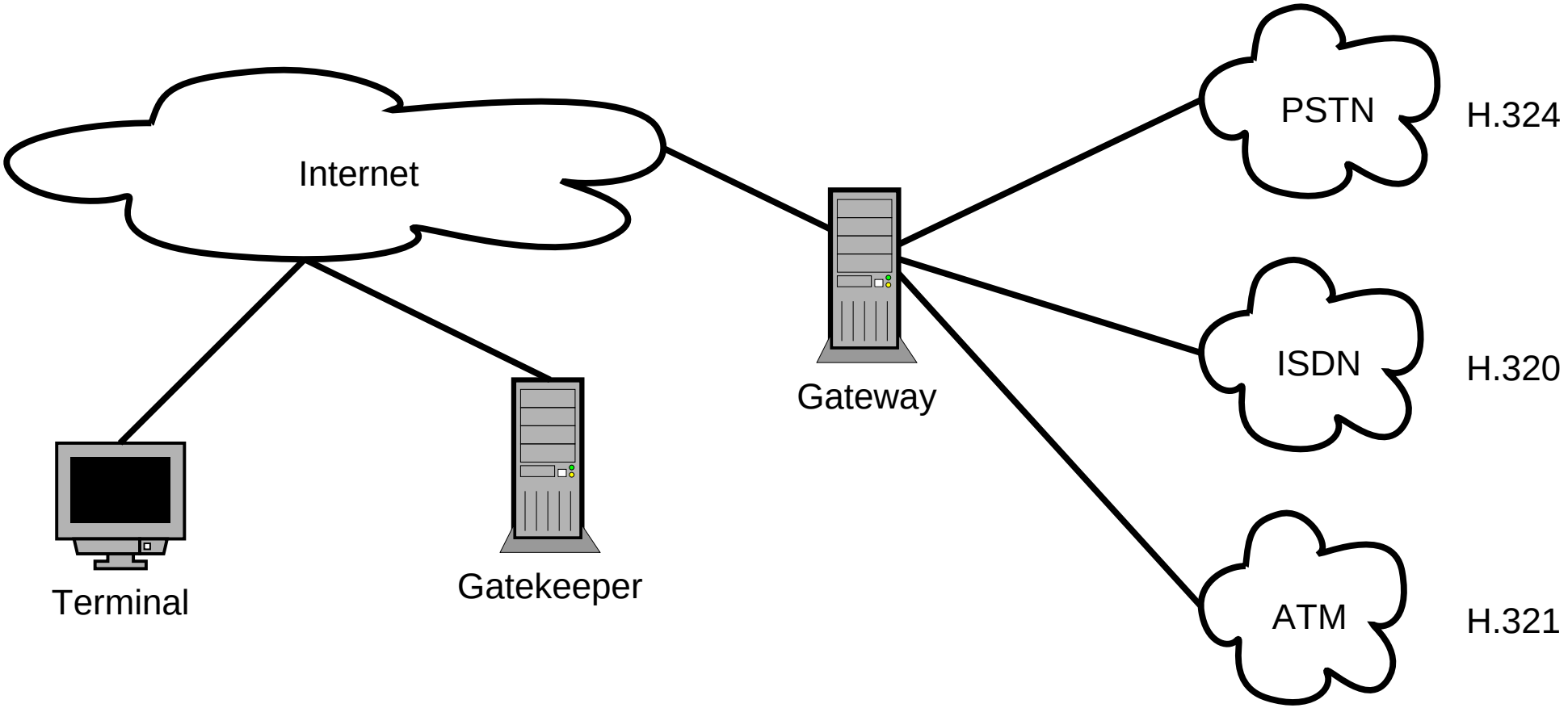
H.323 Terminology

- *Terminal*
  - End system (a phone or multimedia PC) supporting
    - H.225 call control signaling
    - H.245 control channel signaling
    - RTP/RTCP media transport
    - Audio (video) codecs
- *Gateway*
  - Interface to non-H.323 terminals
    - Translation between entities in packet switched networks and circuit switched networks
    - Tramission format and communication procedure translation

H.323 Terminology (cont.)

- *Gatekeeper*
  - Maintains information about terminals (optional).
  - Can perform address translation, admission control, bandwidth control, zone management, call control signaling, call authorization, bandwidth management, call management.
- *Multipoint Control Units*
  - Support for multi-party communication (conferences).
    - The Multipoint Controller (MC) provides control functions.
    - The Multipoint Processor (MP) receives and processes audio, video and/or data streams.

H.323 Interworking



H.323 Communication Establishment

- Communication establishment is done in five steps:
  1. Call setup
  2. Initial communication and capabilities exchange
  3. Audio/video communication establishment
  4. Call services
  5. Call termination

Why SIP and MEGACO

- H.323
  - Too heavy for devices with limited processing power
  - Does not specifically address mobility / roaming
- Session Initiation Protocol (SIP)
  - Designed for lightweight signalling
  - Addresses any media, not voice only
  - Suitable for Internet telephony
- Media Gateway Control Protocols (MGCPs)
  - Protocols for Media Gateways
  - Focuses on PSTN-PSTN via IP

Mean Opinion Scores (MOS) [ITU P.800]

MOS	Quality	Impairment
1	Bad	Very annoying
2	Poor	Annoying
3	Fair	Slightly annoying
4	Good	Perceptible but not annoying
5	Excellent	Imperceptible

- Subjective tests are done by a group of testers. The MOS scores of the testers are averaged to obtain the overall MOS.
- Objective tests are done by using a model to compute MOS scores.

MOS Scores for Several Codecs

Codec	Data Rate	Mean Opinion Score (MOS)
G.711	64 kbit/s	4.1
G.729	8 kbit/s	3.92
G.723.1	6.3 kbit/s	3.9
G.729a	8 kbit/s	3.7
G.723.1	5.3 kbit/s	3.65

- Different codecs achieve different MOS values.
- Trade-off between saving bandwith and quality loss due to the codec itself.

E Model [ITU G.107]

- Idea: Calculate a factor R representing a transmission quality rating
- Definition:

$$R = R_0 - I_s - I_e - I_d + A$$

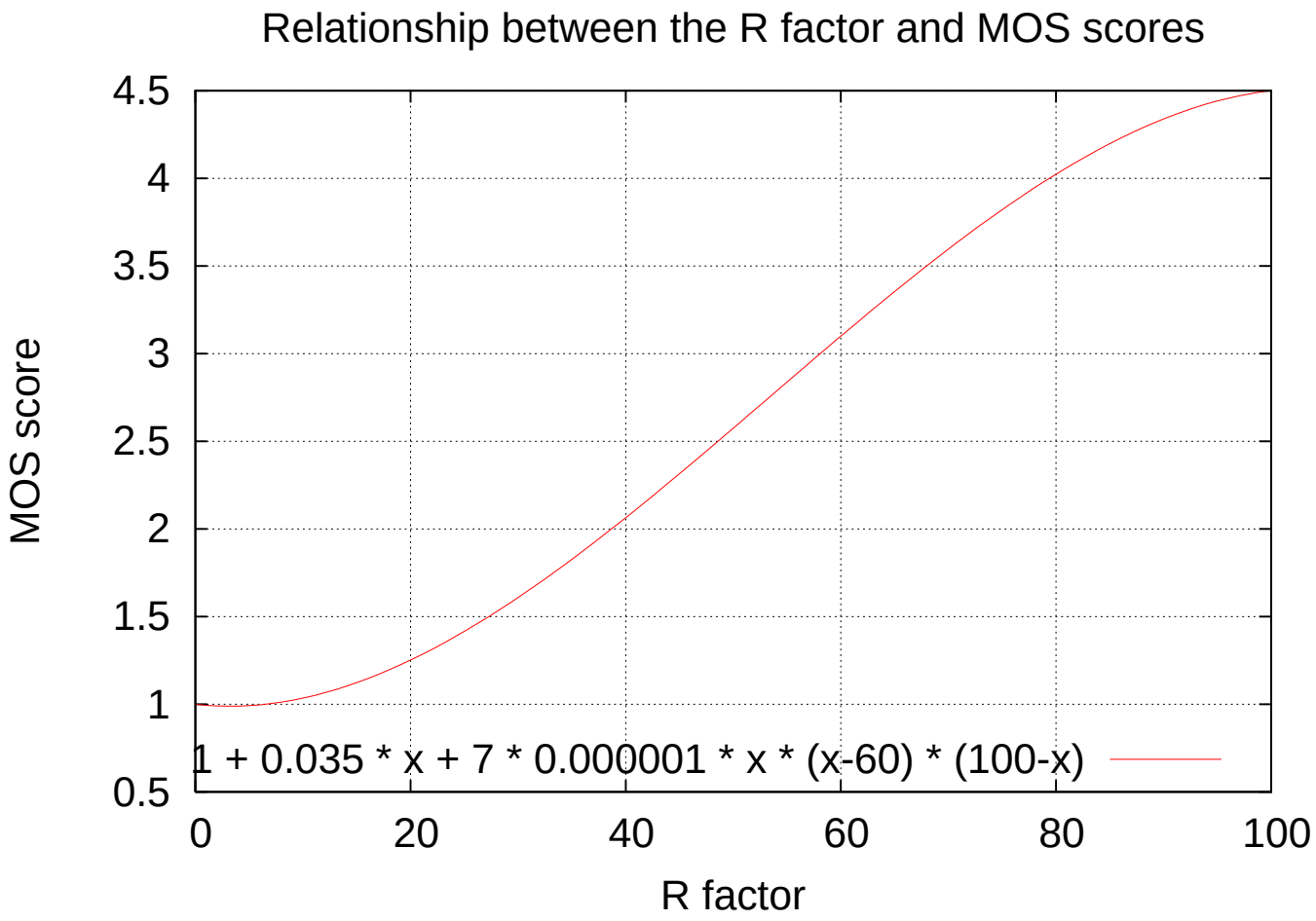
- Parameters:
  - R_0 - noise (S/N at 0 dB)
  - I_s - impairments simultaneous to voice signal
  - I_d - impairments delayed after voice signal
  - I_e - effects of special equipment (e.g., codecs)
  - A - advantage factor
- Assumption: “Psychological factors on the psychological scale are additiv”

Interpretation of the R factor

R-factor	Quality	MOS
90 < R < 100	Best	4.34 - 4.50
80 < R < 90	High	4.03 - 4.34
70 < R < 80	Medium	3.60 - 4.03
60 < R < 70	Low	3.10 - 3.60
50 < R < 60	Poor	2.58 - 3.10

- The R factor is in the range $[0 \ldots 100]$.
- The R factor can be used directly as a quality metric.
- A translation to MOS scores is possible as well.

R factor versus MOS



$$MOS = 1 + 0.035 \cdot R + 7 \cdot 10^{-6} \cdot R \cdot (R - 60) \cdot (100 - R)$$

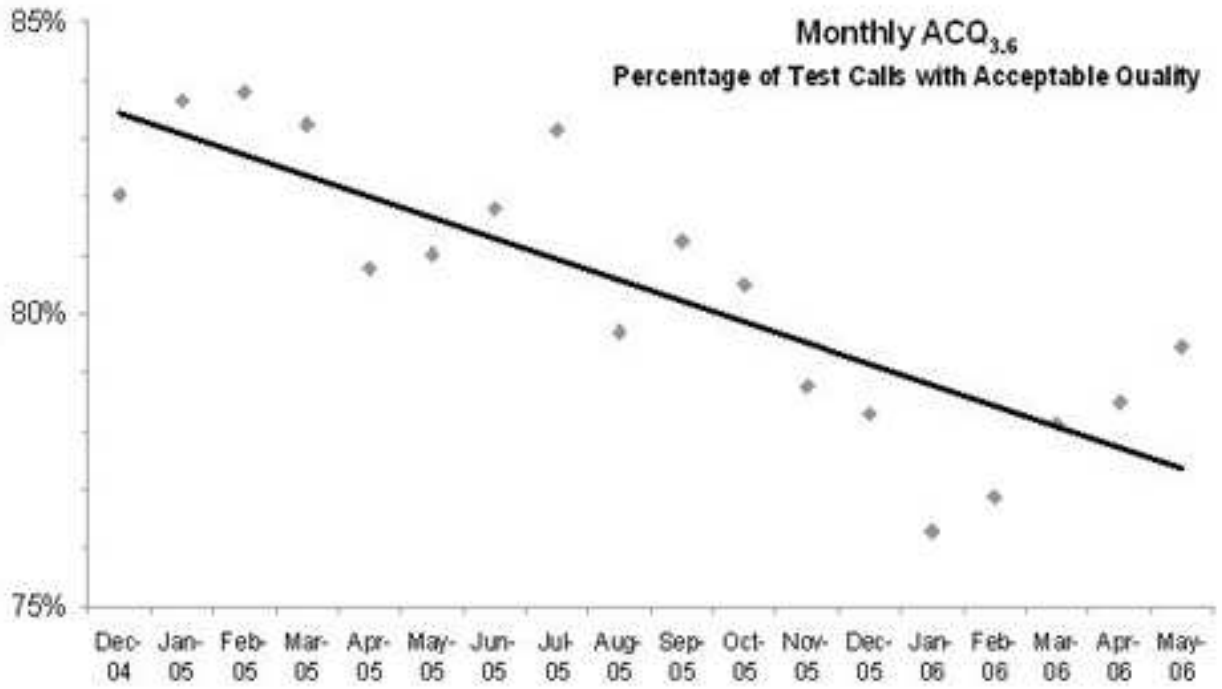
Simplified E Model

- Cole and Rosenbluth [3] have further simplified the model for a VoIP system:

$$R = 94.2 - I_e - Id$$

- Parameters:
  - $I_e = \lambda_1 + \lambda_2 \cdot ln(1 + \lambda_3 e)$
  - $I_d = 0.024 \cdot d + 0.11 \cdot (d - 177.3) \cdot I(d - 177.3)$
  - e - represents the overall packet loss
  - d - represents the total end-to-end delay
  - $I(x)$ - unity step function
  - λ_i - codec parameters (reaction to loss)
- All parameters are easy to measure once the λ_i are known.
- Note that jitter does not at all impact this model.

Large Scale VoIP Measurements



- Brix Networks is running a test site www.TestYourVoIP.com which collects (simulated) call statistics [4].
- The metric $ACQ_{3.6}$ is the number of calls with $MOS \geq 3.6$ recorded on the test systems.

References

[1] ITU. Recommendation P.800: Methods for subjective determination of transmission quality. Recommendation ITU-T P.800, International Telecommunication Union, August 1996.

[2] ITU. Recommendation G.107: A computational model for use in transmission quality. Recommendation ITU-T G.107, International Telecommunication Union, December 1998.

[3] R. G. Cole and J. H. Rosenbluth. Voice over IP performance monitoring. *SIGCOMM Comput. Commun. Rev.*, 31(2):9–24, 2001.

[4] M. Sylor, N. Venna, and H. Ripps. Voice Quality on the Internet in 2005 as Measured by www.TestYourVoIP.com. In *Proc. DSOM 2006*, pages 112–123, Dublin, October 2006. Springer LNCS 3775.

5. Internet Mobility

Motivation

- With the advent of portable devices (PDAs, cell phones, music players, . . .), there is an increasing need to communicate while moving between networks.
- Mobility should ideally not impact any applications running on portable devices.
- Mobility might lead to periods of intermitted network access.
- The Internet was not designed with mobility in mind . . .

Terminology (RFC 3753)

- Fixed Node (FN):
  - A node, either a host or a router, unable to change its point of attachment to the network and its IP address without breaking open sessions.
- Mobile Node (MN):
  - An IP node capable of changing its point of attachment to the network.
  - A Mobile Node may either be a Mobile Host or a Mobile Router.

Terminology (RFC 3753)

- Mobile Host (MH):
  - A Mobile Node that is an end host and not a router.
  - A Mobile Host is capable of sending and receiving packets, that is, being a source or destination of traffic, but not a forwarder of it.
- Mobile Router (MR):
  - A router capable of changing its point of attachment to the network, moving from one link to another link.
  - The MR is capable of forwarding packets between two or more interfaces, and possibly running a dynamic routing protocol modifying the state by which it does packet forwarding.

Terminology (RFC 3753)

- Mobile Network (MN):
  - An entire network, moving as a unit, which dynamically changes its point of attachment to the Internet and thus its reachability in the topology.
  - The mobile network is composed of one or more IP-subnets and is connected to the global Internet via one or more Mobile Routers (MR).
  - The internal configuration of the mobile network is assumed to be relatively stable with respect to the MR.

Handover Terminology (RFC 3753)

- Roaming
  - An operator-based term involving formal agreements between operators that allows a mobile to get connectivity from a foreign network.
- Handover
  - The process by which an active MN changes its point of attachment to the network, or when such a change is attempted.
  - The access network may provide features to minimize the interruption to sessions in progress.
- Seamless Handover
  - A handover in which there is no change in service capability, security, or quality.

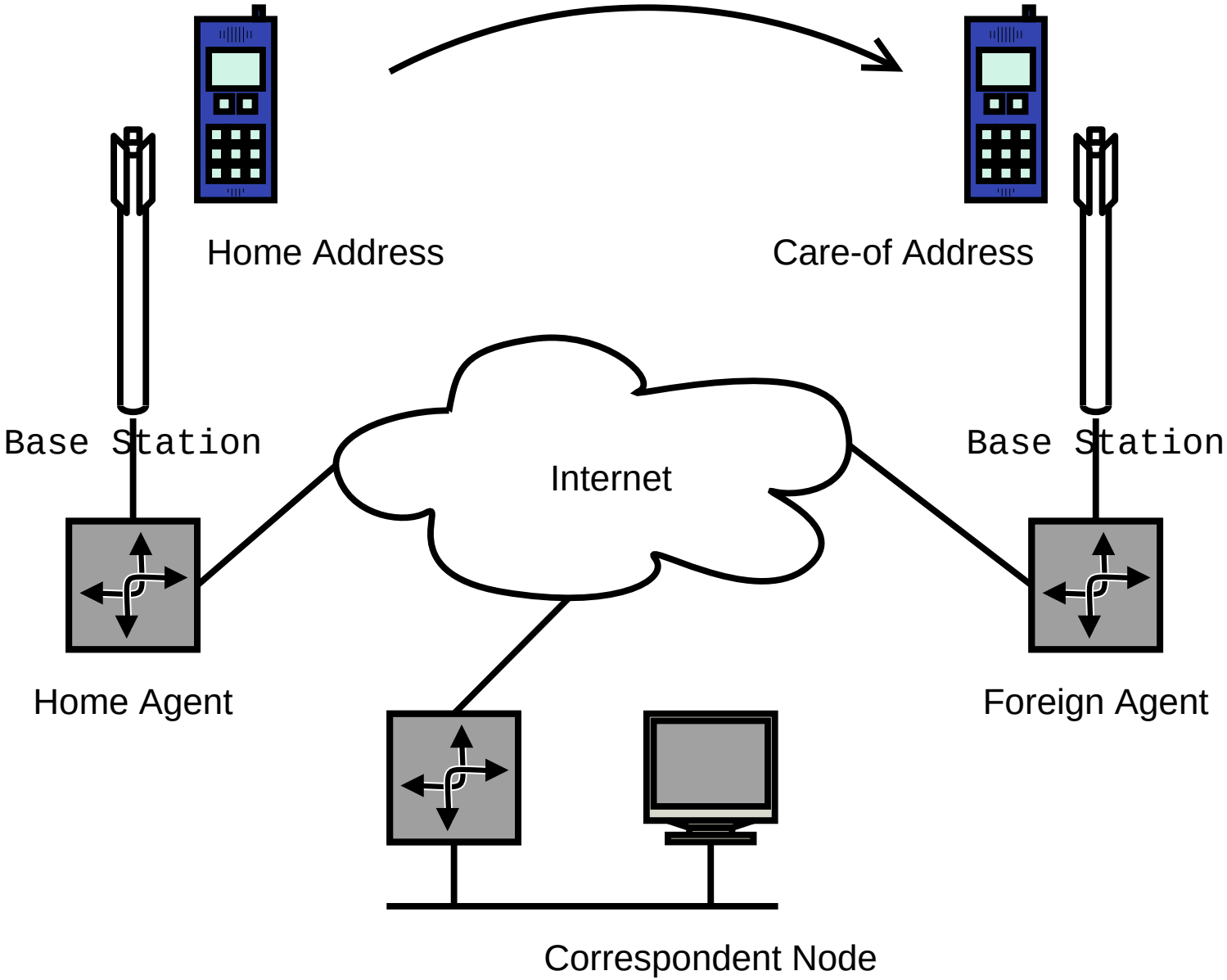
Mobility and Layering

- Link layer
  - xxx
- Network layer
  - xxx
- Transport layer
  - xxx
- Session layer
  - xxx

Mobile IP (MIP) Requirements

- Transparency
  - Mobile nodes keep their IP address
  - Transport and application protocols do not change
- Compatibility
  - Support for the existing layer two protocols
  - No changes needed on existing deployed nodes
- Security
  - Security should not be sacrificed
  - Privacy must be maintained where necessary
- Efficiency and Scalability
  - Minimize overhead for mobility support
  - Must scale to a huge number of mobile nodes (mobile phones)

Mobile IP Scenario



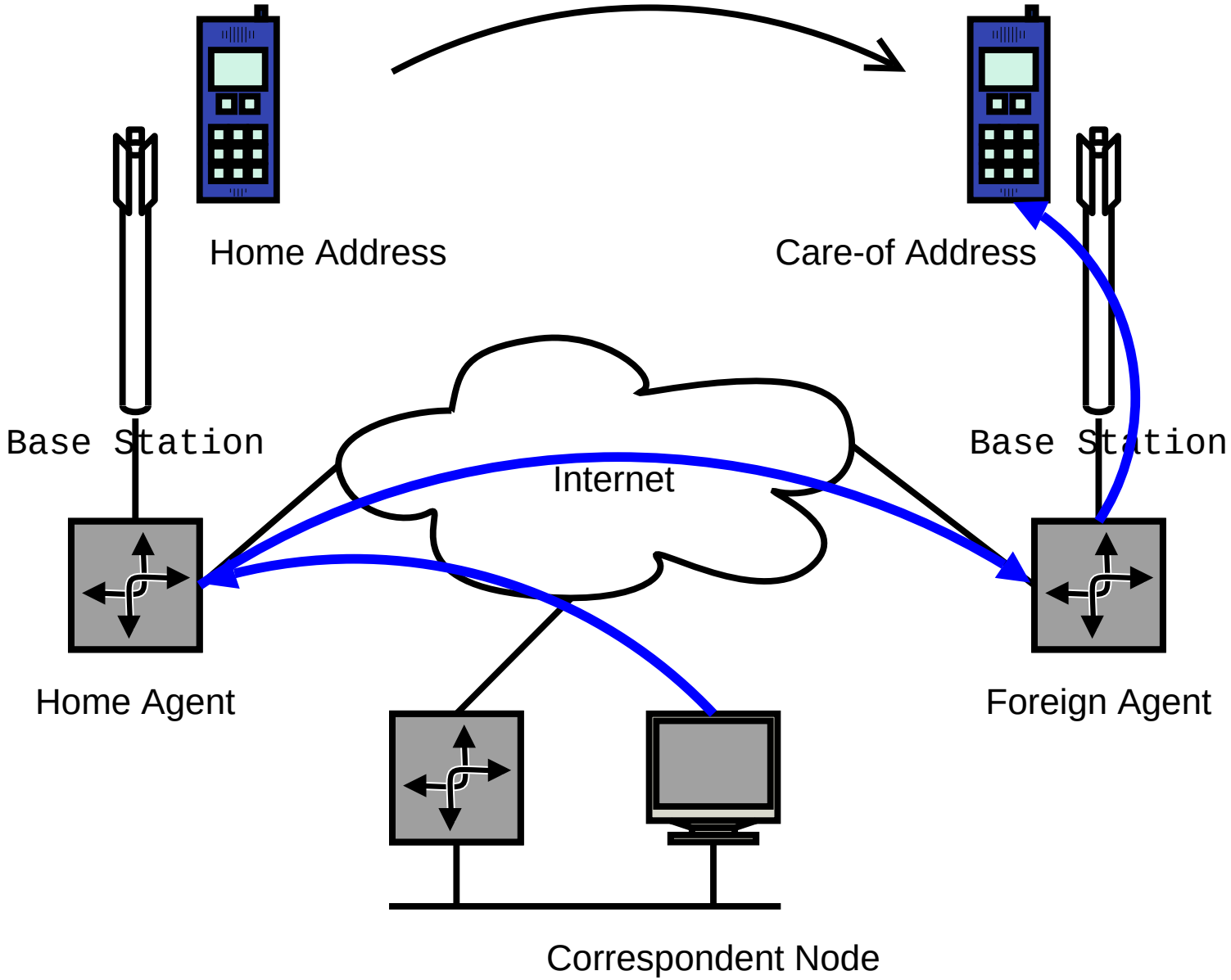
Mobile IPv4 Terminology

- Home Agent
  - A router on a mobile node's home network which tunnels datagrams for delivery to the mobile node when it is away from home, and maintains current location information for the mobile node.
- Foreign Agent
  - A router on a mobile node's visited network which provides routing services to the mobile node while registered.
  - The foreign agent detunnels and delivers datagrams to the mobile node that were tunneled by the mobile node's home agent.
  - For datagrams sent by a mobile node, the foreign agent may serve as a default router for registered mobile nodes.

Mobile IPv4 Terminology

- Home Address
  - An IP address that is assigned for an extended period of time to a mobile node. It remains unchanged regardless of where the node is attached to the Internet
- Care-of Address
  - The termination point of a tunnel toward a mobile node, for datagrams forwarded to the mobile node while it is away from home.
- Correspondent Node
  - A peer with which a mobile node is communicating. A correspondent node may be either mobile or stationary.

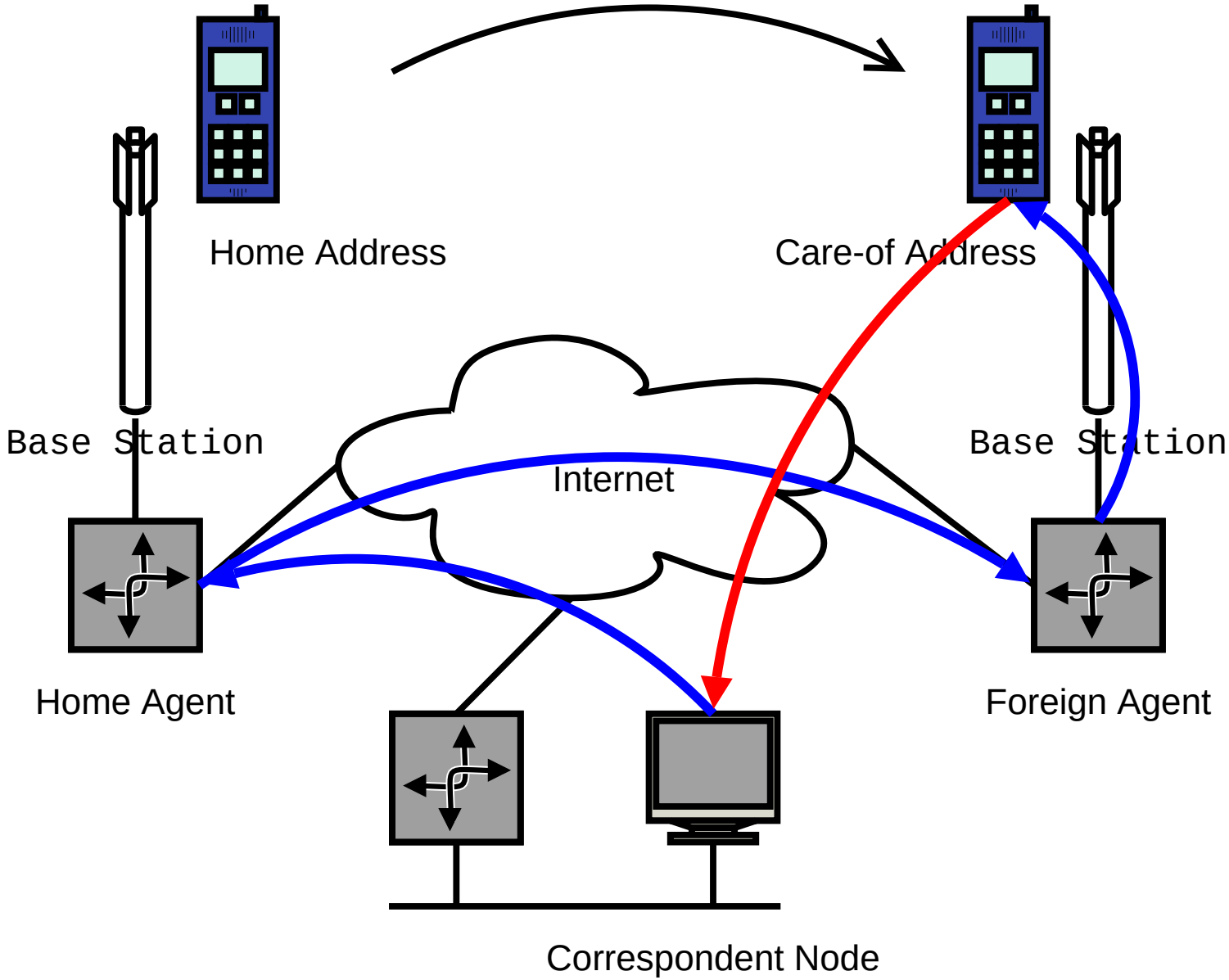
MIPv4 (RFC 3344)



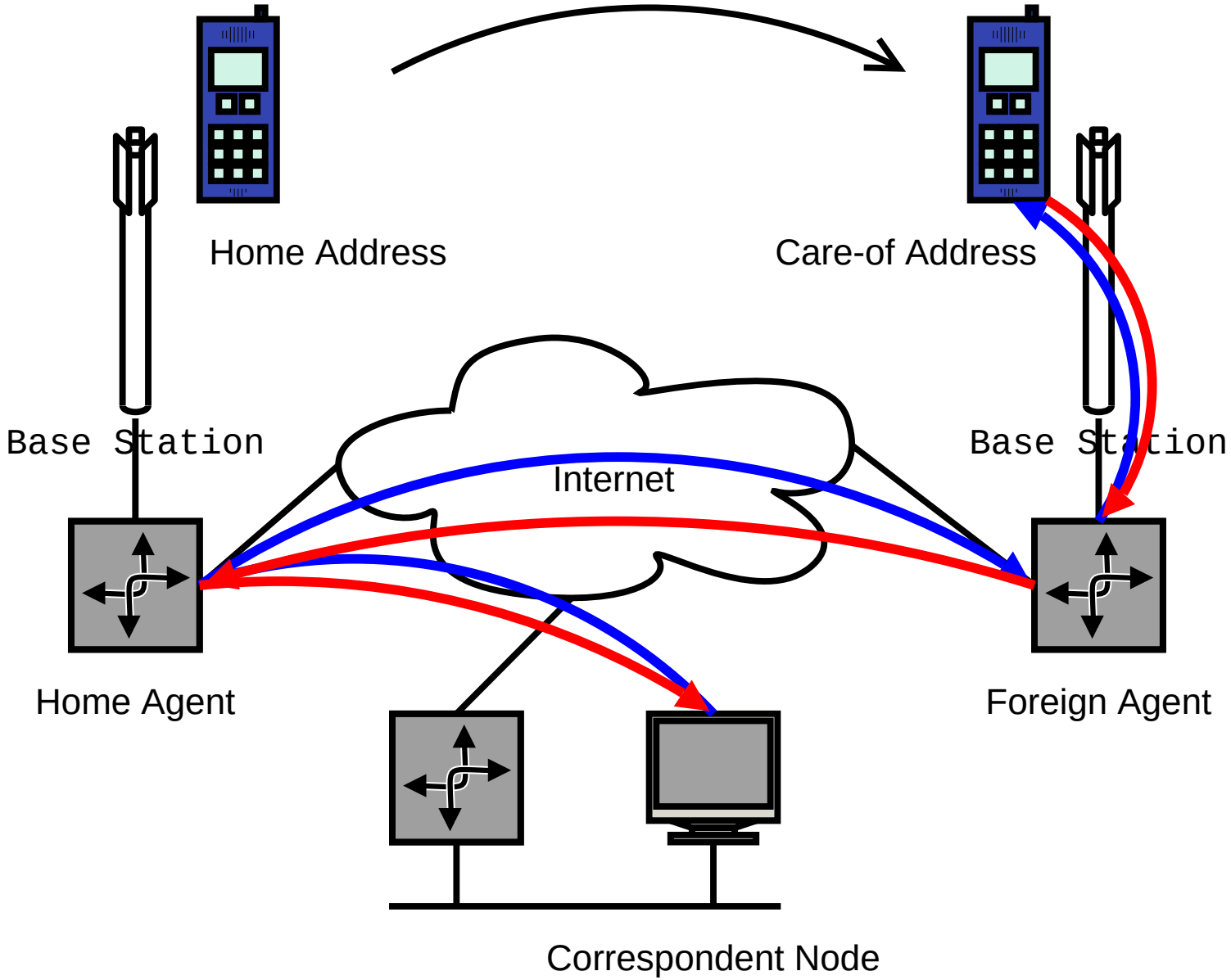
MIPv4 (RFC 3344)

- Correspondent node $\rightarrow$ mobile node:
  - Correspondent node sends IP packets to the home address of the mobile node.
  - Home agent intercepts packets and tunnels them to the current care-of address via the foreign agent.
  - Foreign agent forwards packet to the mobile node.
- Mobile node $\rightarrow$ correspondent node:
  - Direct forwarding (RFC 3344)
  - Reverse tunneling (RFC 3024)

MIPv4 (RFC 3344)



MIPv4 (RFC 3344, RFC 3024)



Direct Forwarding vs. Reverse Tunneling

- Problems with direct forwarding:
  - Asymmetric routing may cause routing problems
  - Firewall problems (topological incorrect addresses)
  - Potential problems with multicast support
  - Loss of transparency
- Problems with reverse tunneling:
  - Double triangular routing costs
  - Reverse tunnel may be misused to bypass firewalls

Other MIPv4 Issues

- Congestion Control
  - Transport protocols may get confused if the properties of the communication path change (PMTU, congestion state, . . .).
- Convergence
  - Convergence time may be significant.
  - Link layer notifications may be used to trigger the handover early.
  - It might be necessary to forward packets via multiple paths during handover.
- Security and Privacy
  - Mobile IP should not introduce any new security and privacy issues (but it does in reality).

MIPv4 Agent Discovery

- Mobile hosts discover agents (home or foreign) by listening to agent advertisements which are multicasted to the "all systems on this link" multicast address (224.0.0.1).
- The Agent Advertisement is send together with a router advertisement and contains paramters (encapsulation mechanisms) and care-of addresses.
- A mobile host can ask for agent advertisements by sending a router solicitation message.

MIPv4 Registration

- Registration messages are exchanged between a mobile node, (optionally) a foreign agent, and the home agent.
- Registration creates or modifies a *mobility binding* at the home agent, associating the mobile node's home address with its care-of address for the specified lifetime.
- Registration messages must be authenticated using keyed message digests and nonces or timestamps for replay protection.
- The home agent uses proxy ARP and gratuitous ARP while the mobile host is registered on a foreign network to intercept and redirect traffic.

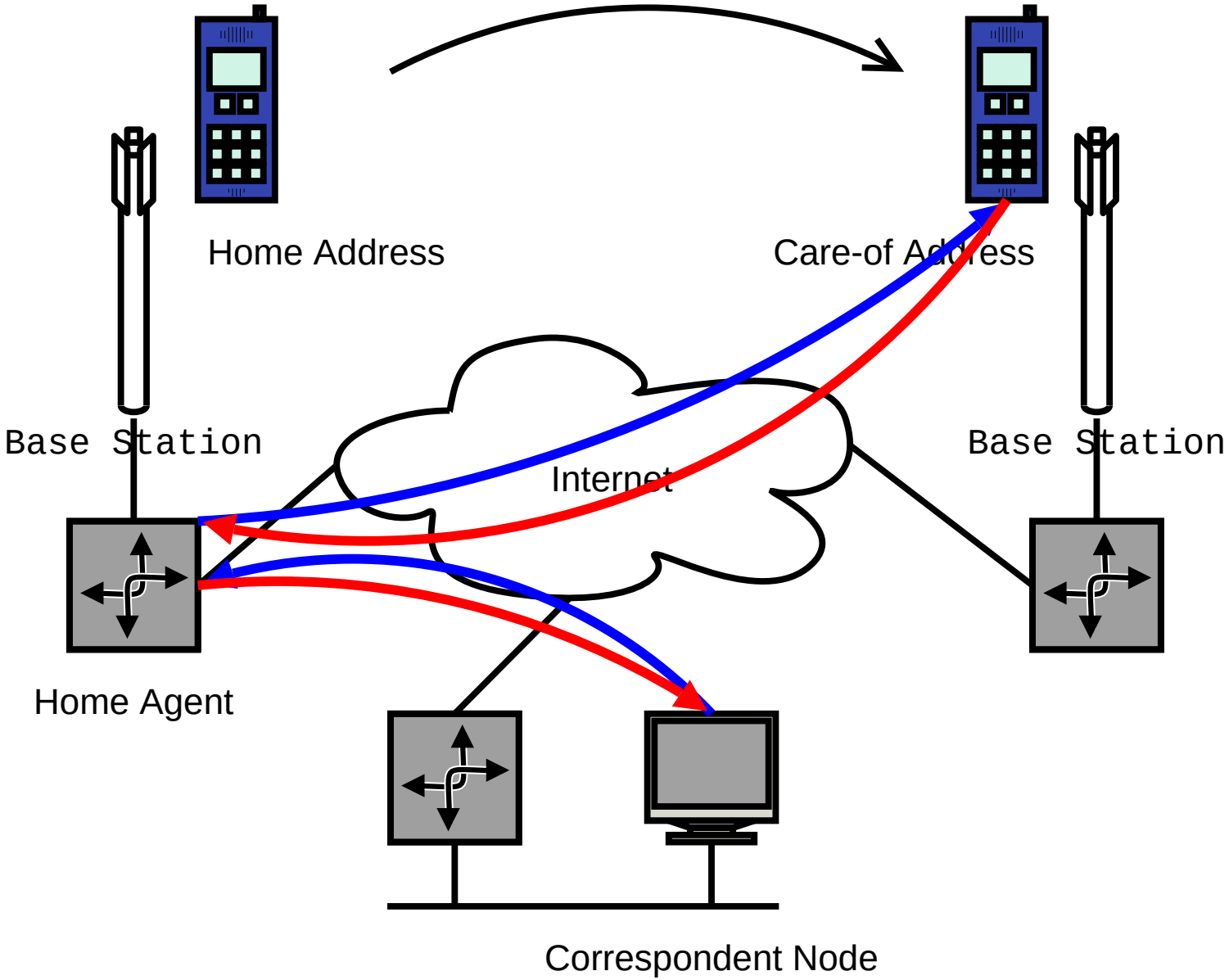
MIPv6 vs. MIPv4 (RFC 3775)

- There is no need to deploy foreign agents in MIPv6.
- Support for route optimization is a fundamental part of MIPv6.
- Mobile IPv6 route optimization can operate securely even without pre-arranged security associations.
- Most packets sent to a mobile node while away from home in Mobile IPv6 are sent using an IPv6 routing header rather than IP encapsulation, reducing the amount of resulting overhead compared to Mobile IPv4.
- Mobile IPv6 is decoupled from any particular link layer, as it uses IPv6 Neighbor Discovery.
- . . .

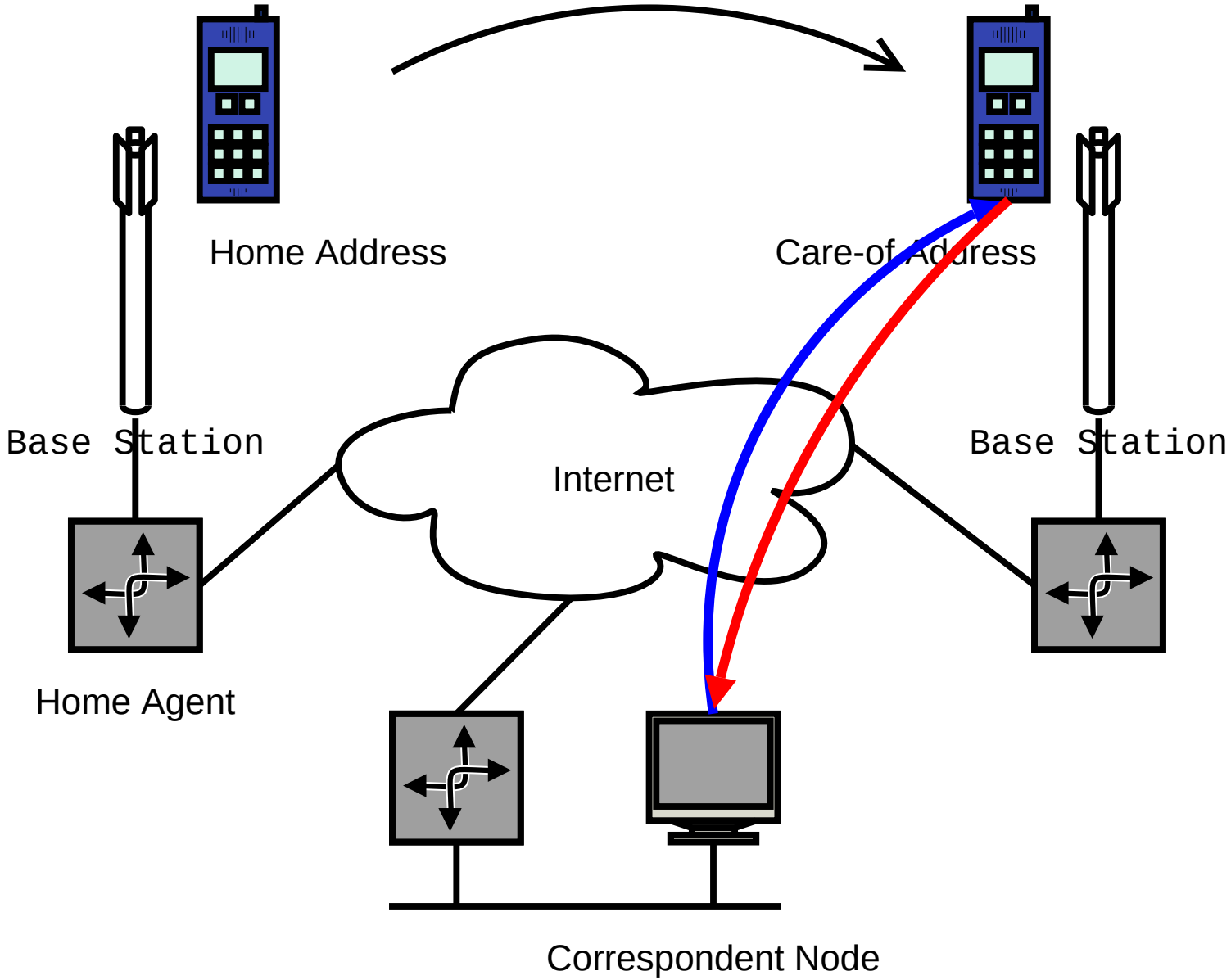
MIPv6 Principles (RFC 3775)

- Mobile nodes acquire care-of addresses via auto-configuration. (Mobile nodes may have multiple care-of addresses.)
- Mobile nodes register their primary care-of address with a router on their home links (binding).
- Mobile nodes can register with correspondent nodes to establish a direct binding.
- Mobile nodes can dynamically discover their home agent, even when the mobile node is away from home.

MIPv6 Tunnel Mode (RFC 3775)



MIPv6 Route Optimization (RFC 3775)



MIPv6 Principles

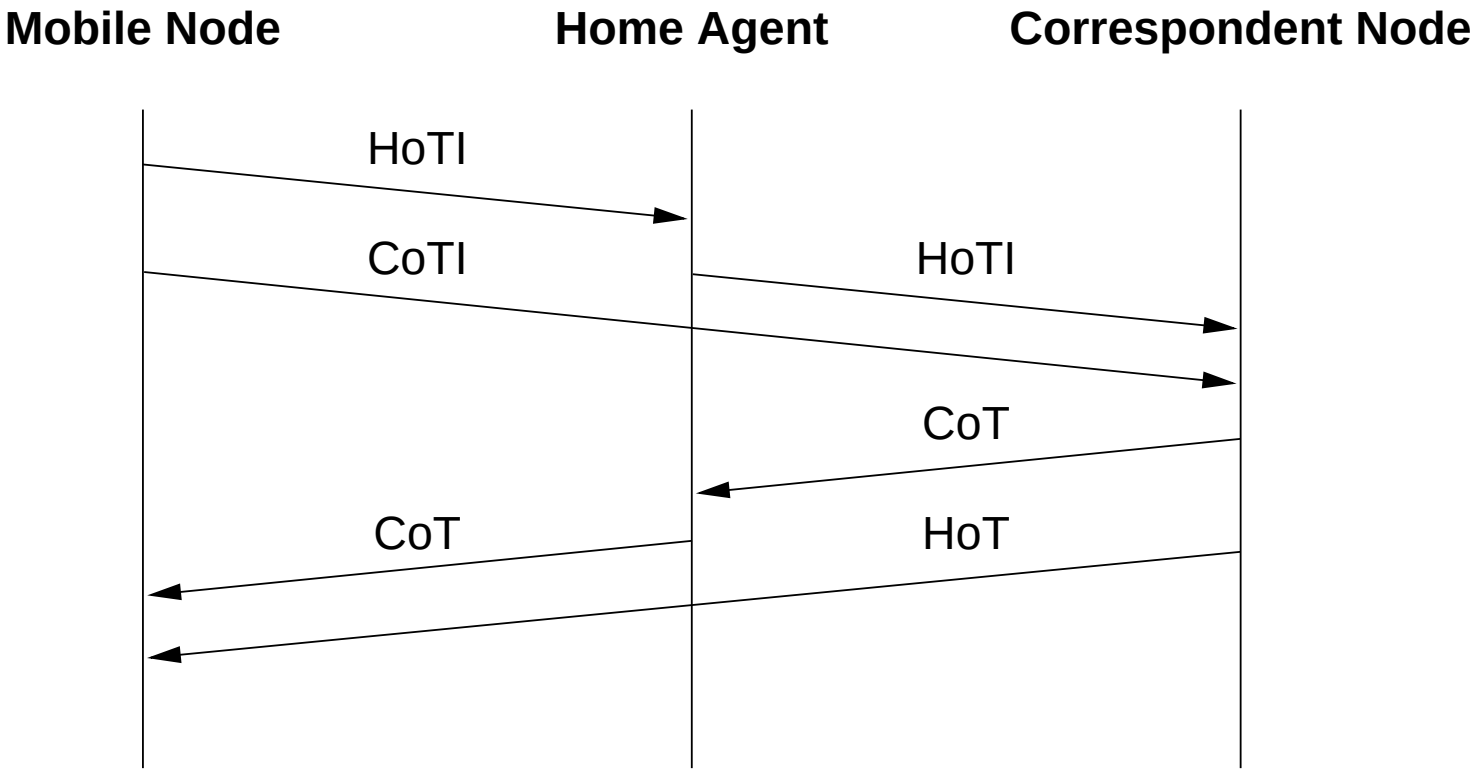
- Bidirectional tunneling mode:
  - The home agent intercepts traffic addressed to the mobile node by neighbor discovery intercept and tunnels the traffic to the mobile node's primary care-of address using IPv6 encapsulation.
- Route optimization mode:
  - After registration at the correspondent node, the correspondent node sends traffic directly to the mobile node using a new type of IPv6 routing header.
  - The mobile node directly sends packets to the correspondent node, storing its home address in a new home address destination option.

MIPv6 Security

- Binding updates between the mobile node and the home agent are protected by IPsec, which requires pre-installed security associations.
- Binding updates between the mobile node and the correspondent nodes use a procedure called "return routability procedure", which does not require an authentication infrastructure.
- The "return routability procedure" enables the correspondent node to obtain some *reasonable assurance* that the mobile node is in fact addressable at its claimed care-of address as well as at its home address.

$\Rightarrow$ Does not protect against attackers who are on the path between the home network and the correspondent node.

Return Routability Procedure

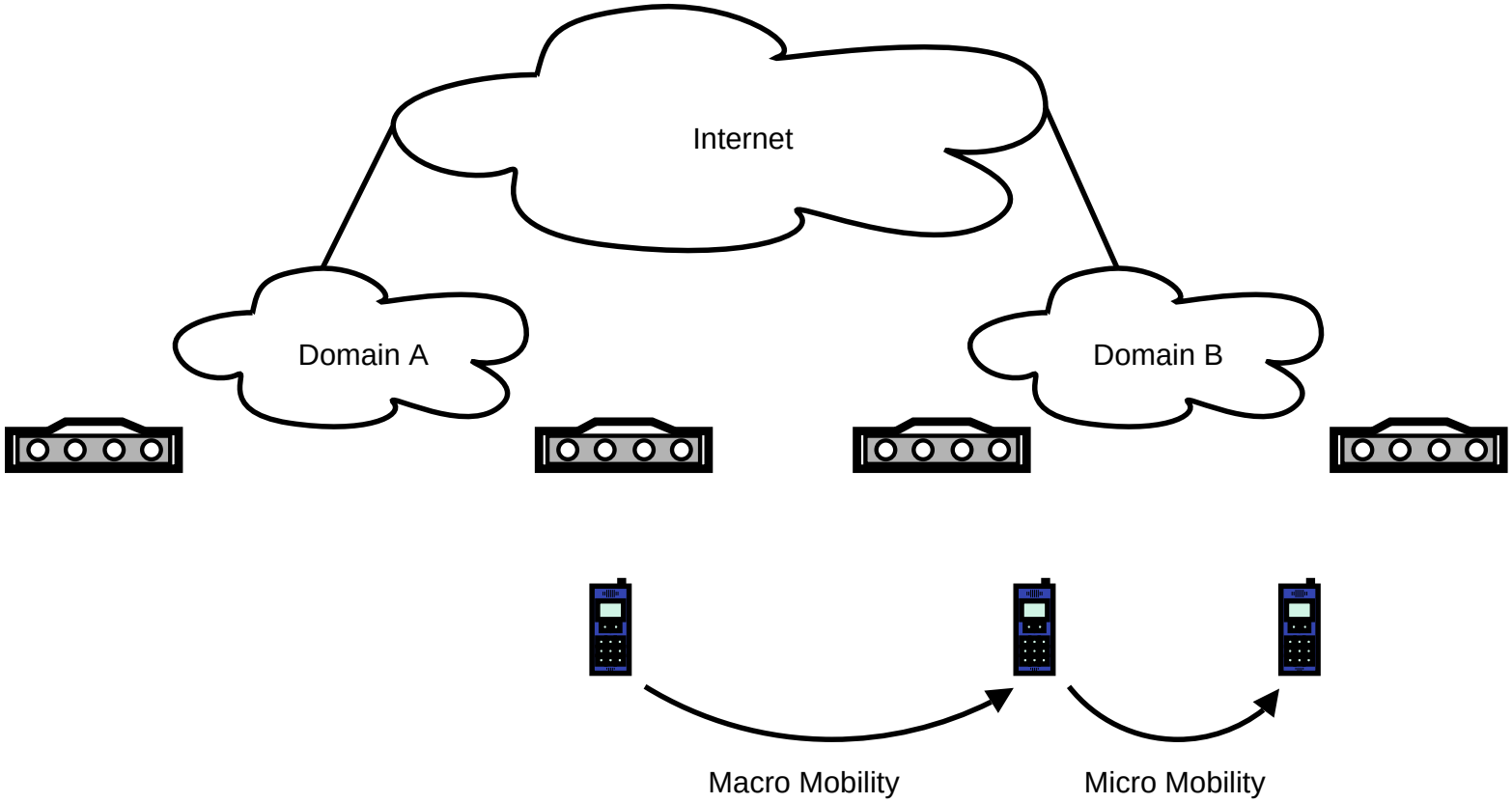


- Home Test Init (HoTI), Home Test (HoT)
- Care-of Test Init (CoTI), Care-of Test (CoT)

Return Routability Procedure

- Testing whether packets addressed to the two claimed addresses are routed to the mobile node.
- A mobile node can pass the test only if it is able to supply proof that it received certain data (the "keygen tokens") which the correspondent node sends to those addresses.
- These data are combined by the mobile node into a binding management key, denoted K_{bm}.
- $K_{bm} = SHA1(home\_keygen\_token|care\_of\_keygen\_token)$
- The binding management key K_{bm} is checked by the correspondent node during the binding update.
- Nonces are used to guarantee the freshness of the messages.

Micro Mobility vs. Macro Mobility



- Mobile IP works reasonably well for macro mobility
- Micro mobility requires additional work for seamless handover

Fast Handover for MIPv6 (RFC 4260)

- Goals:
  - Reduce the time needed to restore IP connectivity.
  - Support real-time services in a mobile network.
- Mechanisms:
  - Leverage information from the link-layer to predict and rapidly respond to handover events.
  - Tunnel data between the old and new access routers.
  - Provide IP connectivity in advance of actual mobile IP registration.
- Generic fast handover extension plus link-layer specific adaptations.

802.11 Handover Procedure

0 A station (S) realizes that a handoff is necessary due to degrading radio transmission environment for the current access point (AP).

1 S performs a scan to see what APs are available. The result of the scan is a list of APs together with physical layer information, such as signal strength.

2 S chooses one of the APs and performs a join to synchronize its physical and MAC layer timing parameters with the selected AP.

3 S requests authentication with the new AP. For an "Open System", such authentication is a single round-trip message exchange with null authentication.

802.11 Handover Procedure (cont.)

4 S requests association or re-association with the new AP. A re-association request contains the MAC-layer address of the old AP, while a plain association request does not.

5 If operating in accordance with 802.11i, S and AP would execute 802.1X EAP-on-LAN procedures to authenticate the association.

6 If operating in accordance with the IAPP, AP may contact the old AP to transfer some information about the session and clean up the state at the old AP.

7 The new AP sends a Layer 2 Update frame on the local LAN segment to update the learning tables of any connected bridges.

802.11 Implementation Issues

- Some NICs scan the network periodically in the background while others do it only when needed.
- Scanning may take several hundred milliseconds to complete.
- Some implementations to the first steps in firmware, making it impossible for the host to get involved.
- Some implementations decide in firmware which AP is being selected, leaving the host without control over this decision.
- The coverage area of an AP is called as its Basic Service Set (BSS). Several APs with a common ESSID can form an Extended Service Set (ESS). Handover between ESSs may require a fast MIP handover.

Fast Handover for MIPv6 and 802.11

a The MN sends a Router Solicitation for Proxy (RtSolPr) to find out about neighboring Access Routers (ARs).

b The MN receives a Proxy Router Advertisement (PrRtAdv) containing [AP-ID, AR-Info] tuples.

c The MN sends a Fast Binding Update (FBU) to the Previous Access Router (PAR).

d The PAR sends a Handover Initiate (HI) message to the New Access Router (NAR).

e The NAR sends a Handover Acknowledge (HAck) message to the PAR.

f The PAR sends a Fast Binding Acknowledgement (FBack) message to the MS the new link.

g The MN sends Fast Neighbor Advertisement (FNA) to the NAR after attaching to it.

FMIPv6 802.11 Scenarios

- 1abcdef234567g (predictive mode)
  - Requires that the scan and join operations (steps 1 and 2) can be performed separately and under host control.
  - The scan data must be recent once the FMIP handover procedure is triggered (otherwise it might choose the wrong AP).
- ab1234567cdefg (reactive mode)
  - Does not require host intervention of the link-layer handover.
  - Requires that the mobile node obtains the link-layer address of the NAR prior to handover.

FMIPv6 802.11 Scenarios (cont.)

- 1234567abcdefg (reactive mode)
  - Completely reactive, consists of soliciting a router advertisement after handover.
  - Does not require support from the firmware.

References

[1] J. Manner and M. Kojo. Mobility Related Terminology. RFC 3753, June 2004.

[2] C. Perkins. IP Mobility Support for IPv4. RFC 3344, Nokia Research Center, August 2002.

[3] G. Montenegro. Reverse Tunneling for Mobile IP, revised. RFC 3024, Sun Microsystems, Inc., January 2001.

[4] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775, Rice University, Nokia Research Center, Ericsson, June 2004.

[5] P. McCann. Mobile IPv6 Fast Handovers for 802.11 Networks. RFC 4260, Lucent Technologies, November 2005.

[6] W.M. Eddy. At what layer does mobility belong? *IEEE Communications Magazine*, 42(10):155–159, October 2004.

[7] T. Koponen, P. Eronen, and Mikko Saärelä. Resilient connections for SSH and TLS. In *Proc. of USENIX Annual Technical Conference 2006*, Boston, May 2006.

6. Multiprotocol Label Switching

References

[1] W. Stallings. MPLS. *The Internet Protocol Journal*, 4(3):2–14, September 2001.

[2] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, Cisco Systems, Force10 Networks, Juniper Networks, January 2001.

[3] E. Mannie. Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945, October 2004.

[4] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas. LDP Specification. RFC 3036, Nortel Networks, Ennovate Networks, IBM, PhotonEx, Cisco Systems, January 2001.

[5] A. Viswanathan, N. Feldman, Z. Wang, and R. Callon. Evolution of Multiprotocol Label Switching. *IEEE Communications Magazine*, 36(5):165–173, May 1998.

[6] A. Banerjee, J. Drake, J. Land, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekther. Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques. *IEEE Communications Magazine*, 39(1):144–151, January 2001.

7. High-Speed TCP

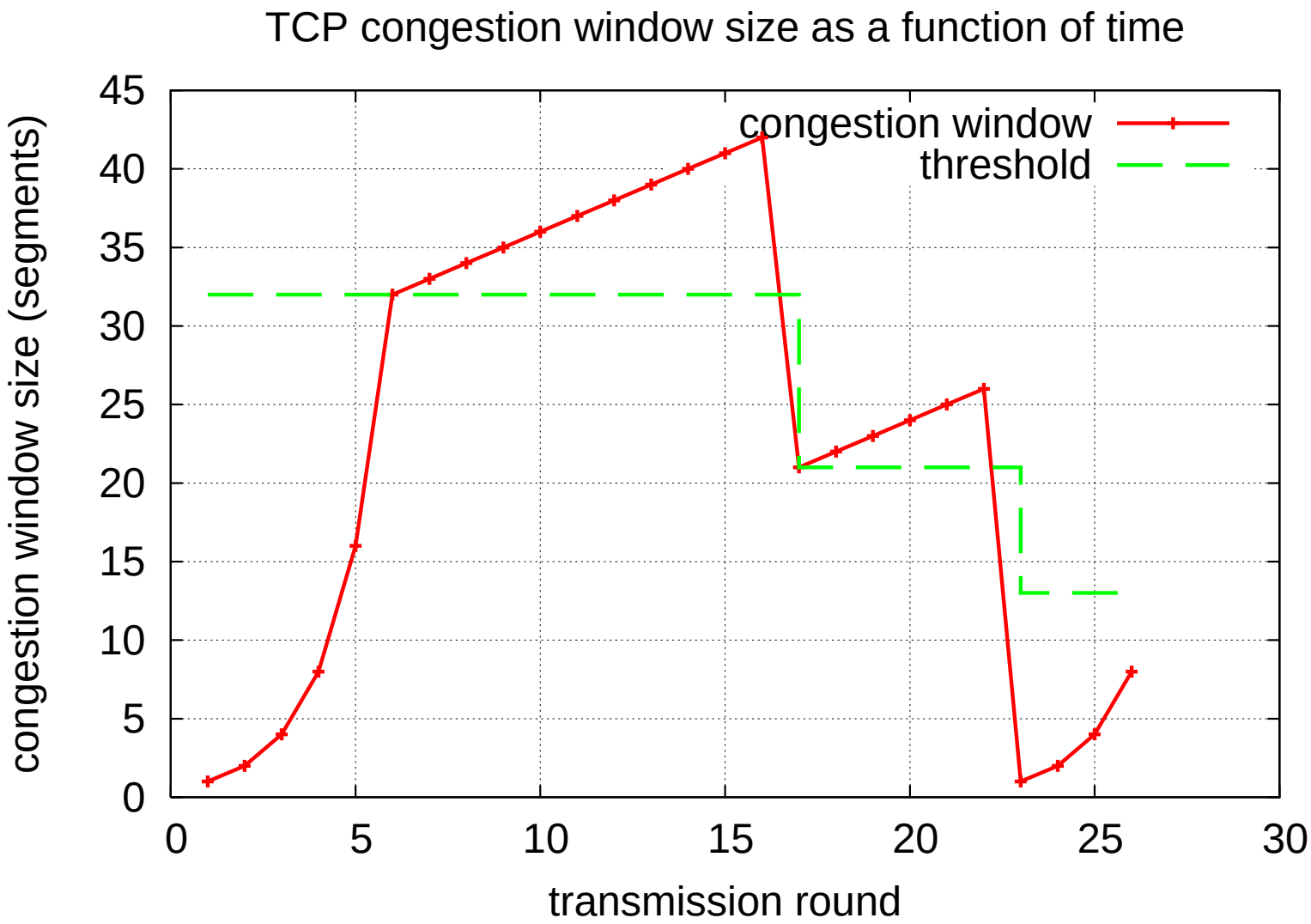
High-Speed TCP

- How well does TCP operate in gigabit / terrabit per second networks?
- What is the speed that can be realized today in real high speed networks spanning large distances?
- What changes are needed to work well over networks with large bandwidth-delay products?

TCP Review

- Slow start mode:
  - Send two TCP segments in response to each ACK that advances the sender's window.
  - Exponential increase of the sending rate
- Congestion avoidance mode:
  - Send an additional segment of data for each loss-free round-trip time interval
  - Linear increase of the sending rate
- Threshold controls transition from slow start mode to congestion avoidance mode
- Timeout causes transition from congestion avoidance mode to slow start mode
- Multiple duplicate ACKs cause TCP to halve the sending rate and to enter congestion avoidance

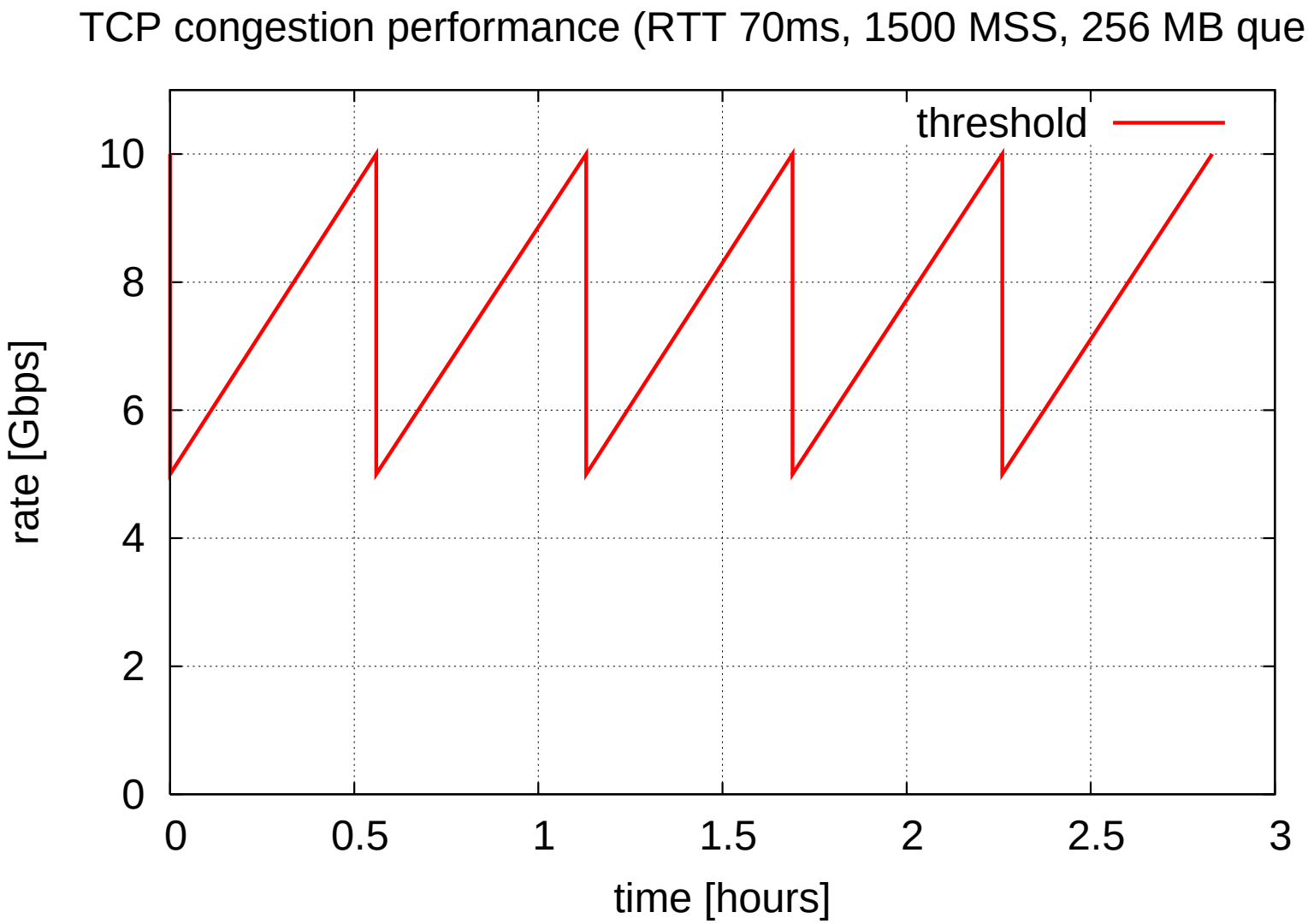
TCP Performance over Time



Example: TCP at High Speed

- 10 Gbps path, 1500 byte segments, 70ms delay
- In slow start, speed doubles every 70ms, so after 17 round trip times, the speed exceeds 10 Gbps and this takes 1.2 seconds
- Duplicate ACKs force congestion avoidance, which halves the congestion window and then increases linearly
- TCP congestion avoidance mode causes a sawtooth oscillation of this idea TCP session between 5 Gbps and 10 Gbps; a single iteration of this cycle takes 34 minutes and 22 seconds
- This model assumes no packet loss due to bit errors and it implies massive data sets to be transferred

TCP Behavior at High Speed



TCP Land Speed Records

- http://lsr.internet2.edu/
- Incredients:
  - It is essential to have the network path all to yourself
  - It is essential to have a fixed latency
  - It is essentail to have an extremly lot bit error rate
  - It is essential to know in advance the round-trip latency and the available bandwidth
- What can we do if we want TCP to work well without these incredients?

Parallel TCP Streams

- Idea:
  - Use multiple TCP streams in parallel
  - Packet loss affects ideally only one stream while the others continue probing
- Properties:
  - Requires that data streams can be partitioned
  - Packet loss can lead to synchonization of the streams
  - Examples: GridFTP, Bittorrent

MulTCP

- Idea:
  - Increase by N segments per RTT
  - Reduce window W by $W/(2N)$ upon packet loss
  - Emulates multiple parallel TCP streams
- Properties:
  - Choosing N too high causes unfairness
  - Choosing N too low means to not use the full network bandwidth

HighSpeed TCP

- Idea:
  - Increase 1 segment up to 10 Mbps, 6 segments up to 100 Mbps, 26 segments at 1 Gbps, 70 segments at 10 Gbps
  - Reduce by $W/2$ at 10 Mbps, $W/3$ at 100 Mbps, $W/5$ at 1 Gbps, $W/10$ at 10 Gbps
  - Limit slow start to not overload the network
- Problems:
  - x
  - x

Scalable TCP

- Idea:
  - Use a multiplicative increase instead a linear increase
  - Reduce to $0.875W$
  - Make probing times proportional only to the round trip time, ignoring the current sending rate
- Properties:
  - Higher frequency of oscillation
  - Use Scalable TCP only used for windows above a certain size to allow smooth coexistance

BIC

- Idea:
- Properties:

CUBIC

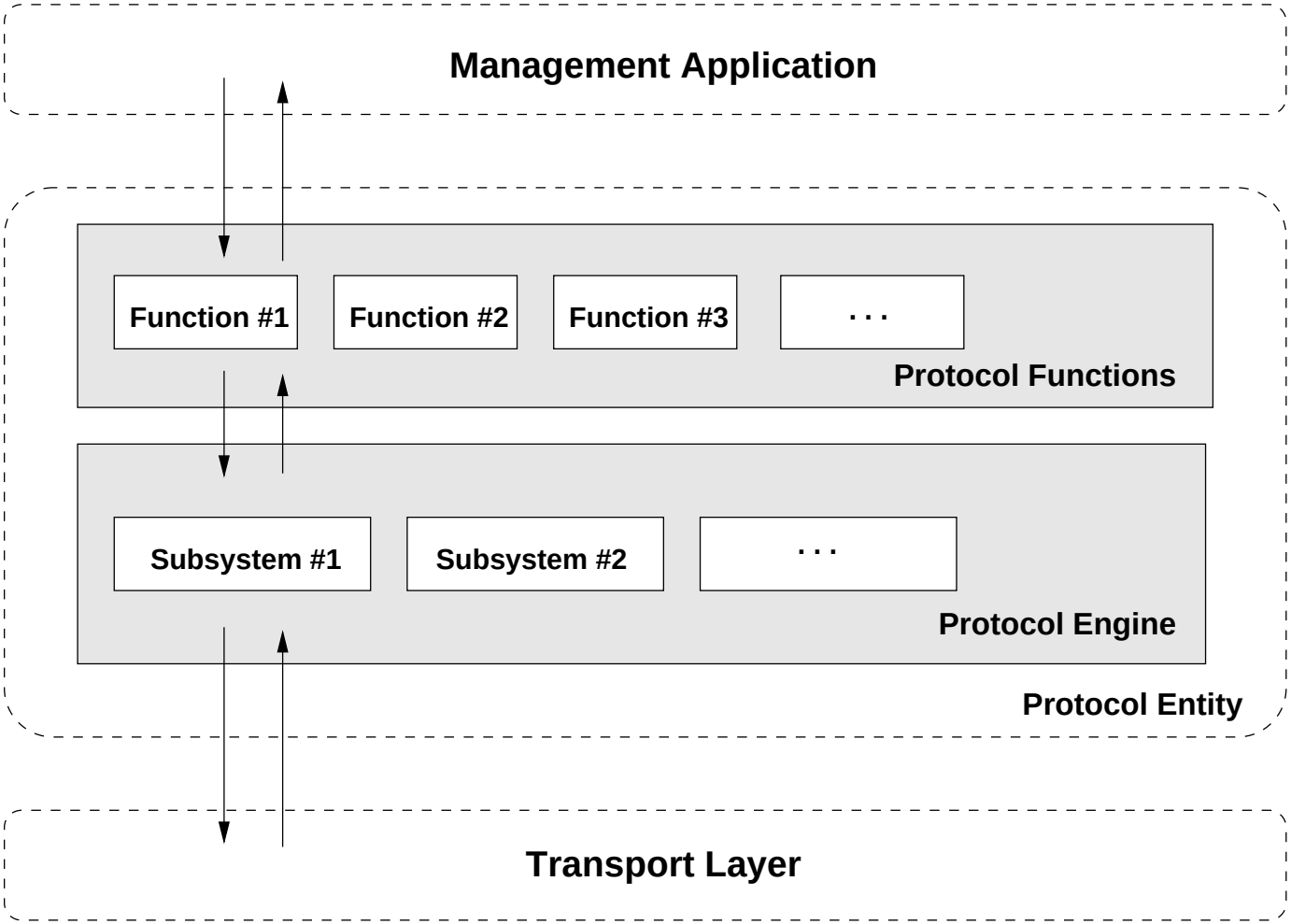
- Idea:
- Properties:

References

[1] G. Huston. Gigabit TCP. *The Internet Protocol Journal*, 9(2), June 2006.

8. Network Management and Measurement

Architectural Concepts



Architectural Concepts

- An protocol *Entity* can be decomposed into a protocol engine and protocol functions.
- A protocol *Engine* is concerned with the handling of protocol messages.
- Protocol *Functions* realize the protocol's functionality.
- *Subsystems* provide a well defined functionality which is accessible through a defined subsystem interface.
- *Models* implement a subsystem interface. There may be one or multiple models that implement the same subsystem interface.

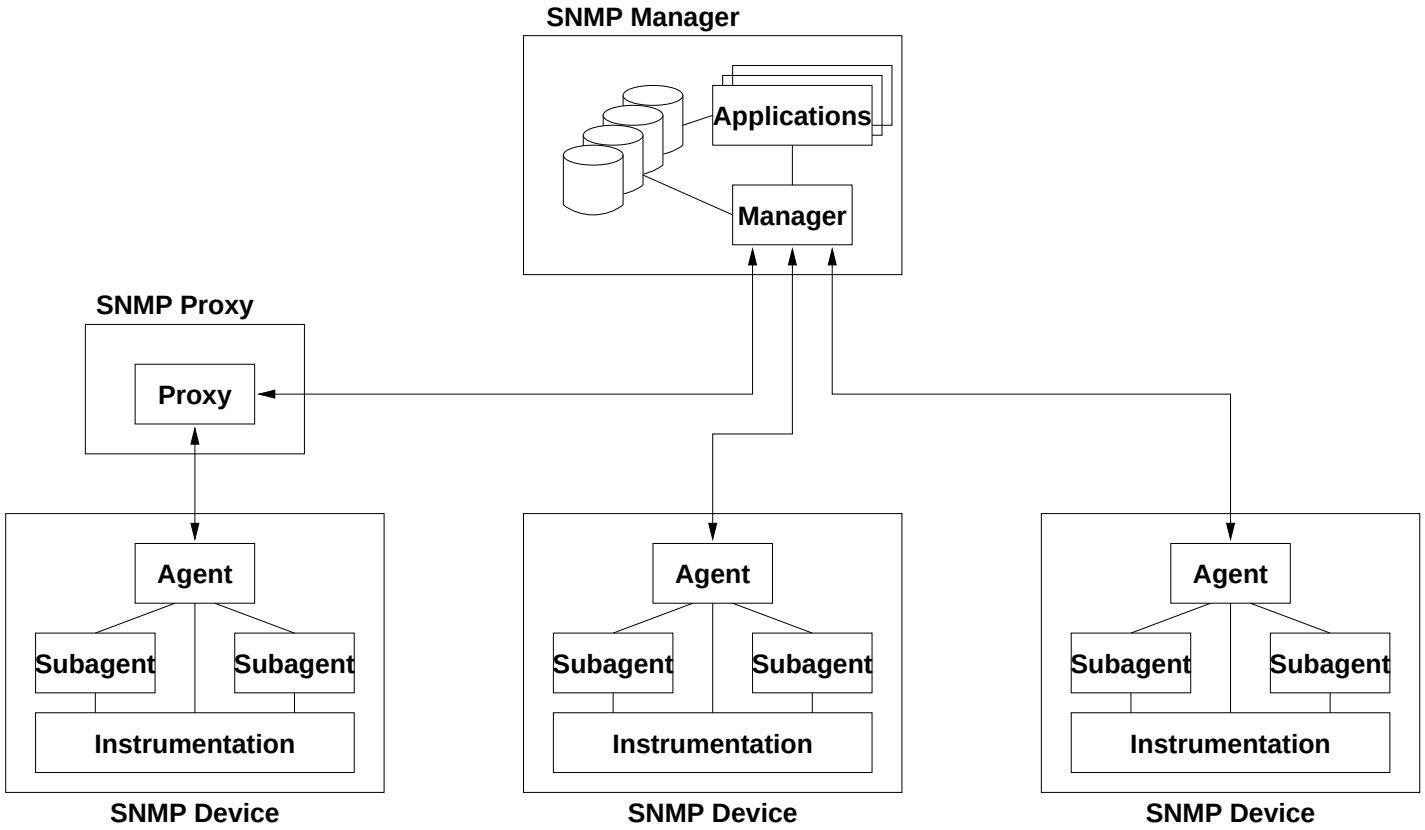
Naming and Addressing

- Naming of Information
  - Identification of a managed object within a naming scope
  - Names may be scoped by a protocol engines (which is identified by an address)
  - Globally unique names may be constructed from the pieces
  - Definition of the naming system is an essential design step
- Addressing of Protocol Engines
  - Use (extended?) transport addresses to identify protocol engines
  - Introduce a new address space for the protocol engines
  - Uniform Resource Identifiers (URIs)

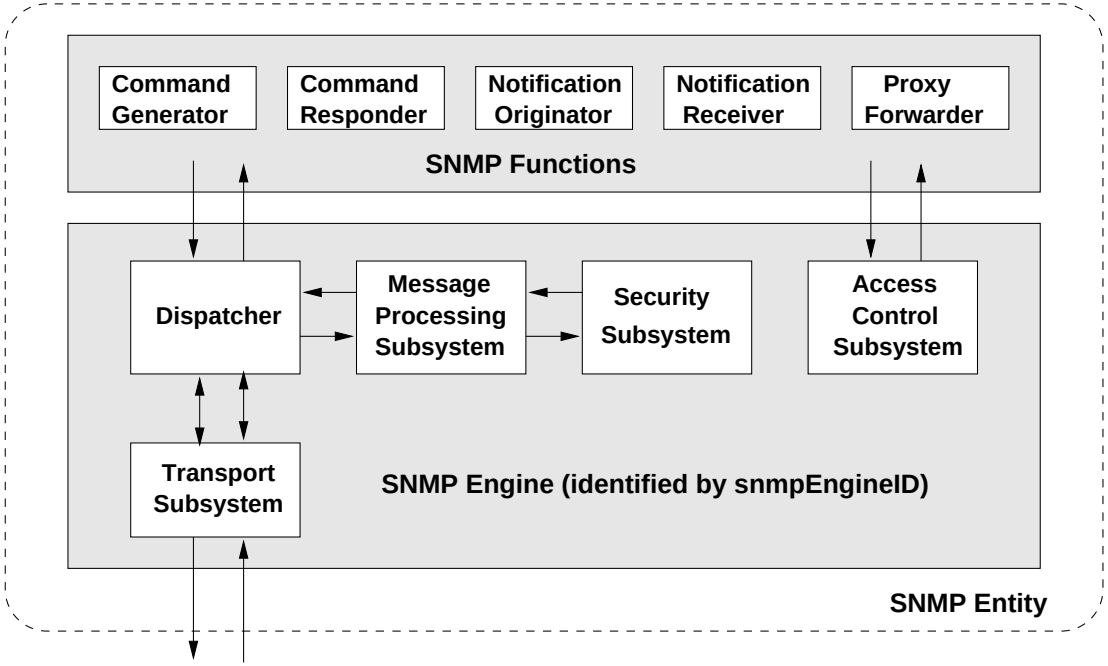
Security and Access Control

- Message-based security
  + Self-contained solution
  + Reduced dependencies on the infrastructure
  - Complex specification and verification
  - Complex implementation and costs
- Session-based security
  + Leveraging existing work
  + Better integration with a security infrastructure
  - Dependency on a security infrastructure
- Explicit access control rules vs. implicit access control vs. no access control

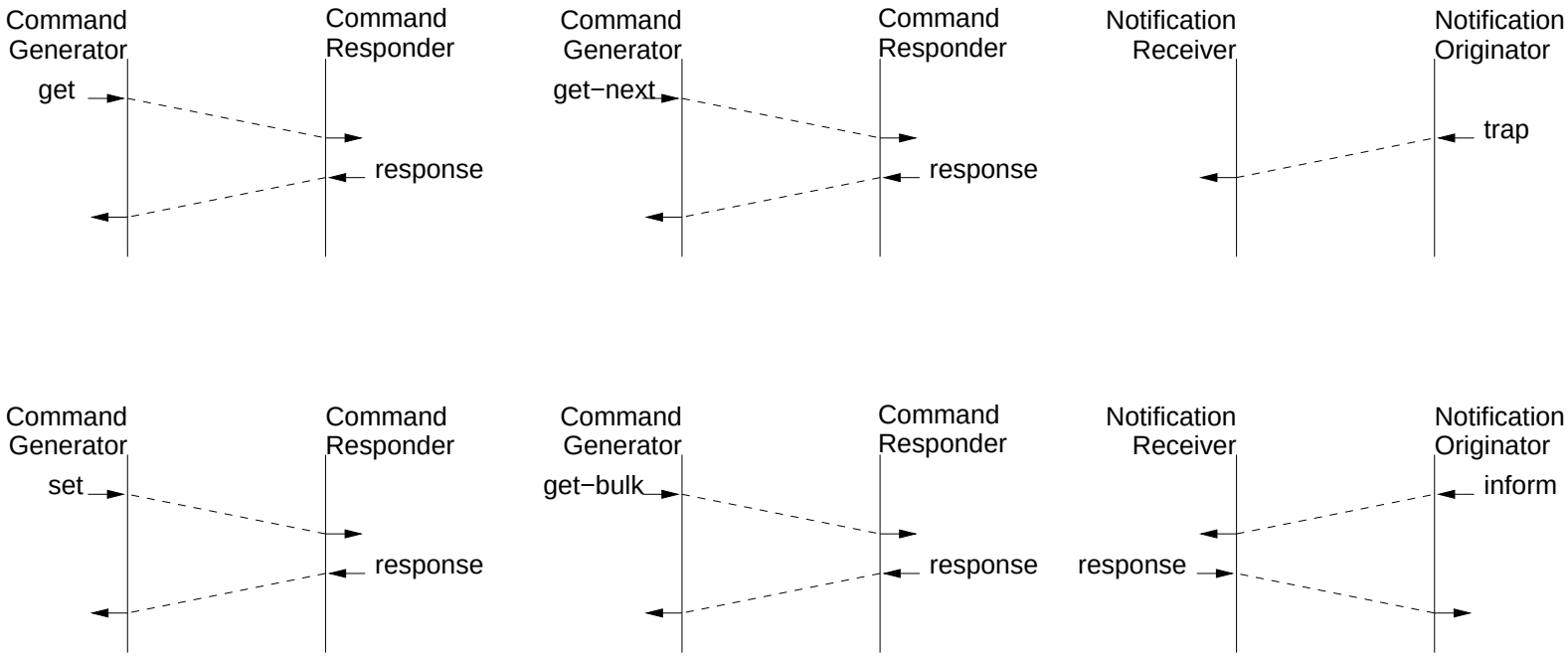
Simple Network Management Protocol (SNMP)



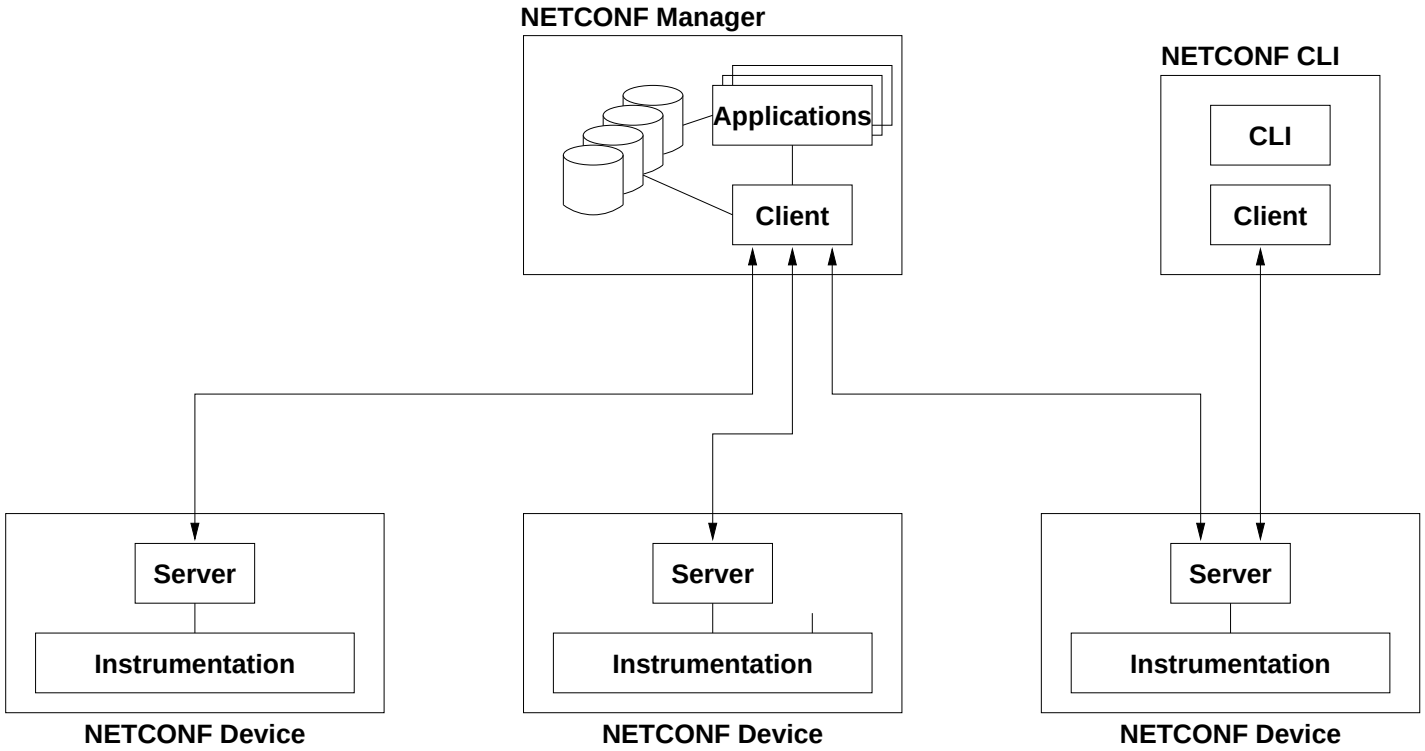
Simple Network Management Protocol (SNMP)



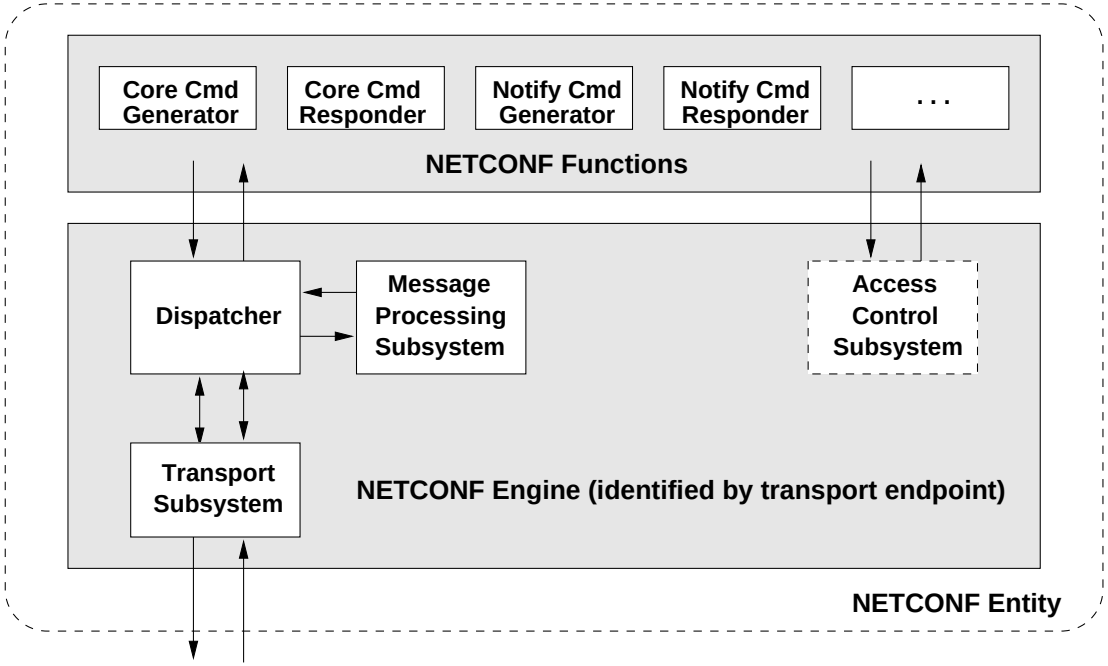
Simple Network Management Protocol (SNMP)



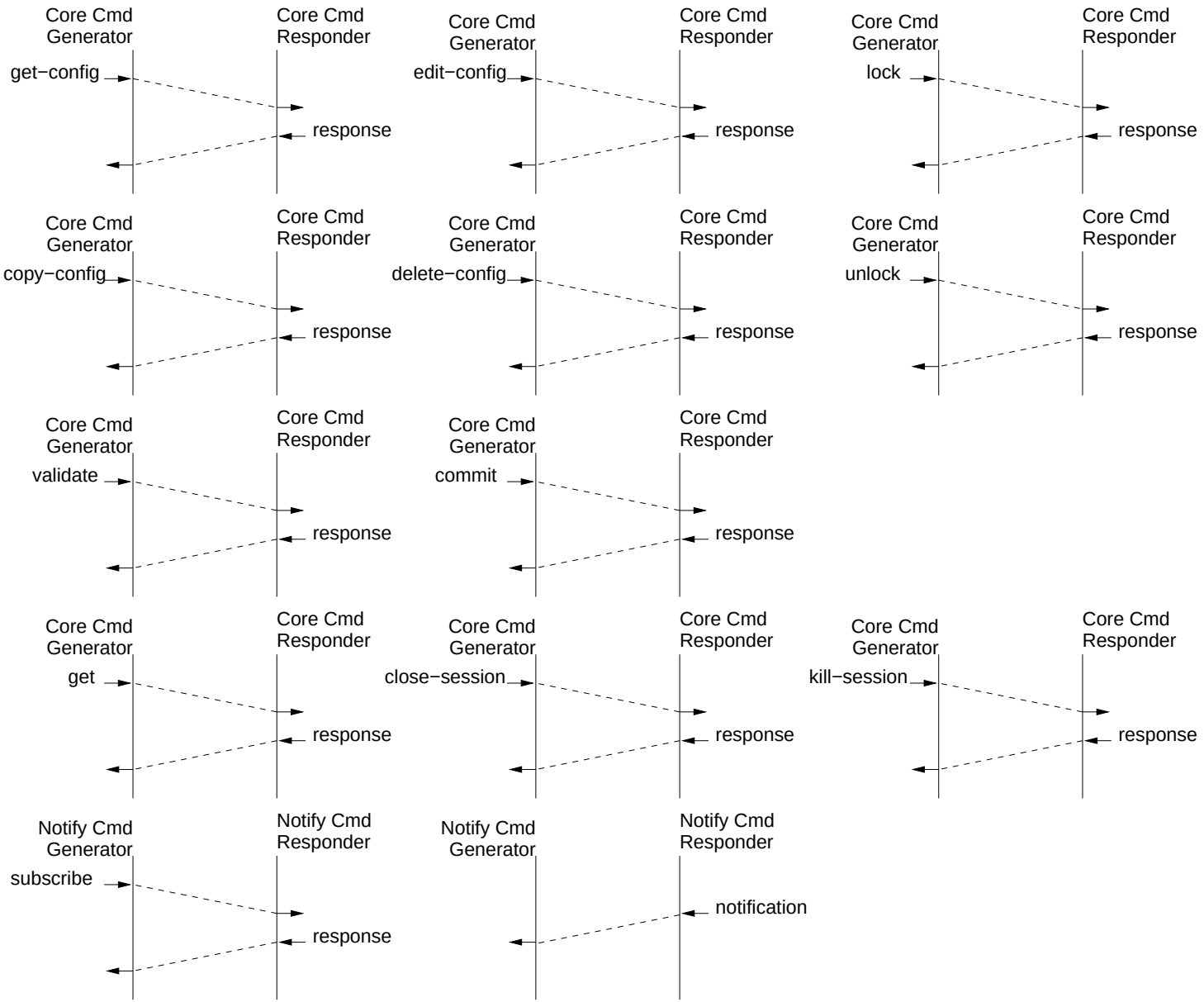
Network Configuration Protocol (NETCONF)



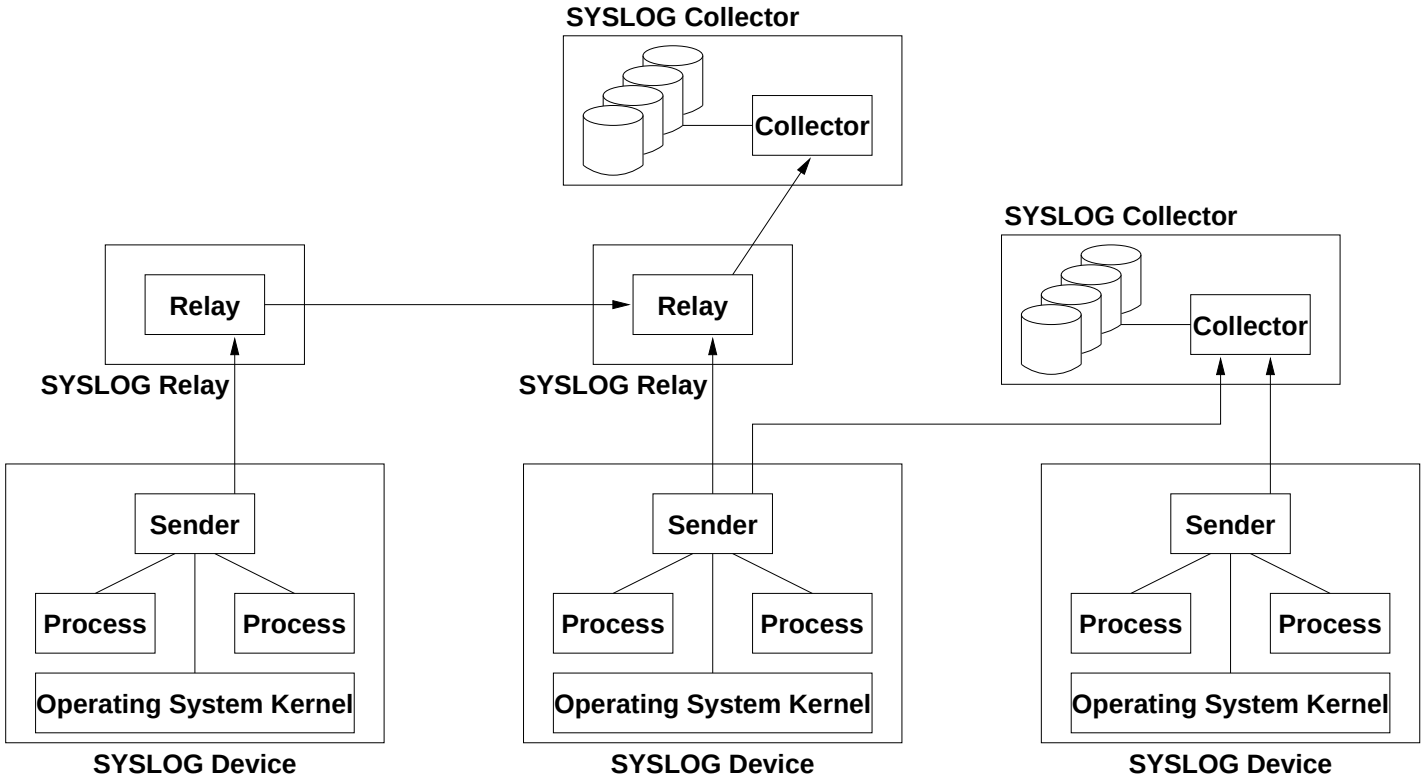
Network Configuration Protocol (NETCONF)



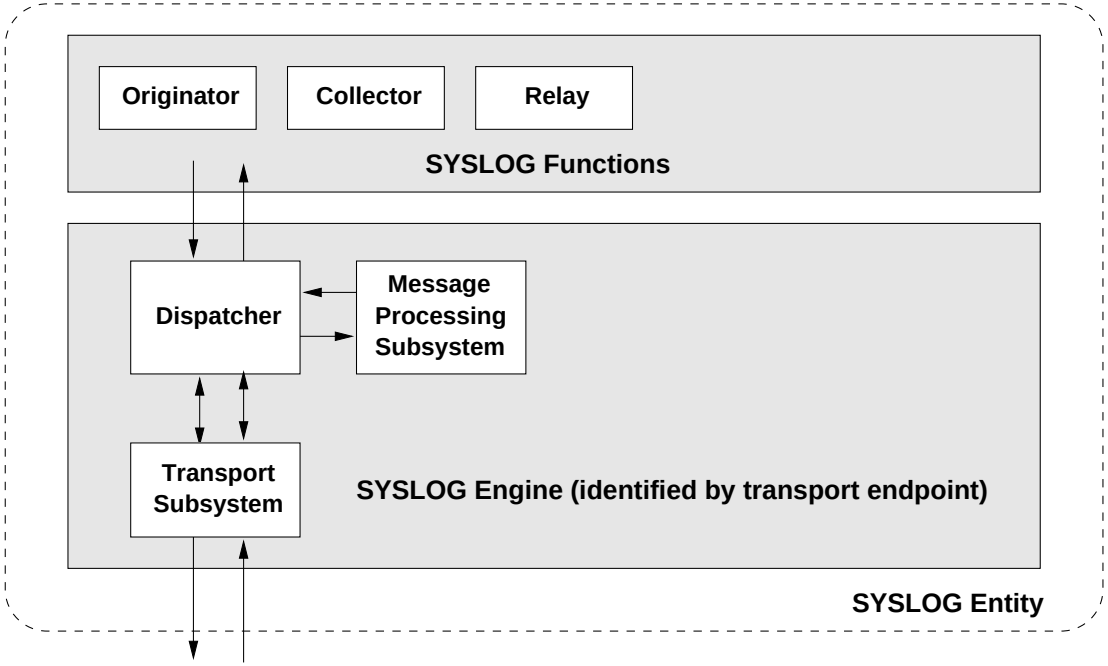
Network Configuration Protocol (NETCONF)



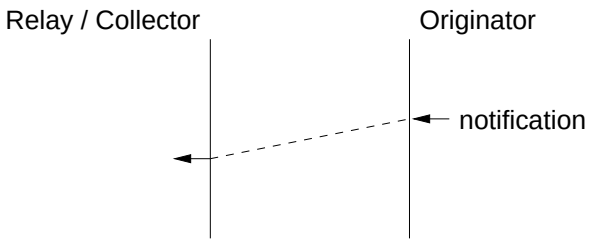
System Logging Protocol (SYSLOG)



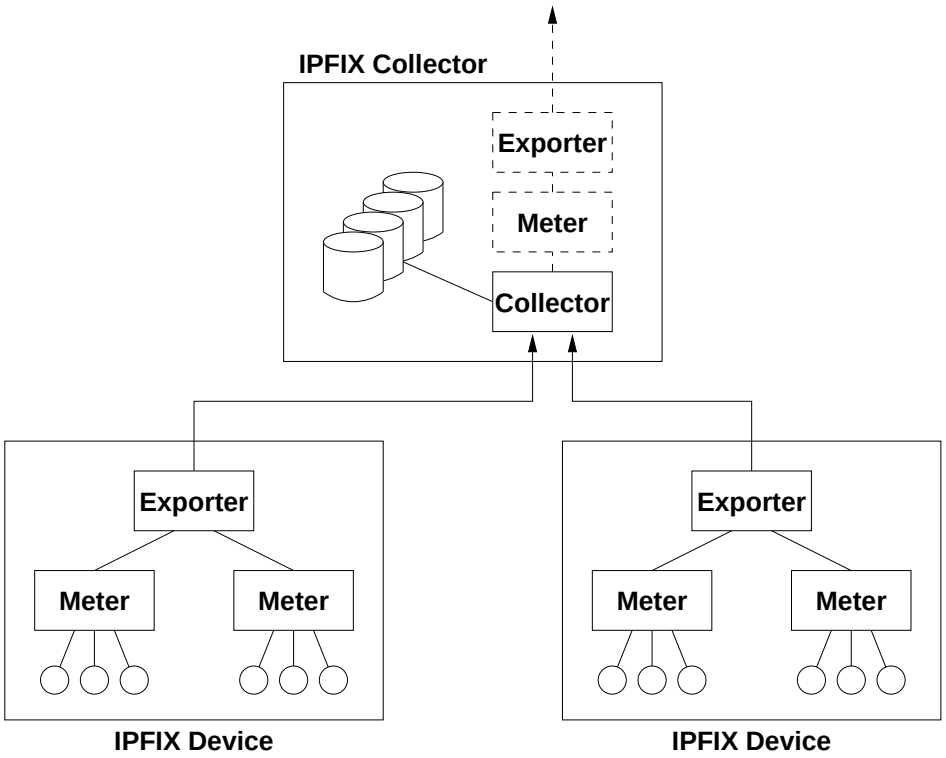
System Logging Protocol (SYSLOG)



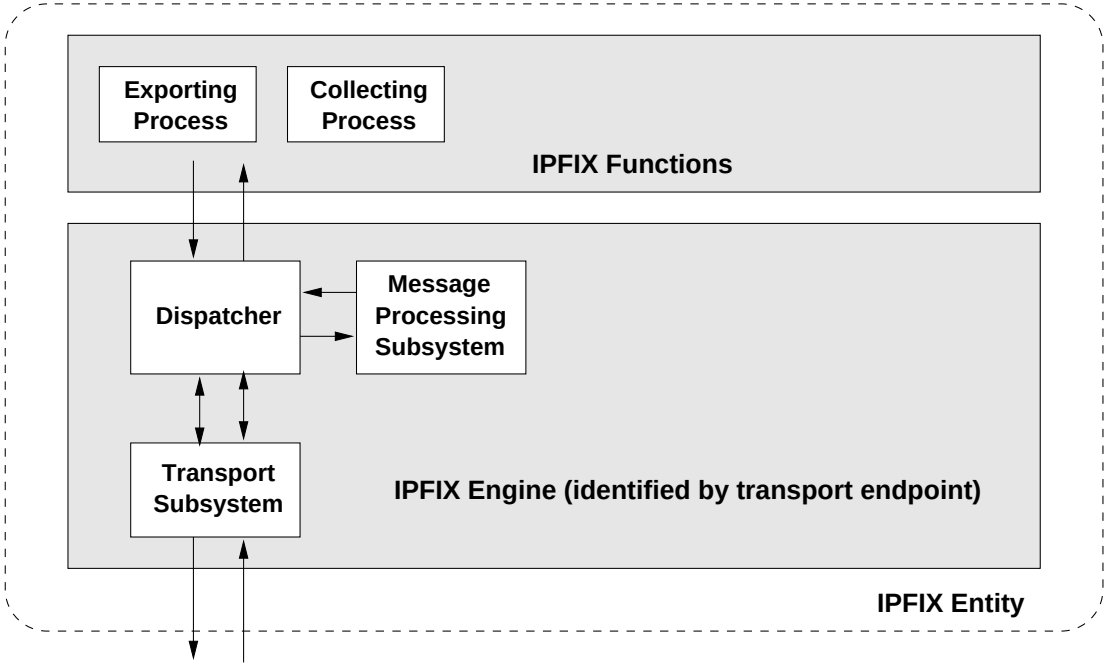
System Logging Protocol (SYSLOG)



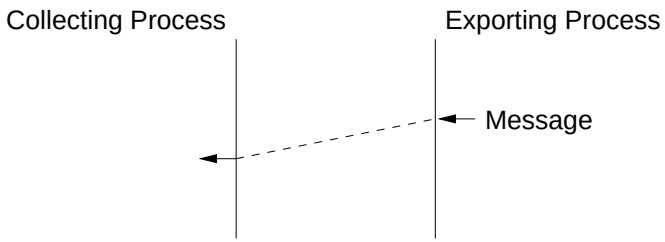
Flow Information Export Protocol (IPFIX)



Flow Information Export Protocol (IPFIX)



Flow Information Export Protocol (IPFIX)



References

[1] J. Schönwälder. *Handbook of Network and System Administration*, chapter Internet Management Protocols. Elsevier, 2007.

[2] D. Harrington, R. Presuhn, and B. Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. RFC 3411, Enterasys Networks, BMC Software, Lucent Technologies, December 2002.

[3] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and Applicability Statements for Internet Standard Management Framework. RFC 3410, SNMP Research, Network Associates Laboratories, Ericsson, December 2002.