INF 5071 – Performance in Distributed Systems

Further Protocols with/-out QoS support

3/10 - 2008

Interactive applications

Interactive applications

- Main examples today
  - Multiplayer games
  - Audio streams
    - Audio conferencing, IP telephony
  - Signaling
    - RTSP for video stream control, SIP for 3G telephone dialing, ...
- Others
  - Remote surgery
  - Robot control
  - Sensing
    - Sensing voice, temperatures, movement, light, ...
  - Bank transactions
  - ...

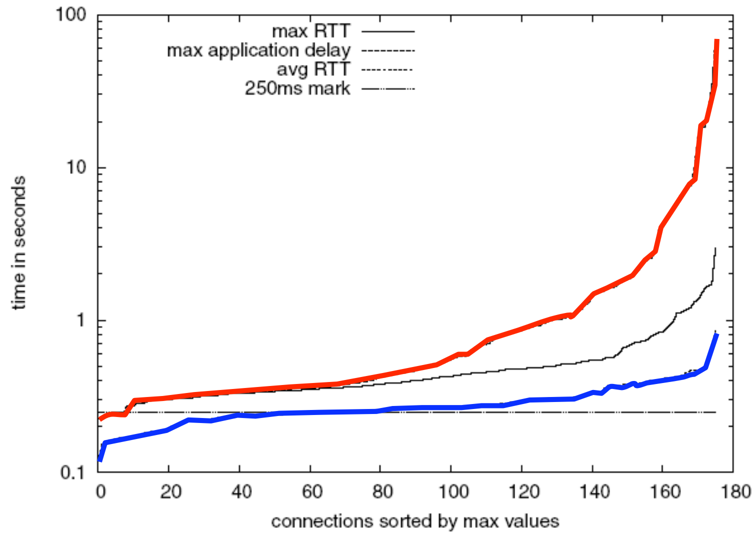
Thin stream applications

Application	Average payload size (byte)	Packet interarrival time (ms)	Bandwidth requirements (bps)
Anarchy Online	93	909	1757
Counterstrike	142	81	19764
Skype	111	30	37906
CASA (radar control)	175	7287	269
Windows remote desktop	111	318	4497
MPEG-2 streaming	1460	3	~4200000

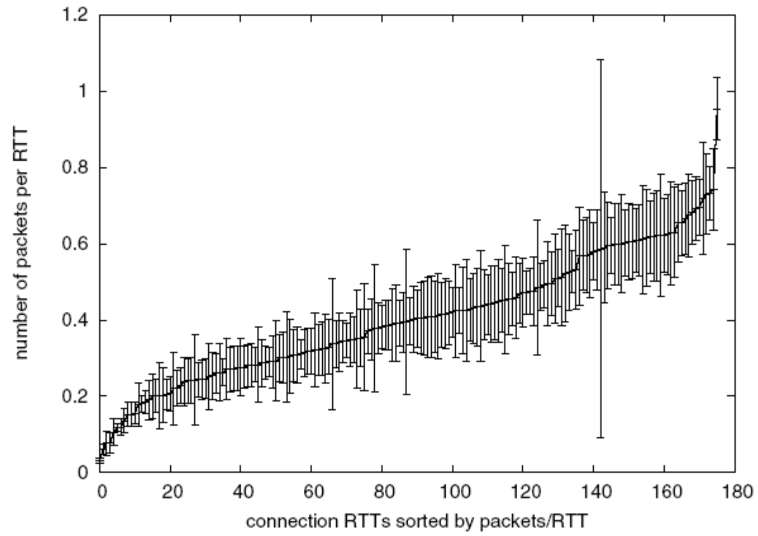
- Analysis of traces for several applications show thin-stream properties
  - Small packets
  - High packet interarrival-time

Thin Streams

- Transport protocols being developed for throughput-bound applications
- BUT, there exist several **low-rate, time-dependent** applications
- Anarchy Online MMORPG Case Study



(a) RTT versus maximum application delay

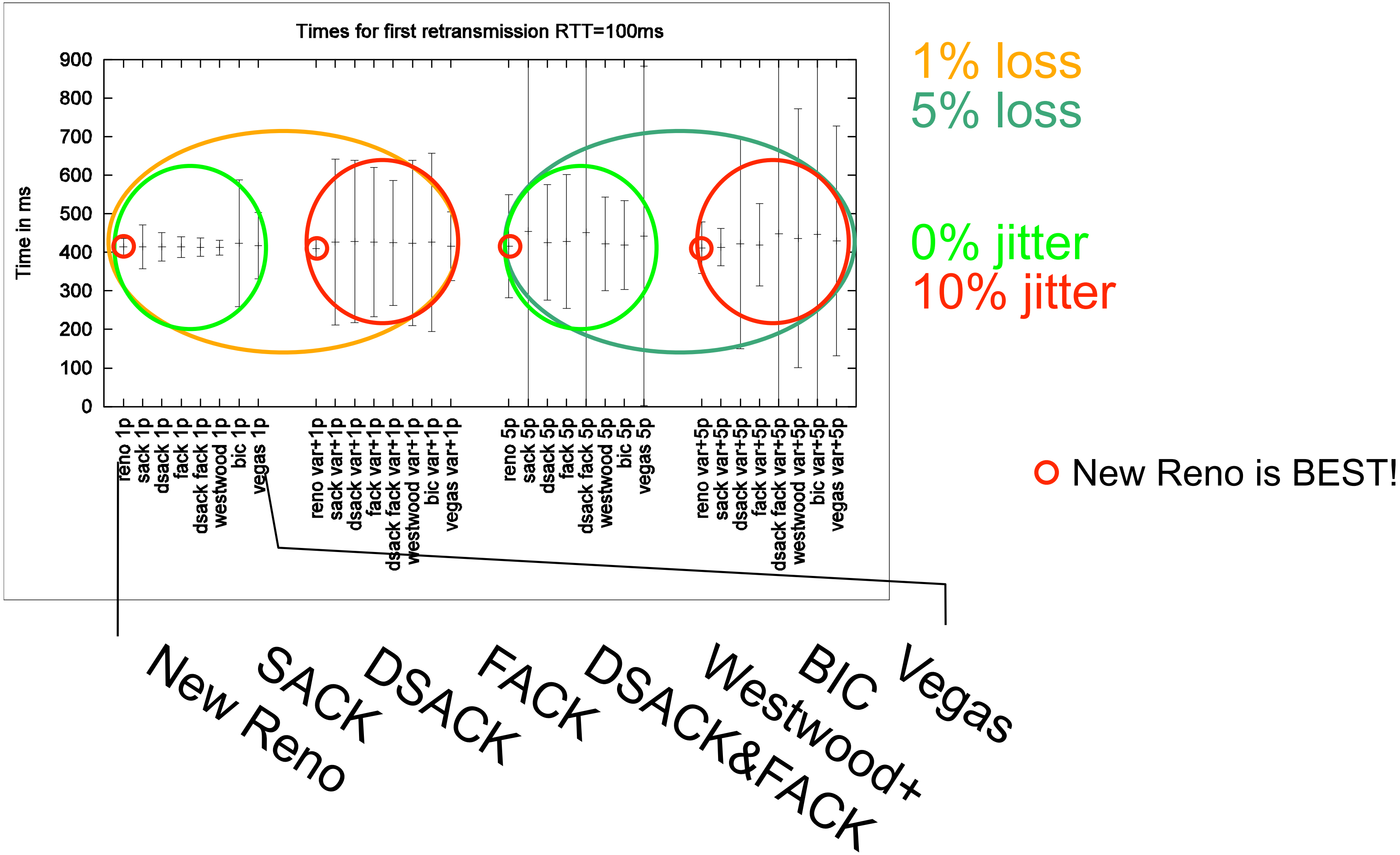


(b) Packets per RTT with standard deviation

- average delay: **~250 ms**
- max delay: **67 seconds (6 retransmissions)**
- packets per second: **< 4 (less then one per RTT)**
- average packet size: **~93 bytes**
- average bandwidth requirement: **~1.8 Kbps**

TCP 1st retransmission

Times of first retransmission, RTT=100 ms



Stream Control Transmission Protocol (SCTP)

- Stream Control Transmission Protocol
  - RFC2960, IETF Standards Track
  - RFC2719, Architectural Framework for Signaling Transport
  - SCTP Unreliable Data Mode Extension (draft-ietf-tsvwg-usctp-00.txt)

- Initial goal
  - Signaling protocol for SS7 transport over IP networks
    - Protocol of the telephony world for IP telephony
  - Supposed to address low latencies
    - “require response between 500 – 1200 ms” … or “initiation of error procedures” [RFC 2719]

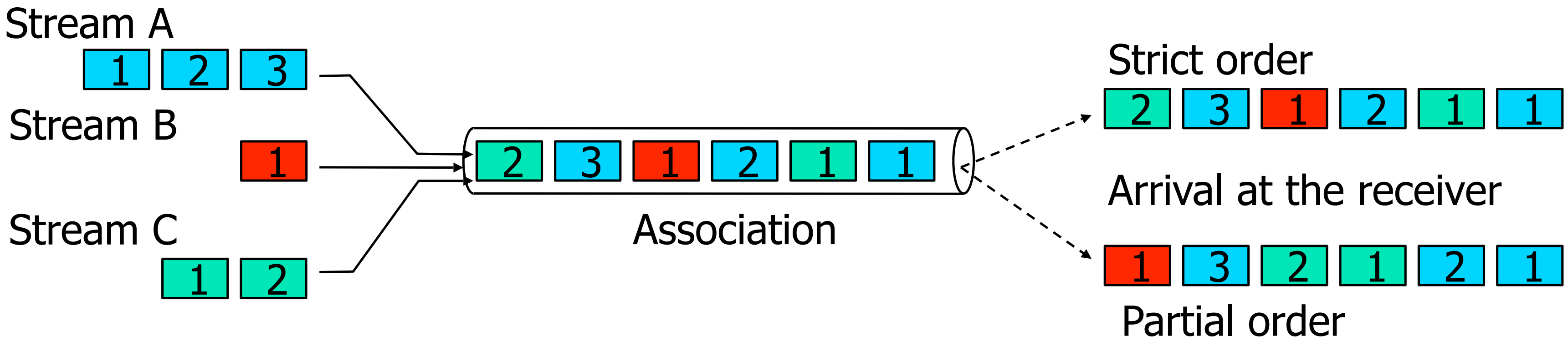
  - Supporters
    - Motorola, Cisco, Siemens, Nortel Networks, Ericsson, Telcordia, UCLA, ACIRI

SCTP Features

	UDP	DCCP	TCP	SCTP
Connection-oriented service		X	X	X
Connectionless service	X			
Ordered			X	X
Partially Ordered				X
Unordered	X	X		X
Reliable			X	X
Partially Reliable				X
Unreliable	X	X		X
With congestion control		X	X	X
Without congestion control	X			
Multicast support	X	X		
Multihoming support				X

Association and Streams

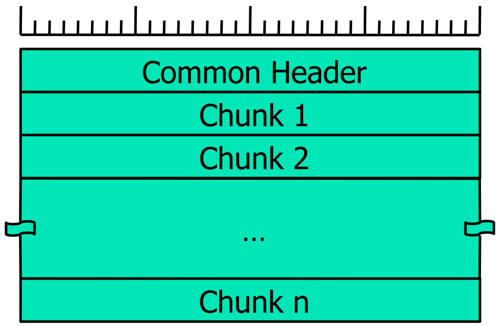
- Reliable data transfer
  - Confirmed, no duplicates, error-free
  - Several **streams** in one **association**

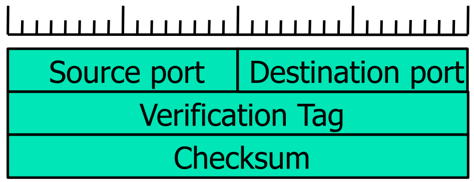


- Strictly ordered delivery
  - keep order within and among streams of an association
  - data transmission stalled if one stream is stalled
- or Partially ordered delivery
  - keep order within a stream of an association
  - transmission for non-stalled streams can continue

Message format

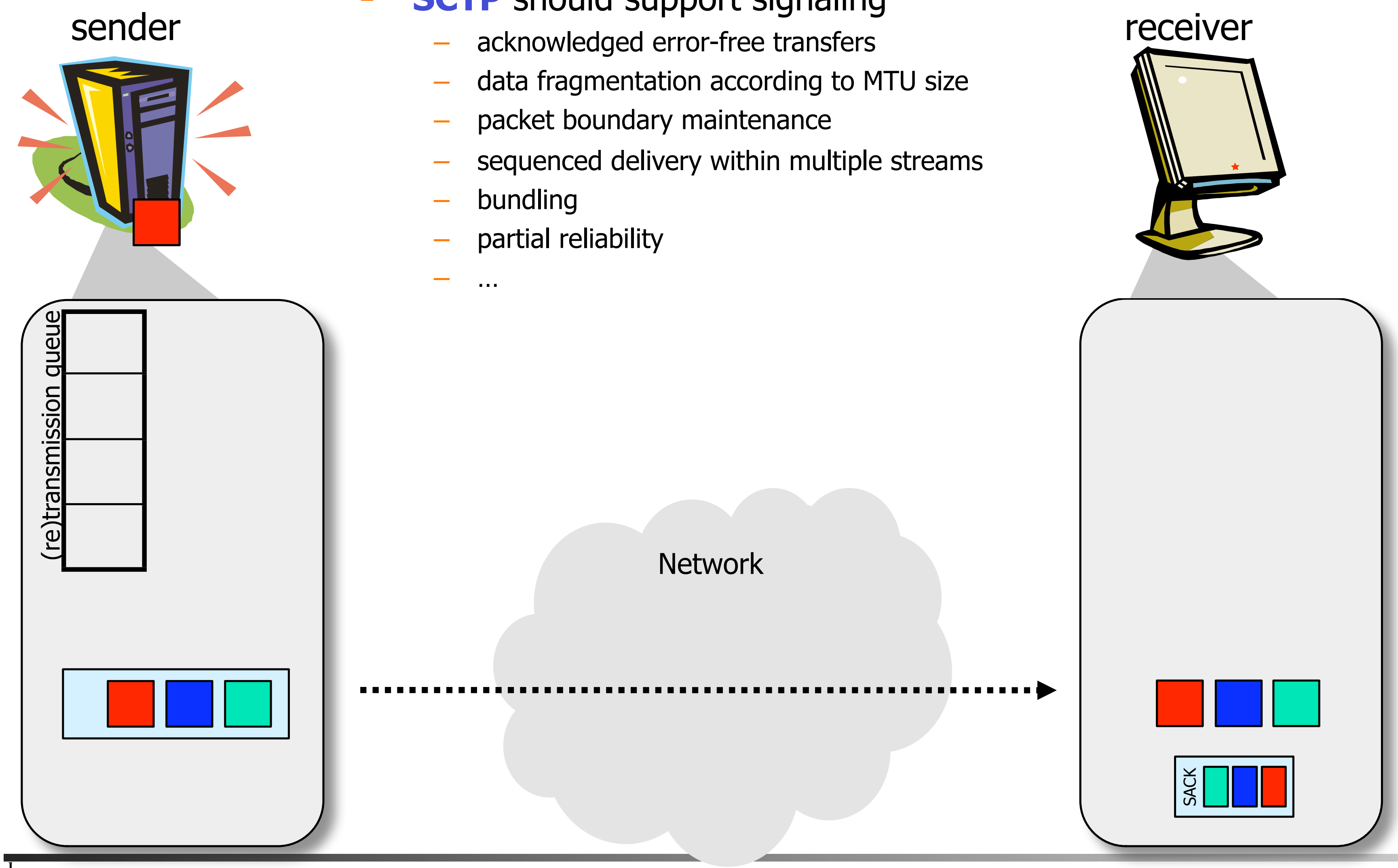
- Multiplexing of several user messages
- One user message: “Chunk”
- Chunk Bundling
  - Chunk: part of an SCTP packet belonging to a single stream
- That means
  - one application “write” is a chunk
  - one application “read” returns a chunk
  - but several chunks in a single IP packet



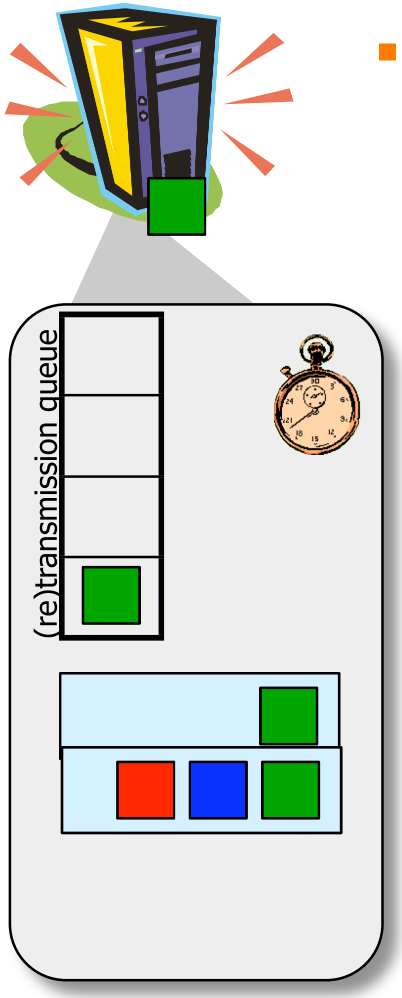


Stream Control Transmission Protocol

- **SCTP** should support signaling
  - acknowledged error-free transfers
  - data fragmentation according to MTU size
  - packet boundary maintenance
  - sequenced delivery within multiple streams
  - bundling
  - partial reliability
  - ...



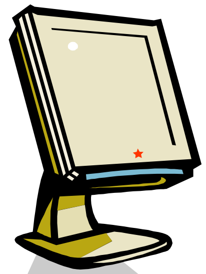
Retransmission by Time-Out

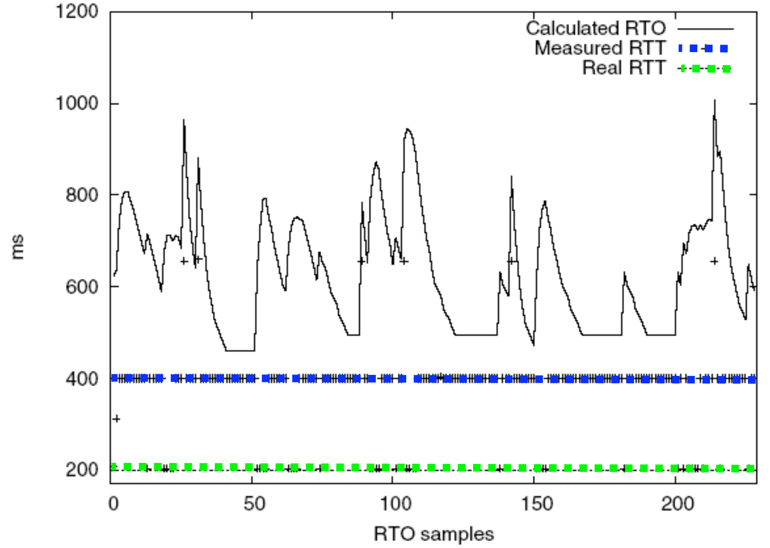


- Timeout is dependent on
  - minRTO = 1000 ms
  - estimated RTT based on SACKs
    - BUT SACKs are delayed
      - one ACK for two packets or
      - 200 ms timer
    - influences estimated RTT, especially for thin streams
    - RTO value grows

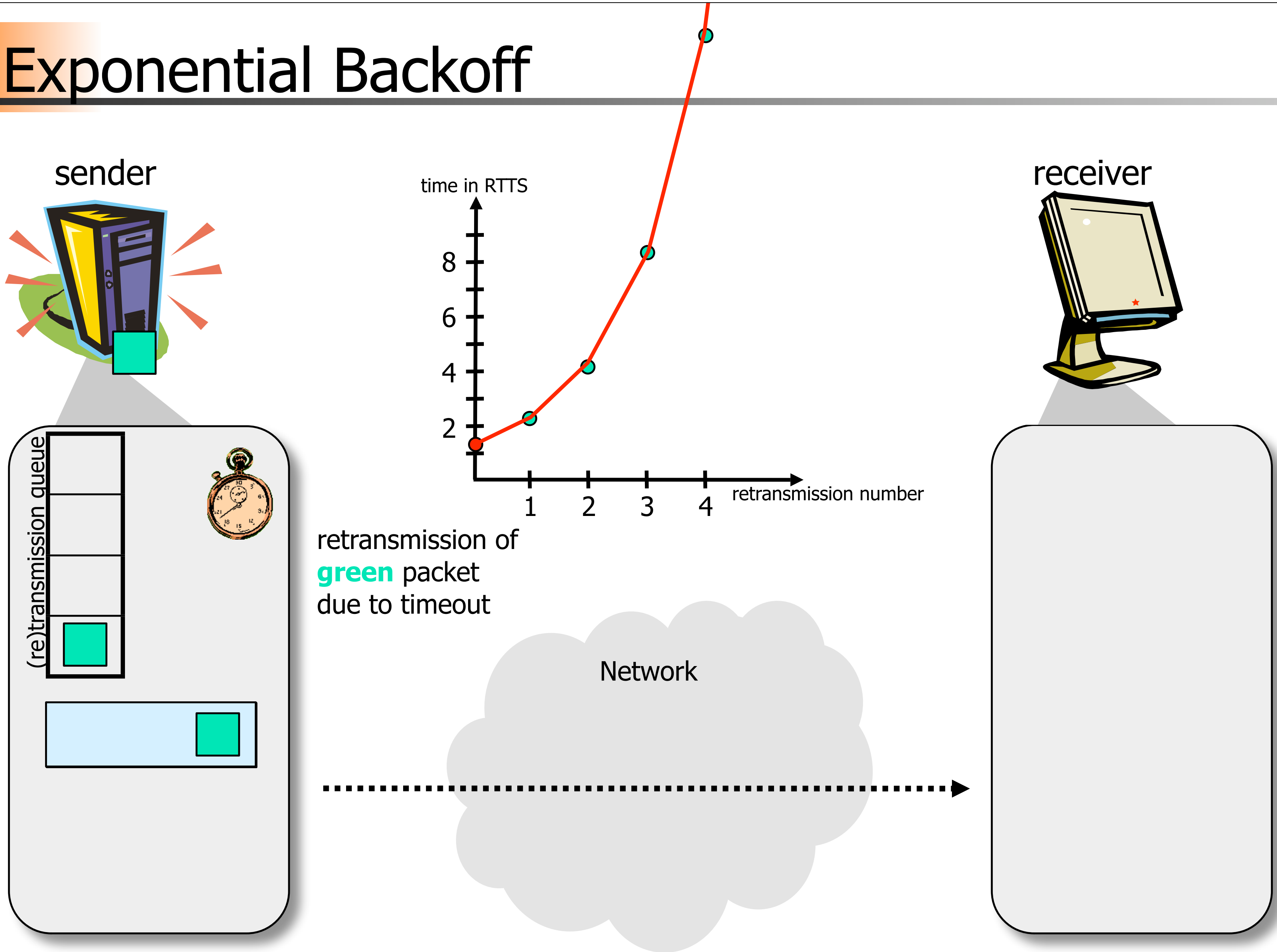
retransmission of packet with **green** chunks due to timeout

Network

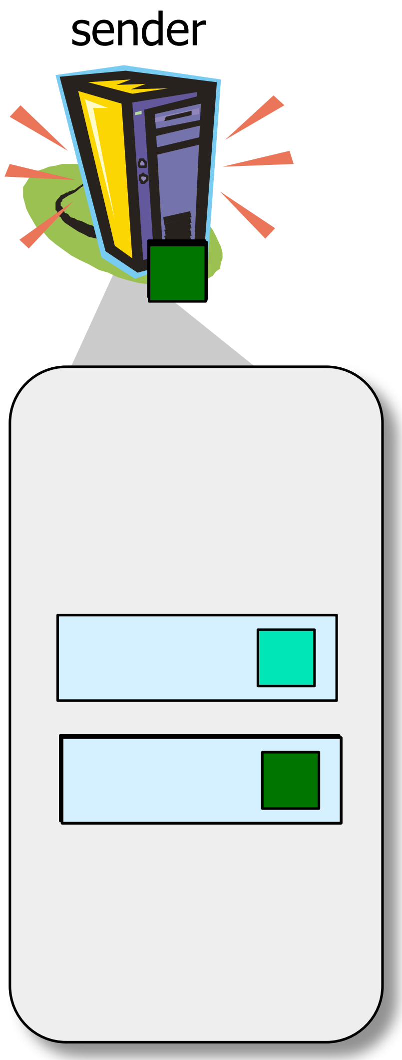




Exponential Backoff



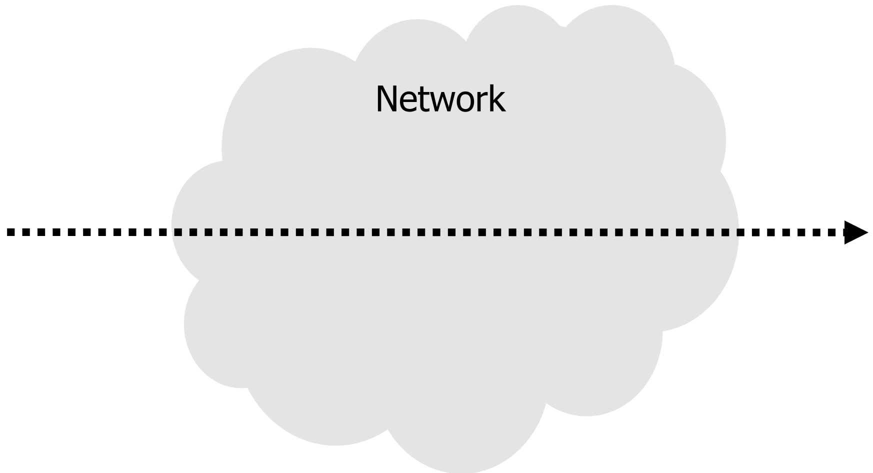
Retransmission by Fast Retransmit

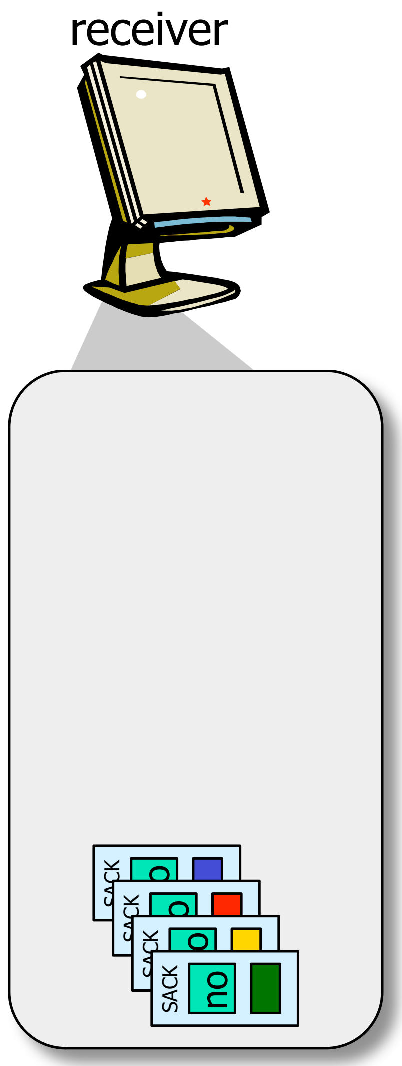


4 SACKs needed for fast retransmit

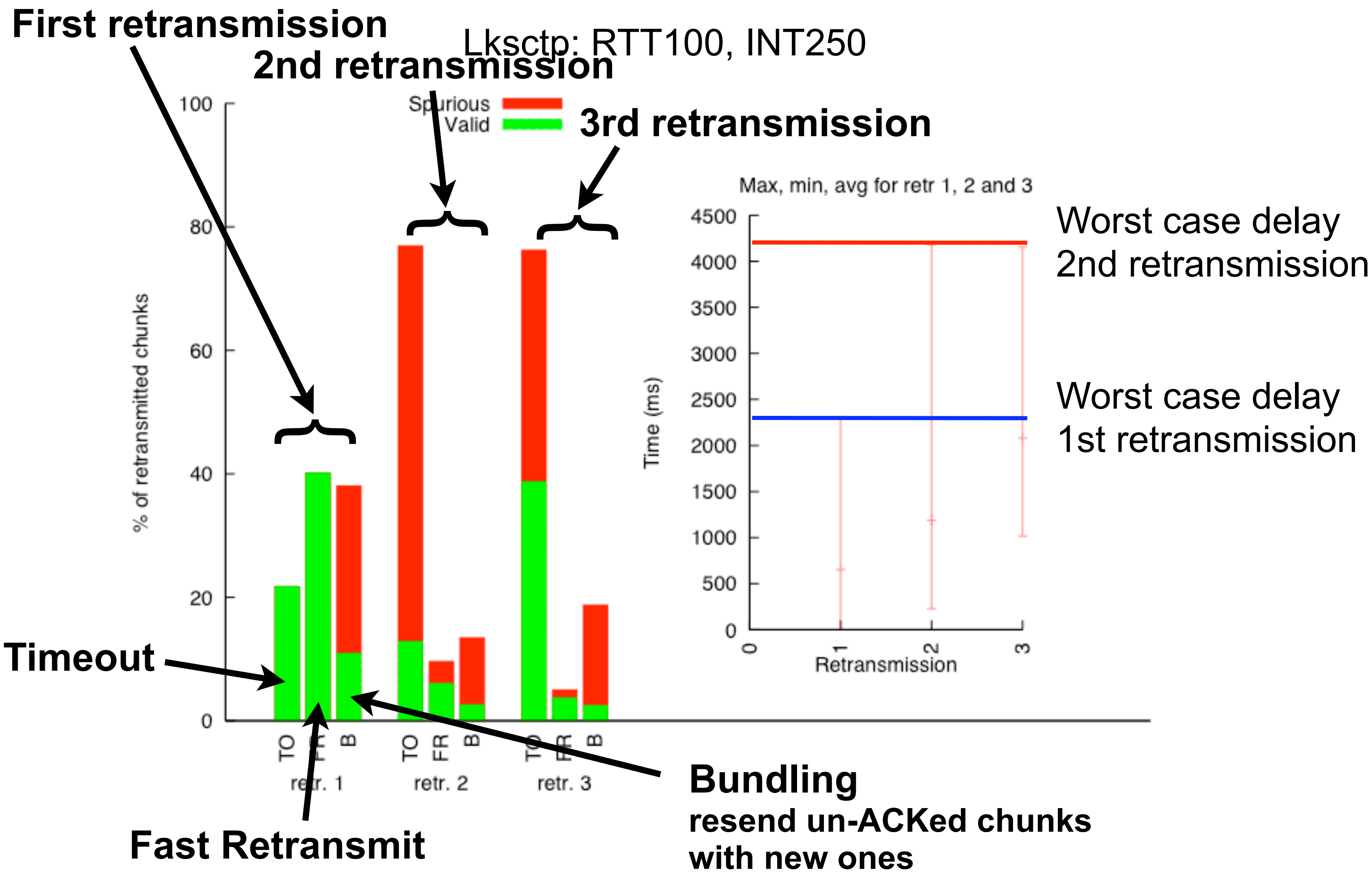
\+ thin streams

= "all" retransmissions due to timeouts





lksctp performance



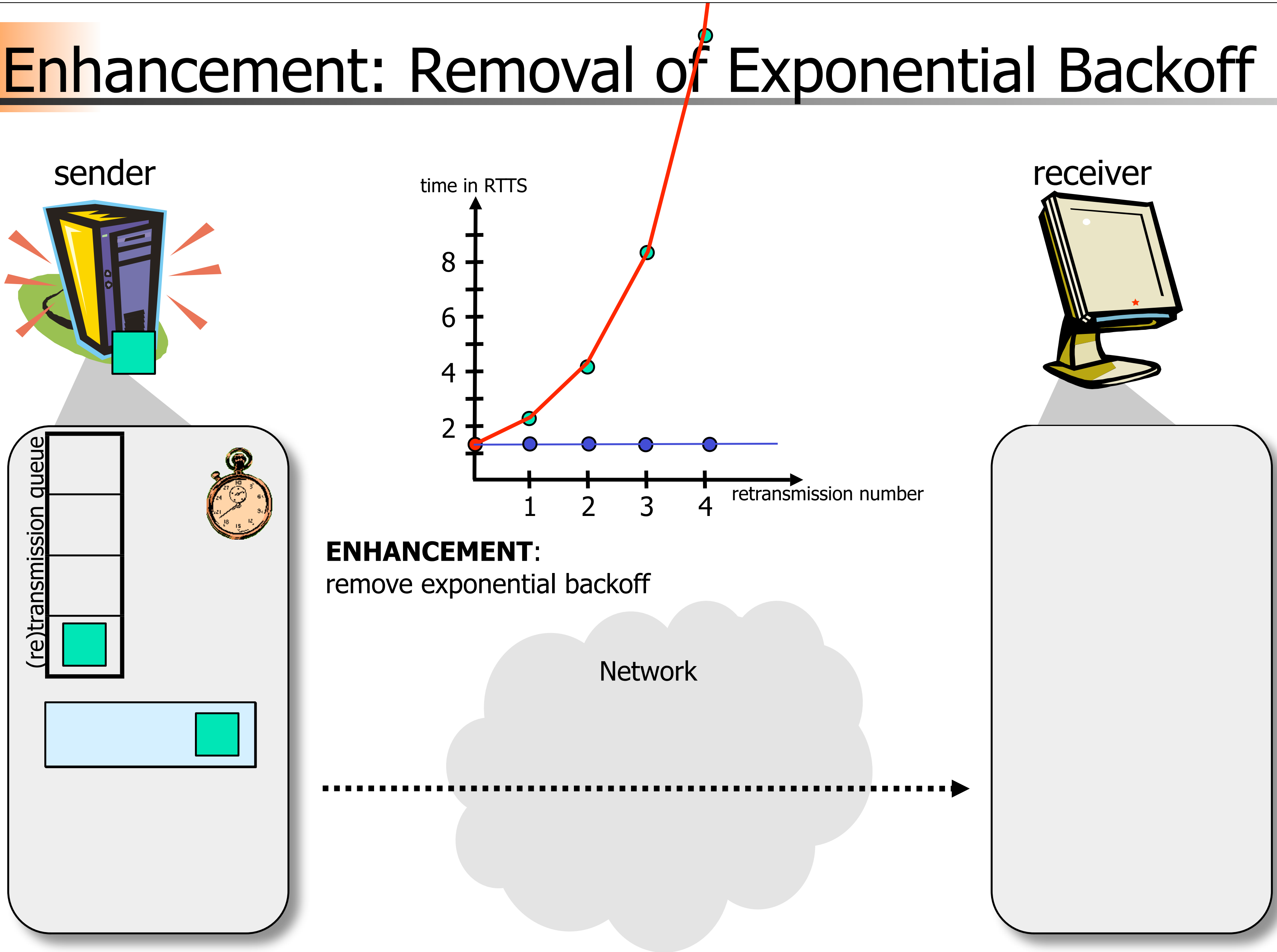
Improvement idea

- Figure out when a stream suffers
  - When it is a "Thin Stream"?

  - Whenever so few packets are in-flight that a fast retransmit can not be triggered
  - Then the sender can only wait until RTO (retransmission timeout) and perform a timeout retransmission

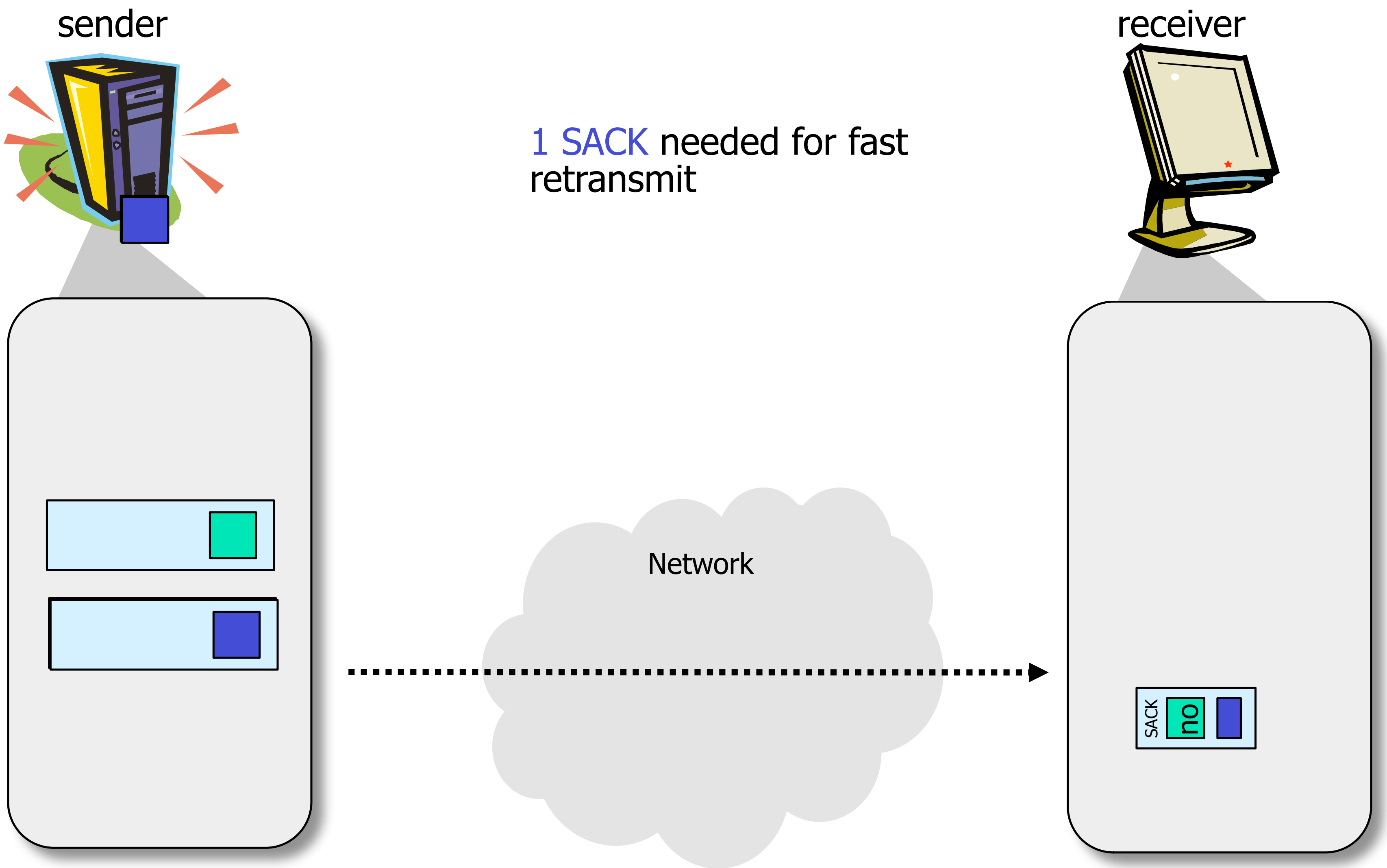
- Then switch on changes
  - No exponential backoff
  - Faster retransmit
  - Minimum retransmission timeout

Enhancement: Removal of Exponential Backoff

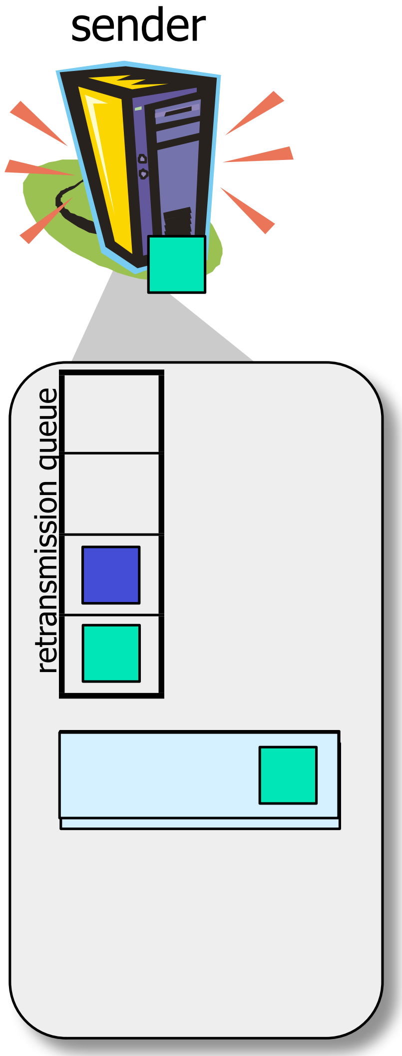


ENHANCEMENT:
remove exponential backoff

Retransmission by Faster Retransmit



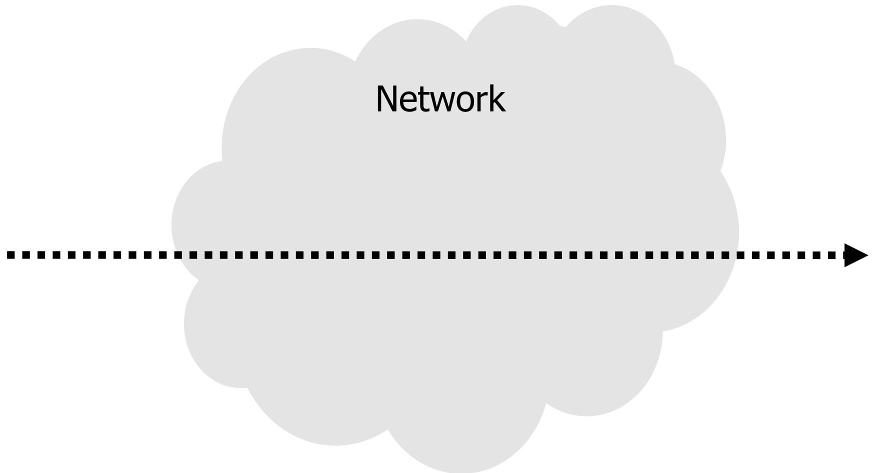
Enhancement: Fast Retransmit Bundling

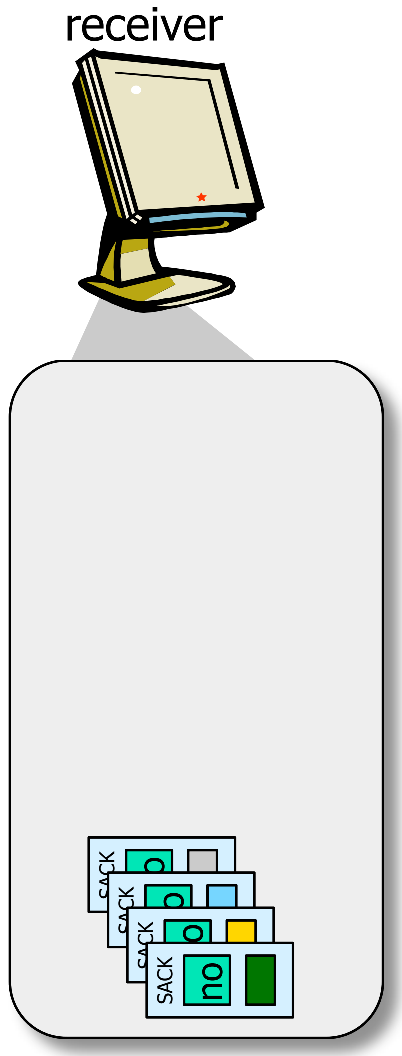


ENHANCEMENT:
piggyback all chunks in retransmission queue

retransmission of **green** packet (chunks) due to **dupACKs**

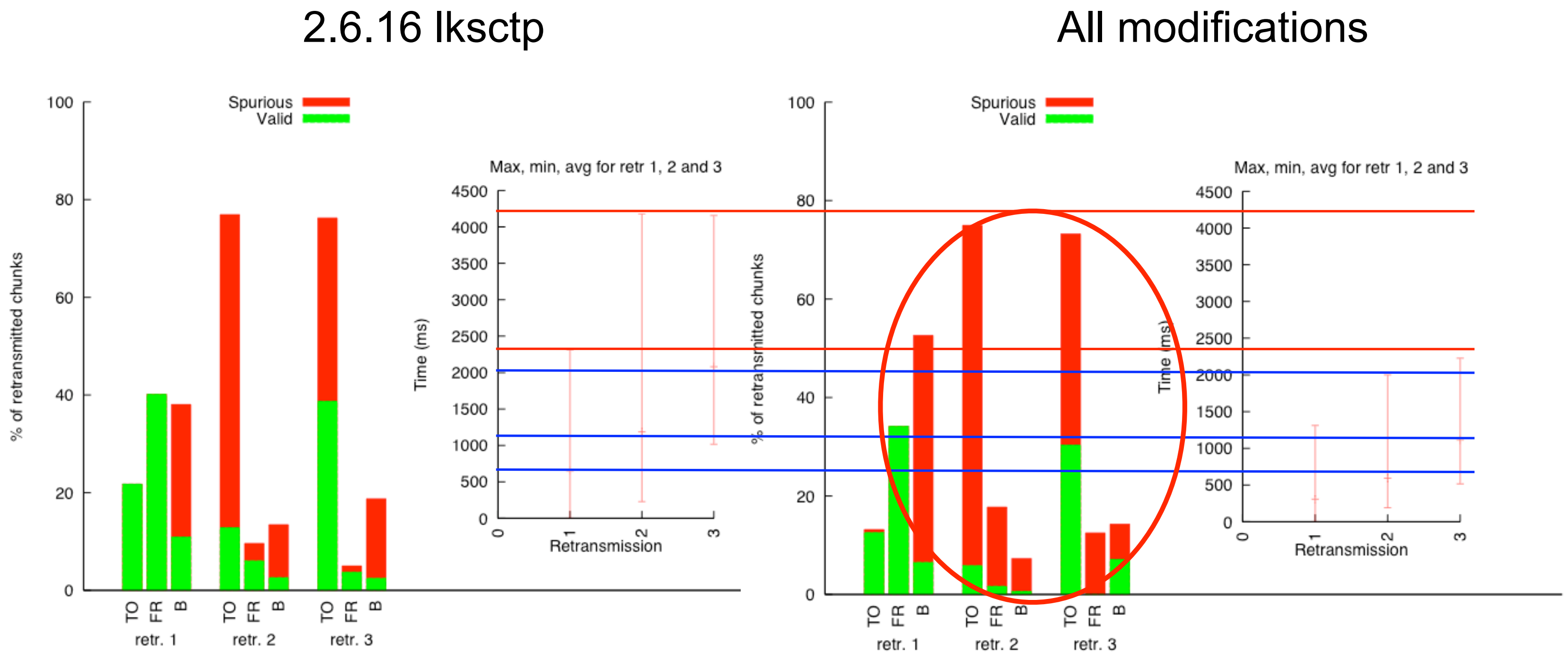
CURRENT IMPLEMENTATIONS:
blue packet is NOT piggybacked when **dupACKs**
(but would be if due to timeout)





lksctp performance

RTT100, INT250



☺ Large reduction in maximum and average latency

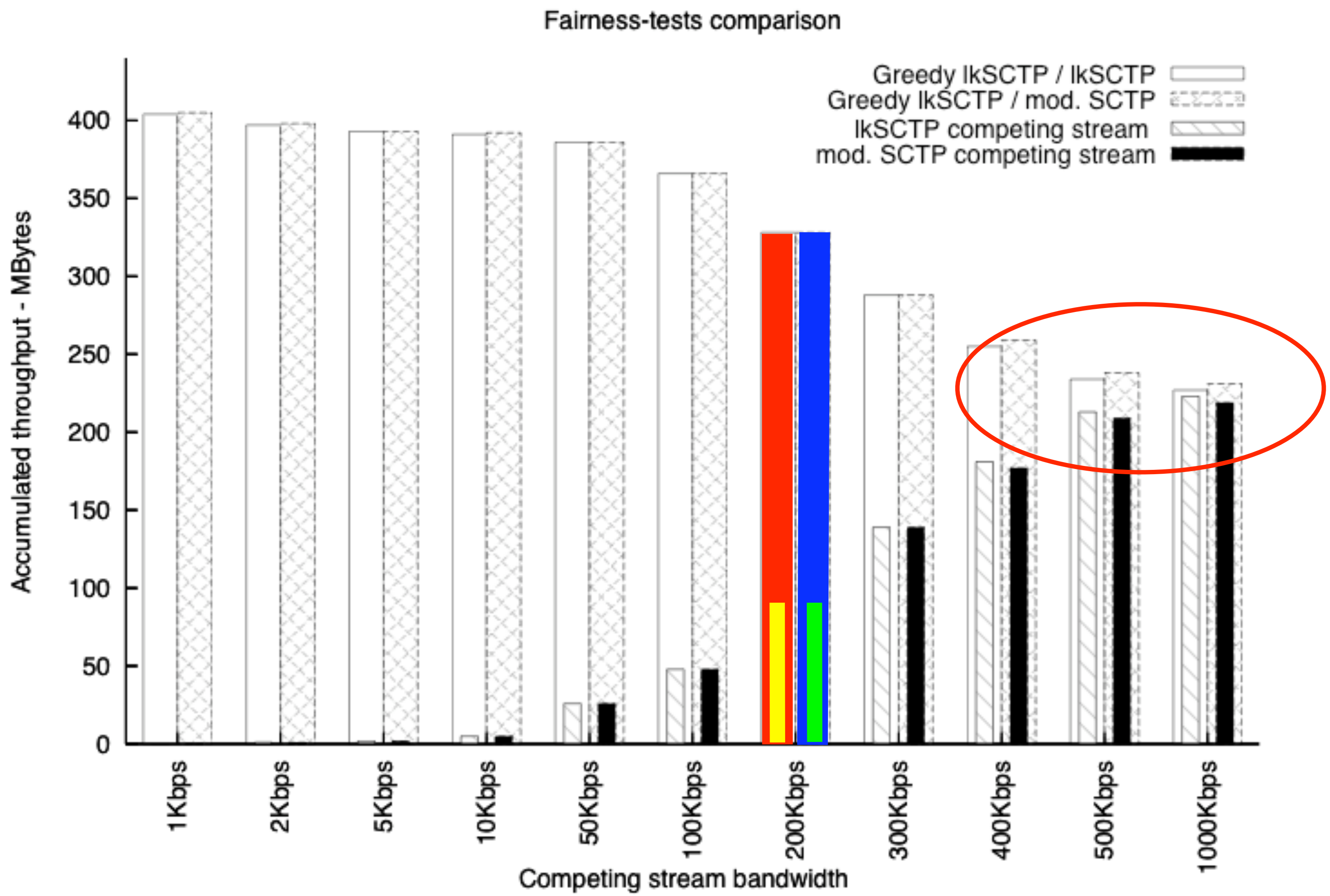
☹ An increase in spurious retransmissions

-Tolerable due to the low datarate

Fairness considerations and tests

- Modifications increases aggressiveness of stream
  - Exponential back-off
  - Fast retransmit
  - Minimum retransmission time out
- We want to test whether fairness is in jeopardy

Fairness considerations and tests



The same for TCP?

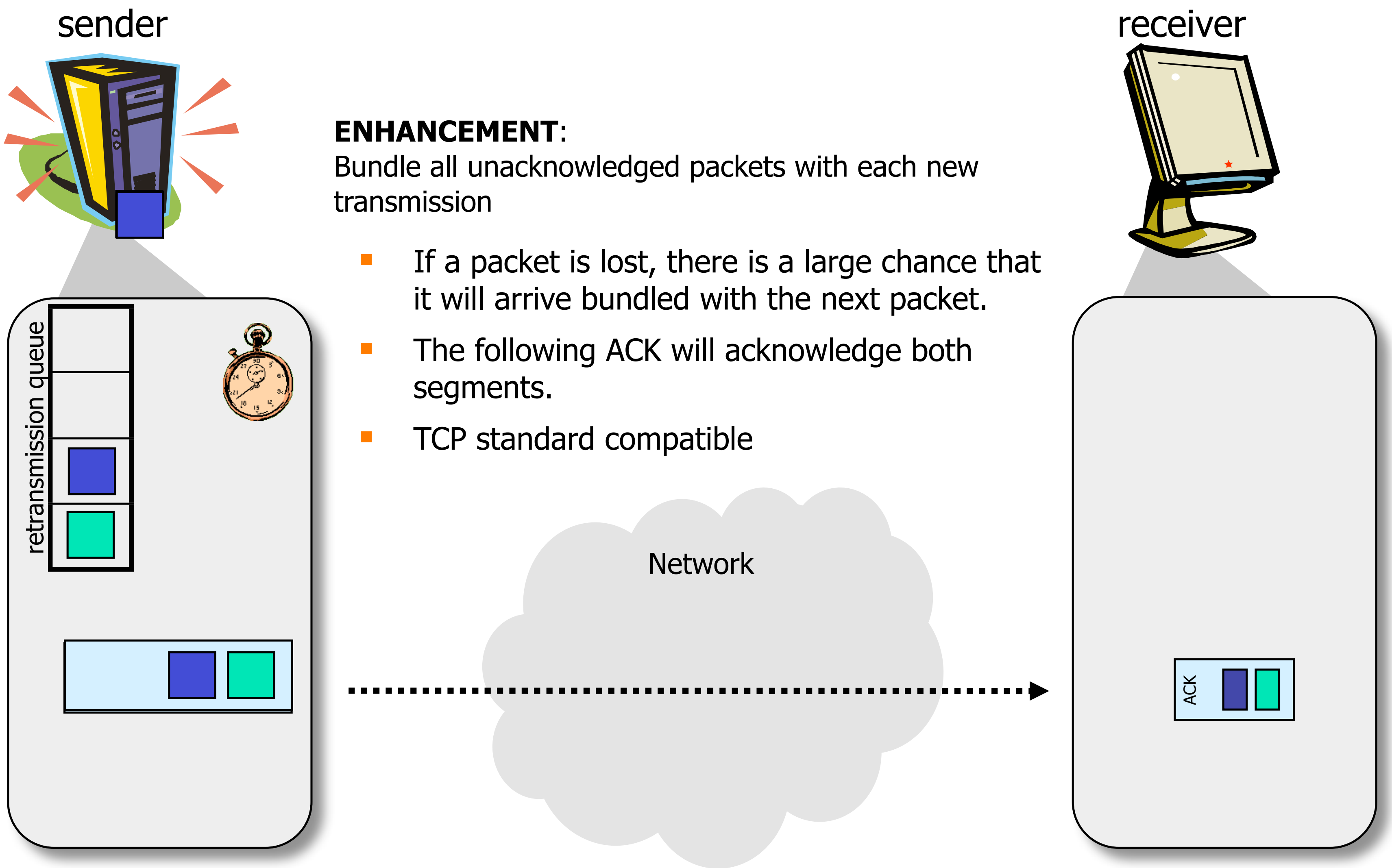
- Useful for TCP as well?
  - TCP uses fast retransmit
    - 3 instead of 4 ACKs needed
  - TCP uses timeout retransmit
    - minRTO lower than 1000ms (usually around 200ms)
  - TCP uses delayed acknowledgements
    - some implementations, sometimes optional
  - TCP does not have chunks

☹

TCP - Redundant Data Bundling

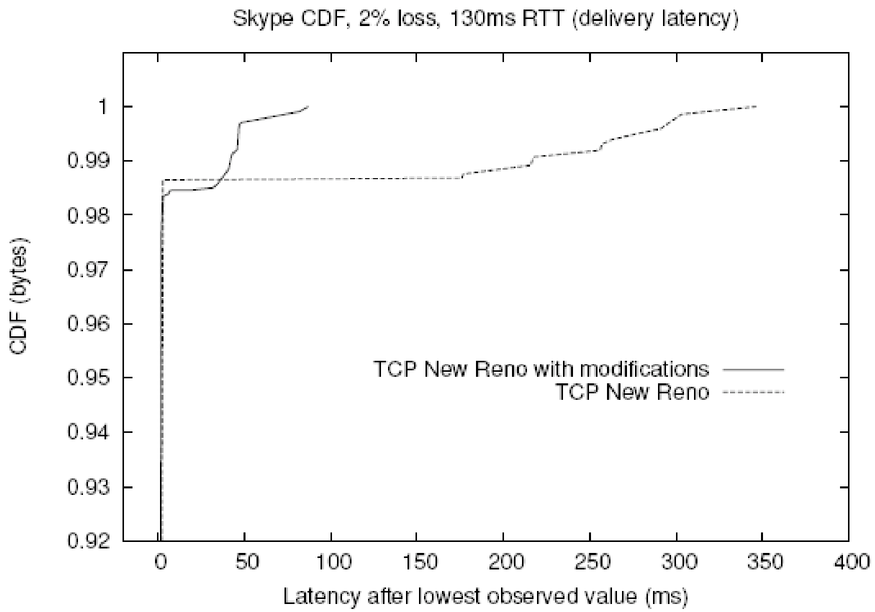
ENHANCEMENT:
Bundle all unacknowledged packets with each new transmission

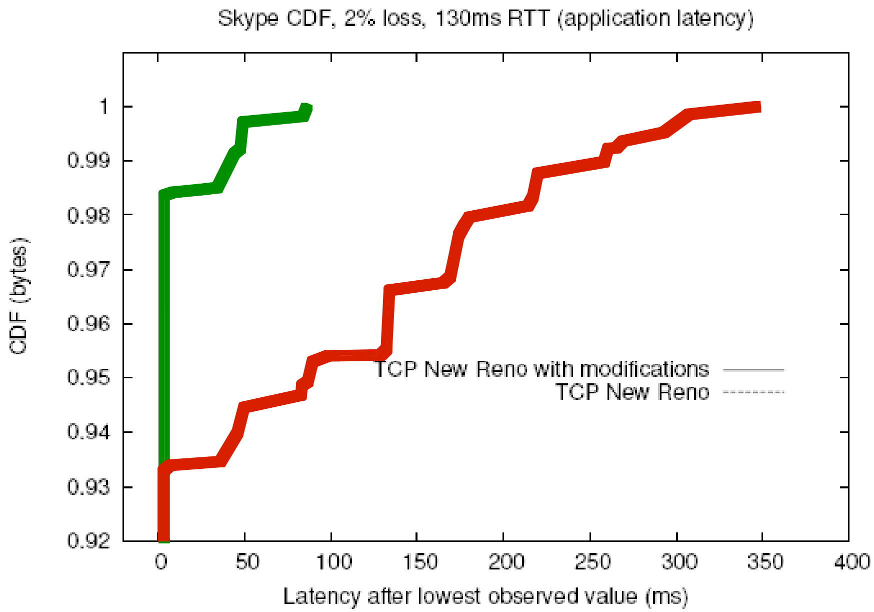
- If a packet is lost, there is a large chance that it will arrive bundled with the next packet.
- The following ACK will acknowledge both segments.
- TCP standard compatible



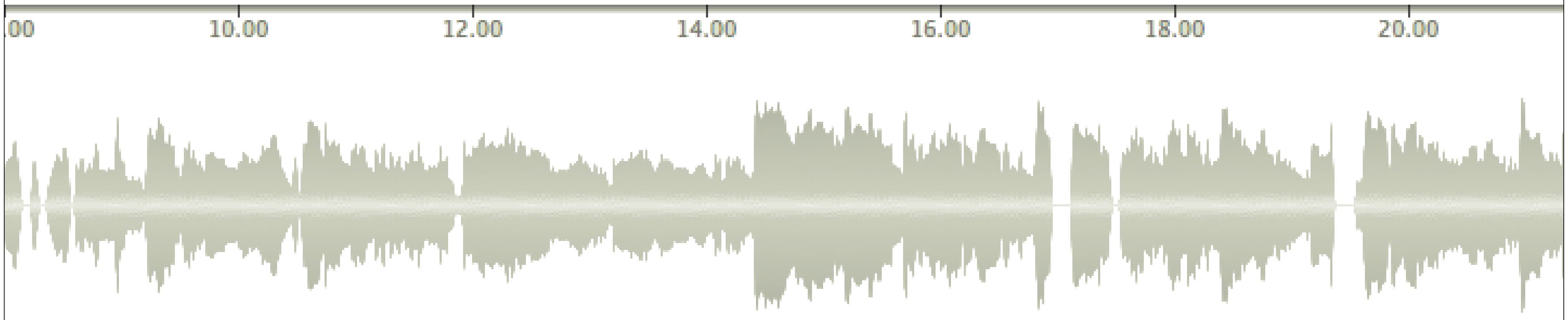
Internet latency improvements

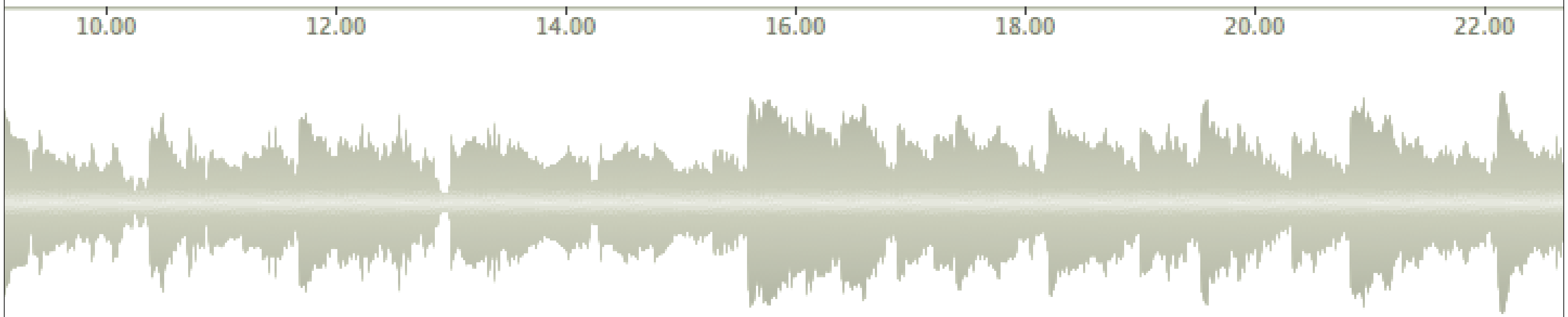
- Performed several tests (VoIP, games, remote terminals) measuring improvements in data delivery latency
- User tests



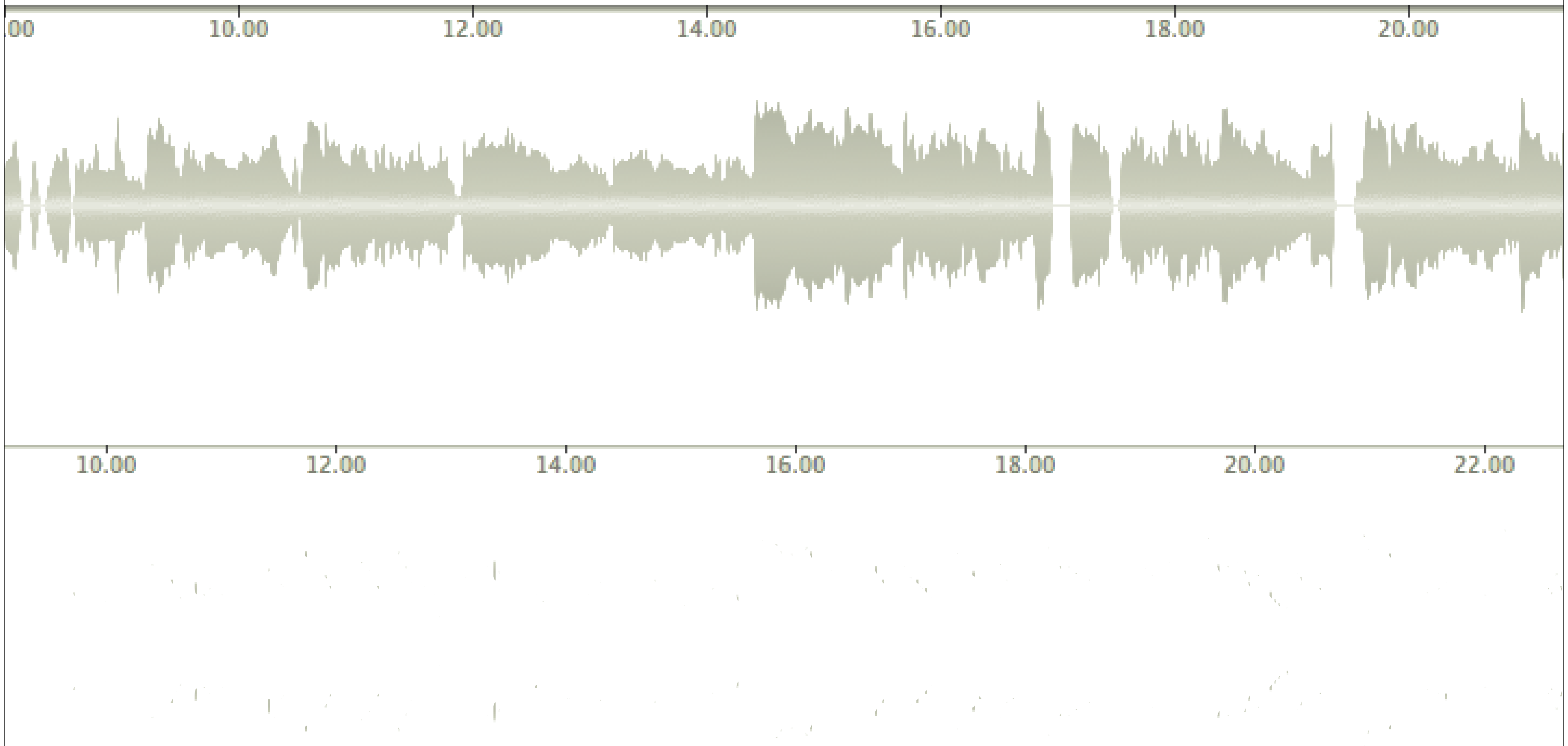


Internet latency improvements

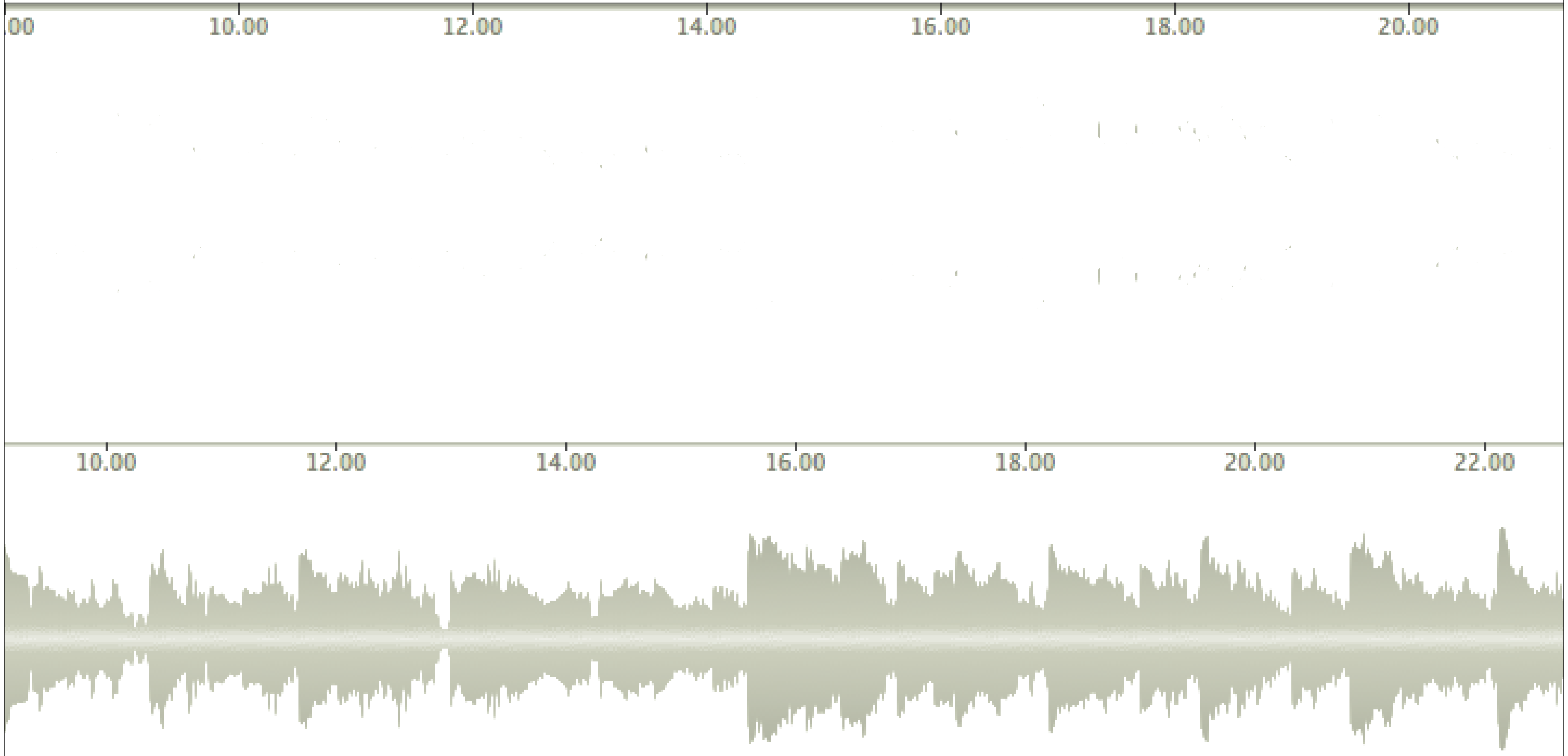




Internet latency improvements



Internet latency improvements



Thin stream mechanism applicability

- From the properties we have discussed, we can derive four "classes" of streams

	Small Packets	Large Packets
High IA	Typical thin stream RDB, retrans, backoff	Rare faster retransmit, backoff
Low IA	Rare RDB	FTP, HTTP Thick

Interactive Applications

- Summary
  - Interactive applications require low latency
  - Current interactive applications generate Thin Streams
  - Our options
    - use UDP,
      fix problems in the application
    - use TCP or SCTP,
      live with high latency
    - use TCP or SCTP,
      fix problems in the protocol

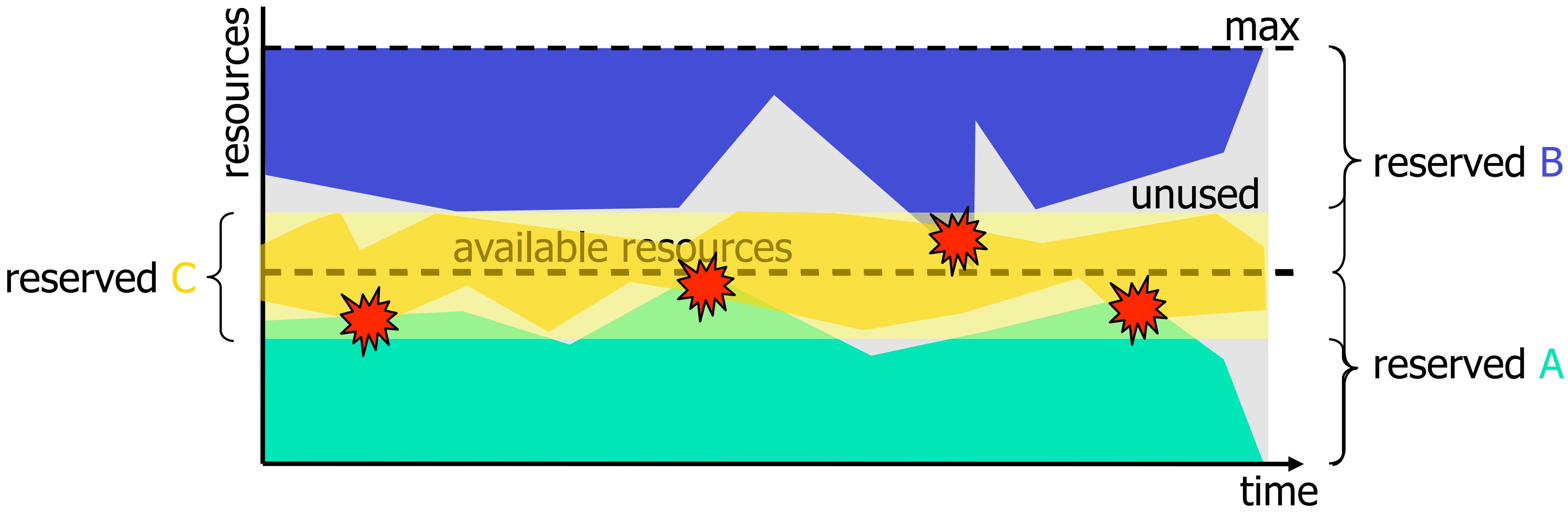
Quality-of-Service

Overview

- Quality-of-Service
- Per-packet QoS
  - IP
- Per-flow QoS
  - Resource reservation
- QoS Aggregates
  - DiffServ, MPLS

Quality–of–Service (QoS)

- Different semantics or classes of QoS:
  - determines reliability of offered service
  - utilization of resources



Quality–of–Service (QoS)

- Best effort QoS:
  - system tries its best to give a good performance
  - no QoS calculation (could be called no effort QoS)

  ☺ simple – do nothing

  ☹ QoS may be violated → unreliable service

- Deterministic guaranteed QoS:
  - hard bounds
  - QoS calculation based on upper bounds (worst case)
  - premium better name!!??

  ☺ QoS is satisfied even in the worst case → high reliability

  ☹ over-reservation of resources → poor utilization and unnecessary service rejects

  ☹ QoS values may be less than calculated hard upper bound

Quality–of–Service (QoS)

- Statistical guaranteed QoS:
  - QoS values are statistical expressions (served with some probability)
  - QoS calculation based on average (or some other statistic or stochastic value)

  ☺ resource capabilities can be statistically multiplexed → more granted requests

  ☹ QoS may be temporarily violated → service not always 100 % reliable

- Predictive QoS:
  - weak bounds
  - QoS calculation based previous behavior of imposed workload

Quality–of–Service

- Applicability: QoS support
  - A dream of early network researchers
    (lots of research topics)
  - Guarantees that distributed systems work as promised
- QoS doesn't exist?
  - IP doesn't support QoS
  - Equality is the Internet's mantra
    (do you listen to the net neutrality debate?)
  - Violates Internet philosophy
    (shunned by the gurus)
- QoS requirement
  - Companies and end-users demand guarantees
  - What's being done?

Per-packet QoS

Internet Protocol version 4 (IPv4)

[RFC1349]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Pre|    ToS   |0|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                               |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PRE | D | T | R | C | 0

ToS

ToS

- Type of Service
  - D – minimize delay
  - T – maximize throughput
  - R – maximize reliability
  - C – minimize cost

PRE

- Precedence Field
  - Priority of the packet

Internet Protocol version 4 (IPv4)

[RFC2474]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |    DSCP    |0 0|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                              |     Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

0 0

Class selector codepoints
of the form xxx000

DSCP

- Differentiated Services Codepoint
  - xxxxx0 reserved for standardization
  - xxxx11 reserved for local use
  - xxxx01 open for local use, may be standardized later

Internet Protocol version 6 (IPv6)

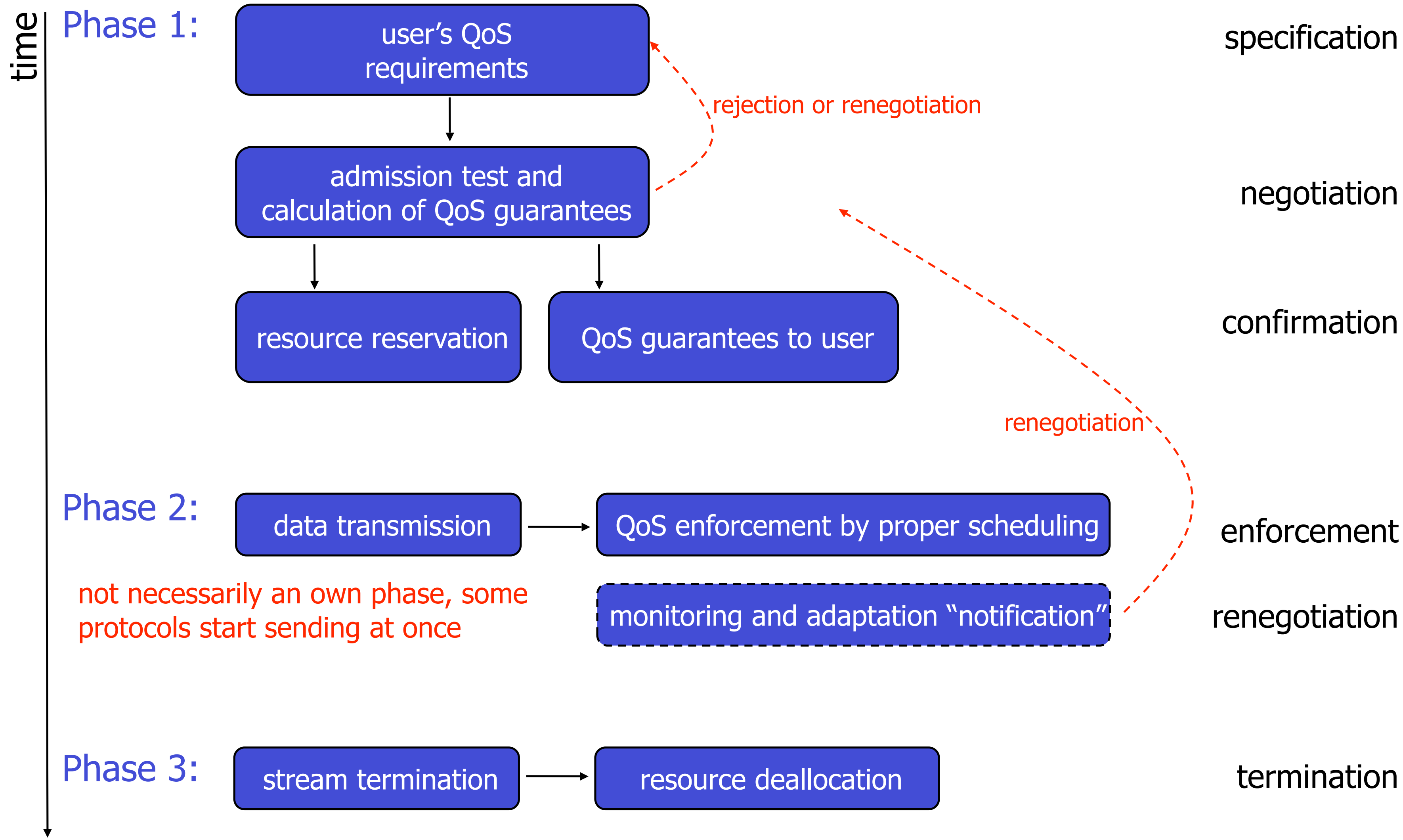
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Traffic class
  - Interpret like IPv4's DS field

Per-flow QoS

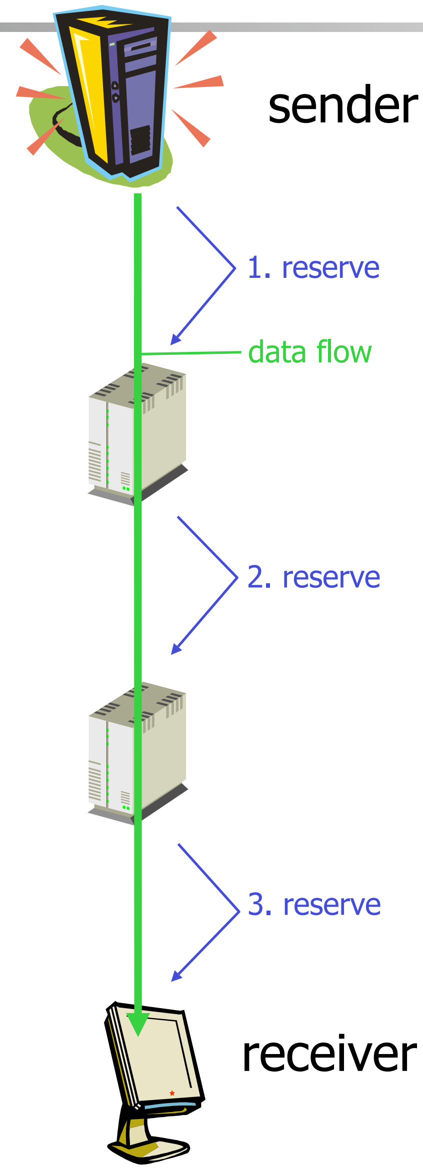
Resource Reservation

Resource Management Phases



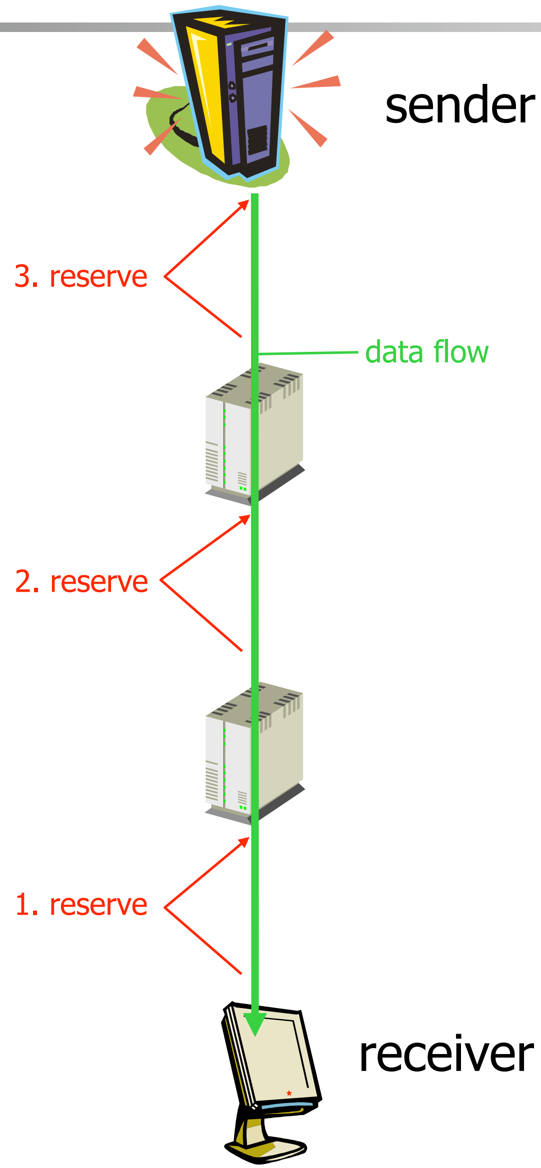
Reservation Directions

- Sender oriented:
  - sender (initiates reservation)
    - must know target addresses (participants)
    - in-scalable
    - good security



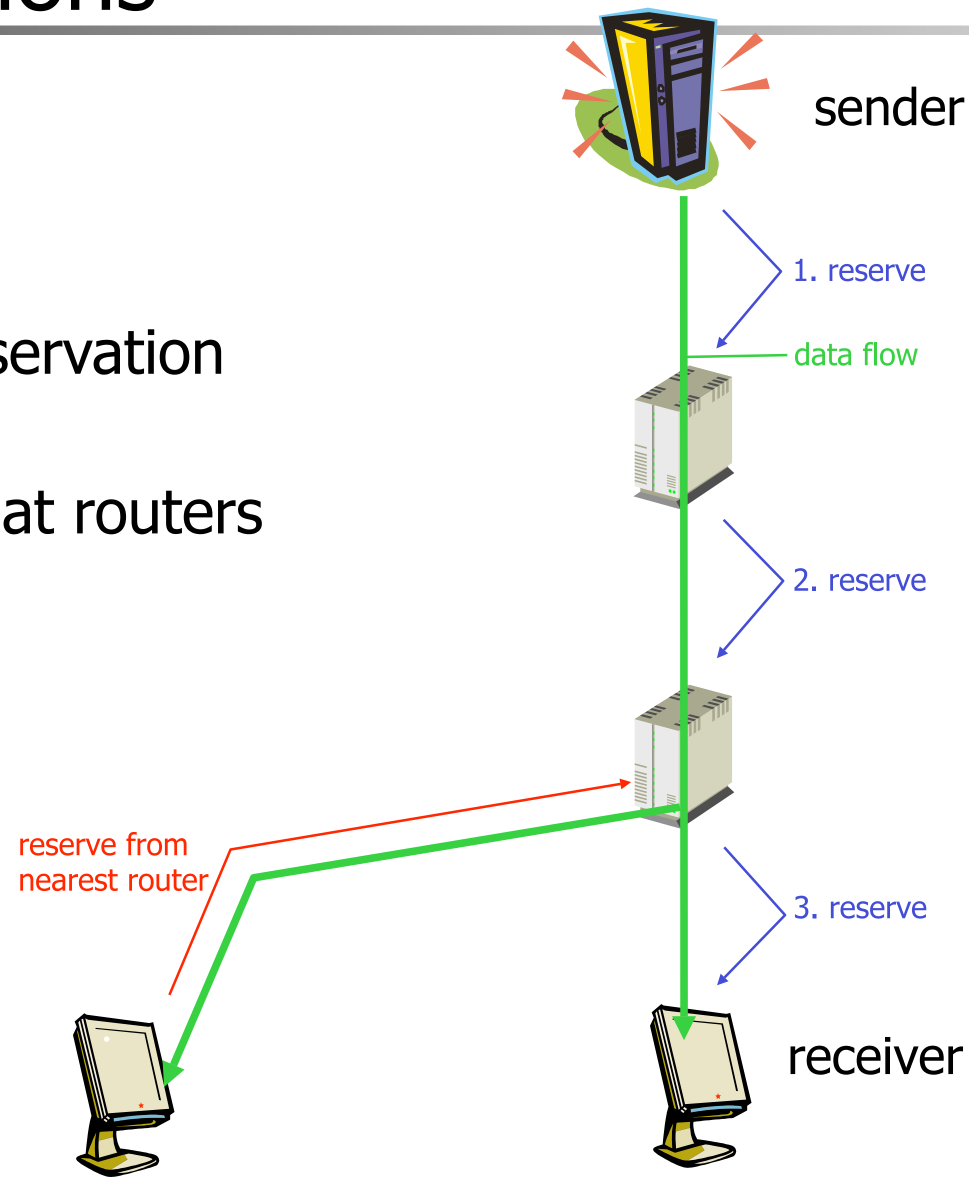
Reservation Directions

- Receiver oriented:
  - receiver (initiates reservation)
    - needs advertisement before reservation
    - must know "flow" addresses
  - sender
    - need not to know receivers
    - more scalable
    - in-secure



Reservation Directions

- Combination?
  - start sender oriented reservation
  - additional receivers join at routers (receiver based)



Per-flow QoS

Integrated Services

Integrated Services (IntServ)

- Framework by IETF to provide individualized QoS guarantees to individual application sessions

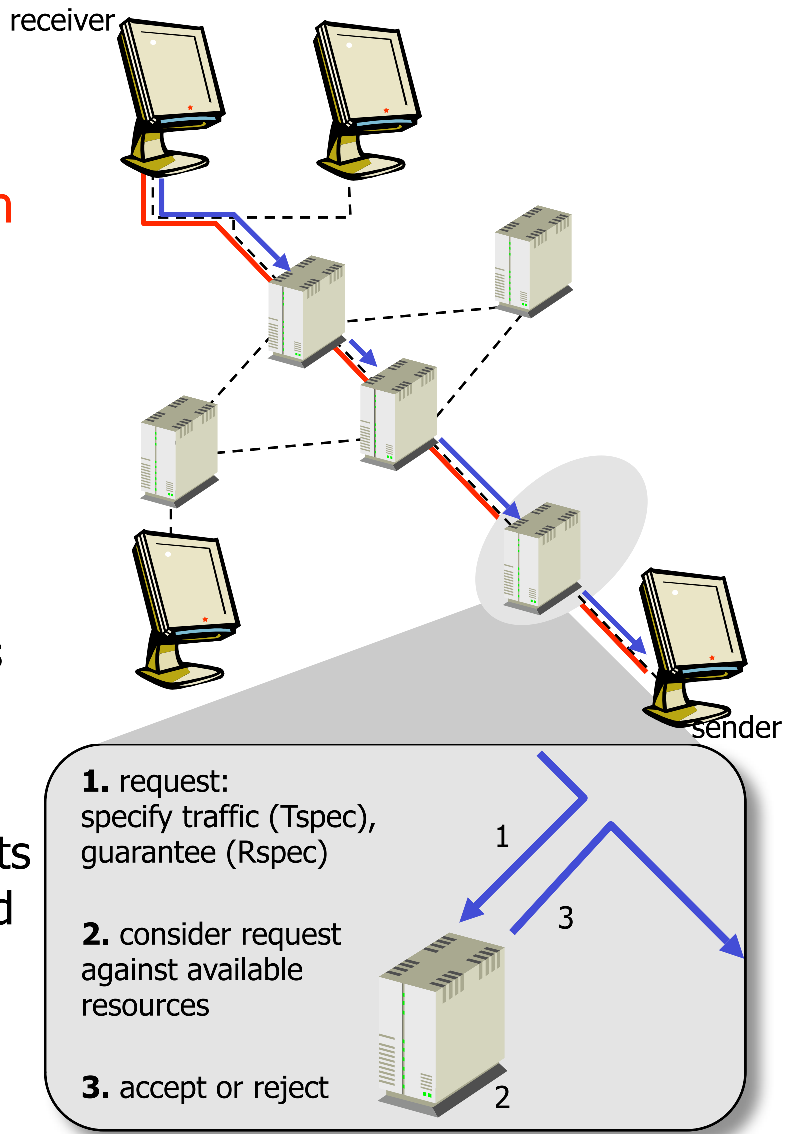
- Goals:
  - efficient Internet support for applications which require service guarantees
  - fulfill demands of multipoint, real-time applications (like video conferences)
  - do not introduce new data transfer protocols

- In the Internet, it is based on IP (v4 or v6) and RSVP
  - RSVP – Resource reSerVation Protocol

- Two key features
  - reserved resources – the routers need to know what resources are available (both free and reserved)
  - call setup (admission call) – reserve resources on the whole path from source to destination

Integrated Services (IntServ)

- Admission call:
  - traffic characterization and specification
    - one must specify the traffic one will transmit on the network (Tspec)
    - one must specify the requested QoS (Rspec – reservation specification)
  - signaling for setup
    - send the Tspec and Rspec to all routers
  - per-element admission test
    - each router checks whether the requests specified in the R/Tspecs can be fulfilled
    - if YES, accept; reject otherwise



Integrated Services (IntServ)

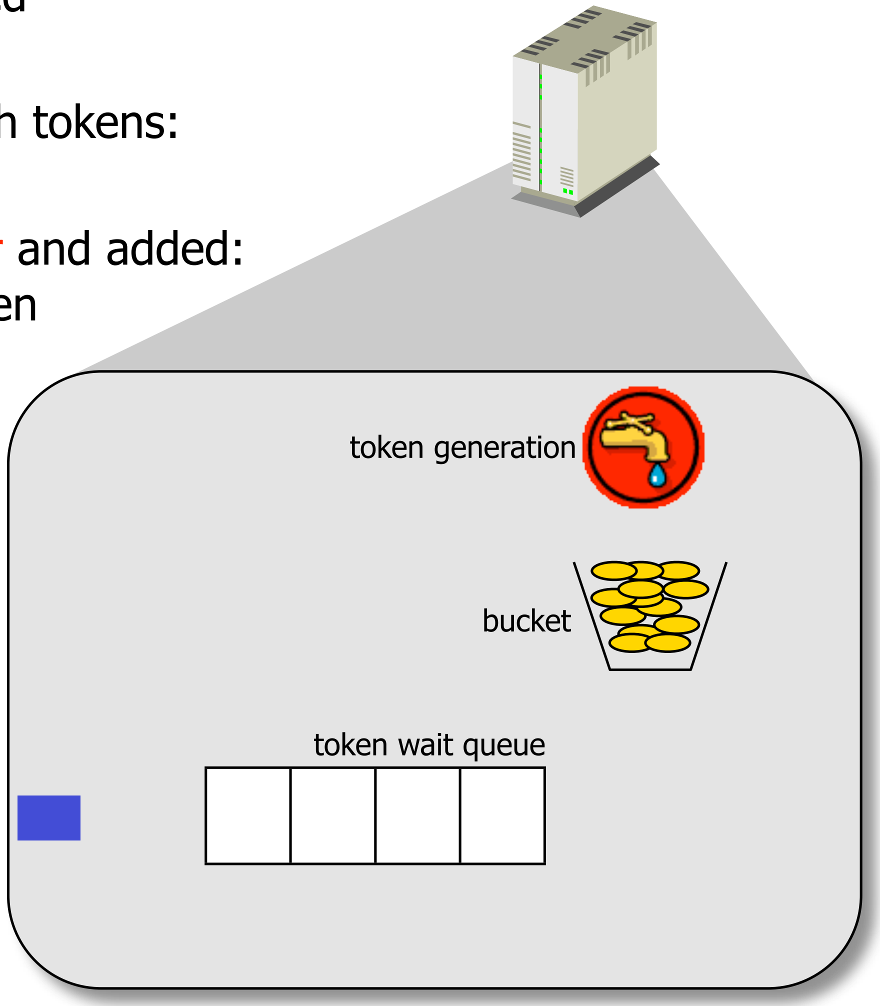
- IntServ introduces two new services enhancing the Internet's traditional best effort:

  - guaranteed service
    - guaranteed bounds on delay and bandwidth
    - for applications with real-time requirements

  - controlled-load service
    - "a QoS closely to the QoS the same flow would receive from an unloaded network element" [RFC 2212], i.e.,
      similar to best-effort in networks with limited load
    - no quantified guarantees,
      but packets should arrive with "a very high percentage"
    - for applications that can adapt to moderate losses, e.g.,
      real-time multimedia applications

Integrated Services (IntServ)

- Both service classes use token bucket to police a packet flow:
  - packets need a token to be forwarded
  - each router has a b-sized bucket with tokens: if bucket is empty, one must wait
  - new tokens are generated at a rate r and added: if bucket is full (little traffic), the token is deleted
  - the token generation rate r serves to limit the long term average rate
  - the bucket size b serves to limit the maximum burst size



Integrated Services (IntServ)

- Today implemented
  - in every router
  - for every operating system
    (its signaling protocol RSVP was even switched on by default from Windows NT to Windows XP)
- … and not used
- Arguments
  - too much overhead
  - too large memory requirements
  - too inflexible
  - “net neutrality” argument
  - no commercial model

QoS Aggregates

Protocols

Differentiated Services (DiffServ)

- IntServ and RSVP provide a framework for per-flow QoS, but they …
  - … give complex routers
    - much information to handle
  - … have scalability problems
    - set up and maintain per-flow state information
    - periodically PATH and RESV messages overhead
  - … specify only a predefined set of services
    - new applications may require other flexible services

⇨ DiffServ [RFC 2475] tries to be both scalable and flexible

Differentiated Services (DiffServ)

- ISPs favor DiffServ
- Basic idea
  - multicast is not necessary

  - make the core network simple - support to many users
  - implement more complex control operations at the edge
  - aggregation of flows –
    reservations for a group of flows, not per flow

  ⇨ *avoid scalability problems on routers with many flows*

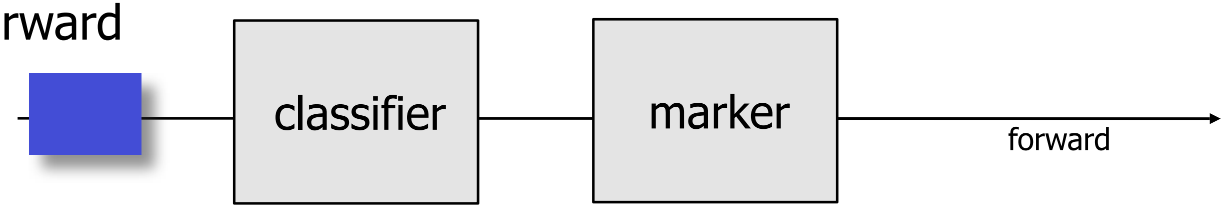
  - do not specify services or service classes
  - instead, provide the functional components on which services can be built

  ⇨ *support flexible services*

Differentiated Services (DiffServ)

- Two sets of functional elements:
  - edge functions: packet classification and traffic conditioning
  - core function: packet forwarding

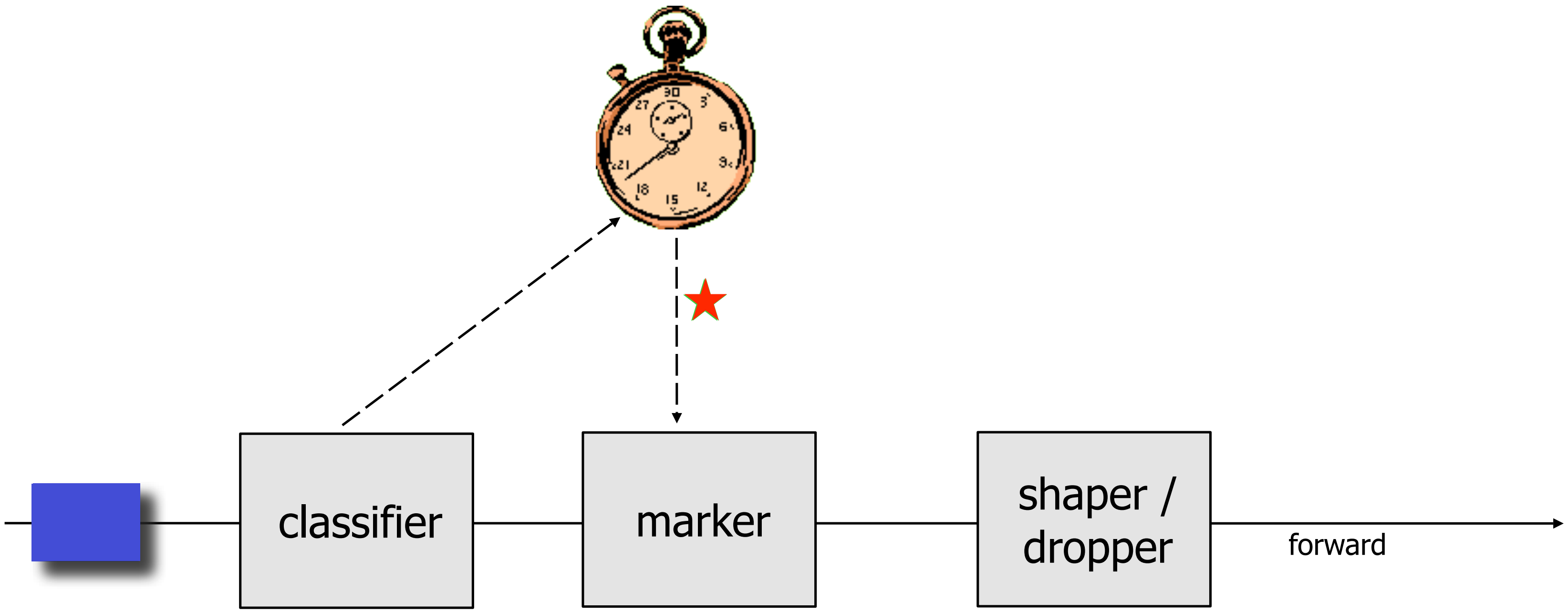
- At the edge routers, the packets are tagged with a DS-mark (differentiated service mark)
  - uses the type of service field (IPv4) or the traffic class field (IPv6)
  - different service classes (DS-marks) receive different service
  - subsequent routers treat the packet according to the DS-mark
  - classification:
    - incoming packet is classified (and steered to the appropriate marker function) using the header fields
    - the DS-mark is set by marker
    - once marked, forward



Differentiated Services (DiffServ)

- Note: there are no "rules" for classification – it is up to the network provider

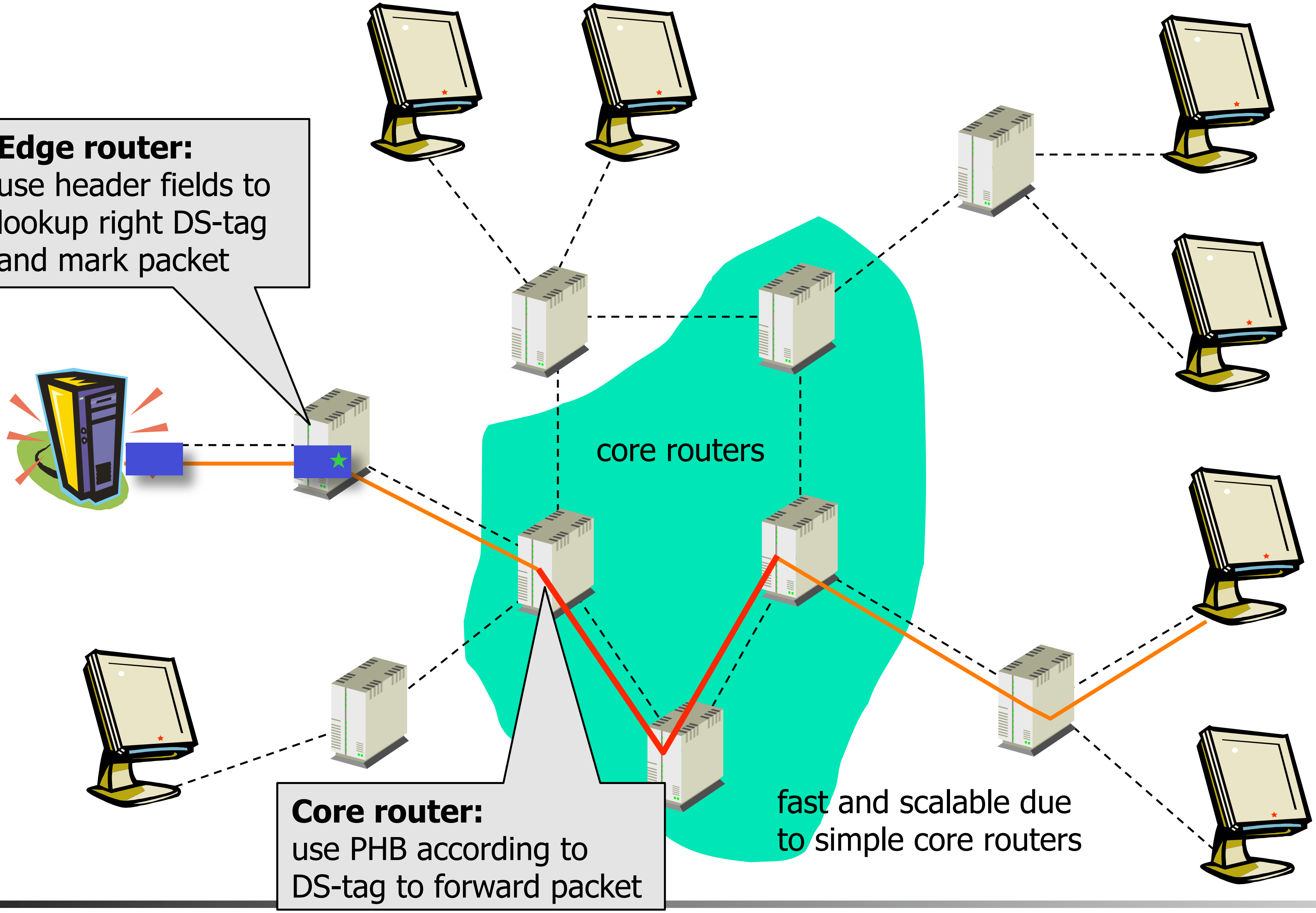
- A metric function may be used to limit the packet rate:
  - the traffic profile may define rate and maximum bursts
  - if packets arrive too fast, the metric function assigns another marker function telling the router to delay or drop the packet



Differentiated Services (DiffServ)

- In core routers, DS-marked packets are forwarded according to their per-hop behavior (PHB) - by looking up the meaning of their DS-tag
  - the PHB determines how the router resources are used and shared among the competing service classes
  - the PHB should be based on the DS-tag only
    - no other state in the router
  - traffic aggregation
    - packets with same DS-tag are treated equally
    - regardless of original source or final destination
  - a PHB can result in different service classes receiving different performance
  - performance differences must be observable and measurable to allowing monitoring of the system performance
  - no specific mechanism for achieving these behaviors are specified

Differentiated Services (DiffServ)



Differentiated Services (DiffServ)

- First two defined PHBs are
  - expedited forwarding [RFC 3246]
    - specifies a minimum departure rate of a class
      this implies a guaranteed bandwidth for the class
    - the guarantee is independent of other classes, i.e.,
      enough resources must be available regardless of competing traffic
  - assured forwarding [RFC 2597]
    - divide traffic into four classes
    - each class is guaranteed a minimum amount of resources
    - each class is further partitioned into one of three "drop" categories
      (if congestion occurs, the router drops packets based on "drop" value)

Multiprotocol Label Switching (MPLS)

- Multiprotocol Label Switching
  - Separate path determination from hop-by-hop forwarding
  - Forwarding is based on labels
  - Path is determined by choosing labels

- Distribution of labels
  - On application-demand
    - LDP – label distribution protocol
  - By traffic engineering decision
    - RSVP-TE – traffic engineering extensions to RSVP

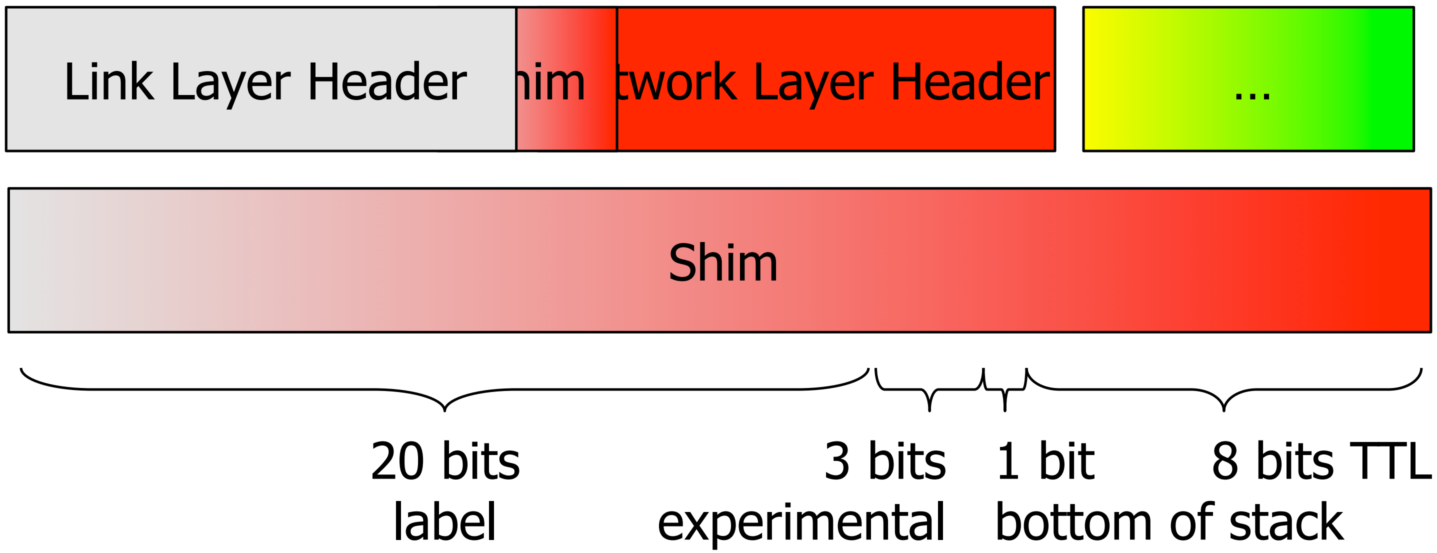
Multiprotocol Label Switching (MPLS)

- MPLS works above **multiple** link layer **protocols**
- Carrying the **label**
  - Over ATM
    - Virtual path identifier or Virtual channel identifier
    - Maybe shim
  - Frame Relay
    - data link connection identifier (DLCI)
    - Maybe shim
  - Ethernet, TokenRing, …
    - Shim

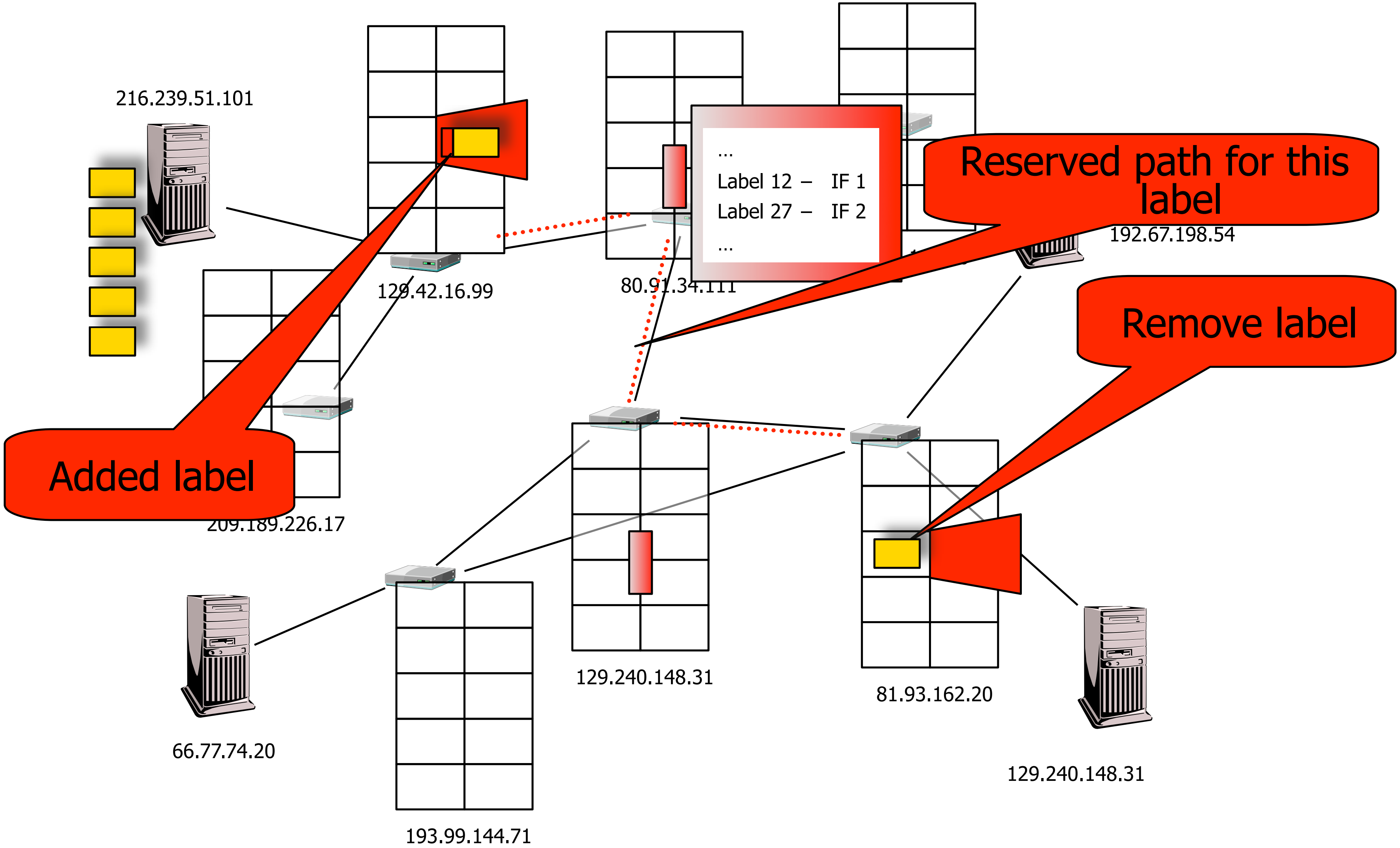
- Shim?

Multiprotocol Label Switching (MPLS)

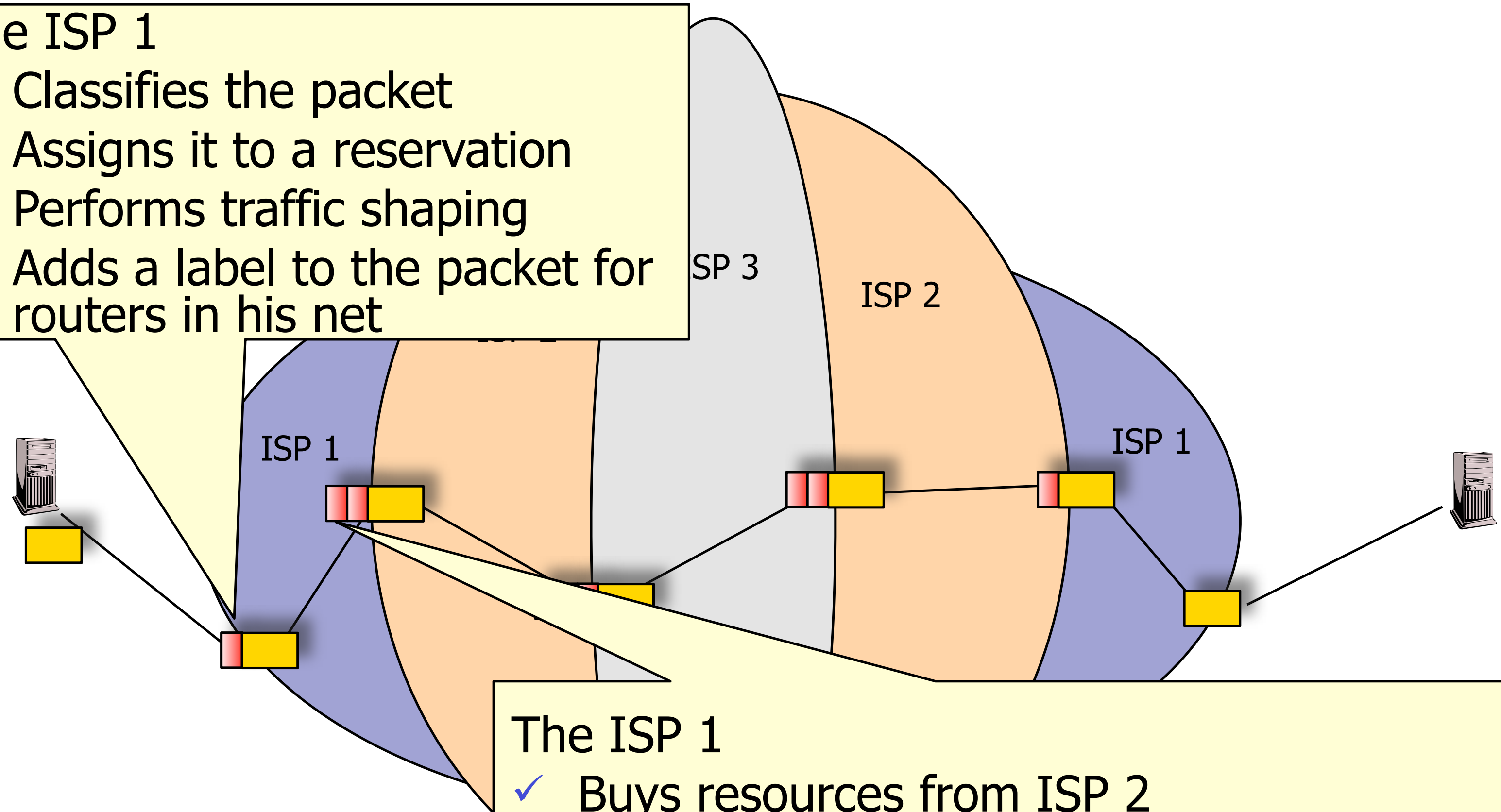
- Shim: the label itself



Routing using MPLS



MPLS Label Stack

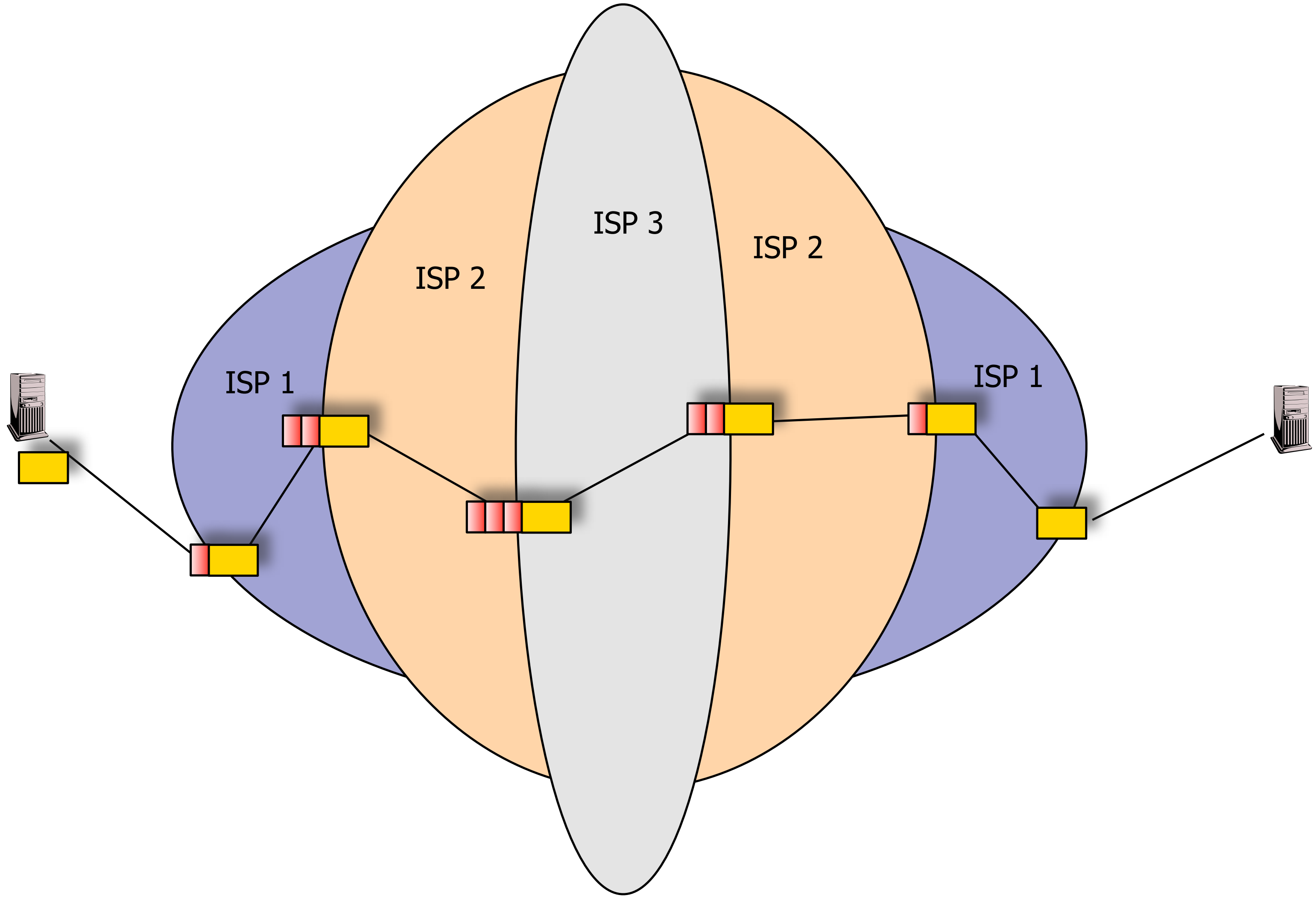


The ISP 1
- ✓ Classifies the packet
- ✓ Assigns it to a reservation
- ✓ Performs traffic shaping
- ✓ Adds a label to the packet for routers in his net

The ISP 1
- ✓ Buys resources from ISP 2

The ISP 2
- ✓ Repeats classifying, assignment, shaping
- ✓ Adds a label for the routers in his net
- ✓ He **pushes a label on the label stack**

MPLS Label Stack



Summary

Directions of Network QoS

[Liebeherr]

- Old-style QoS is dead
  - ATM,
    IntServ,
    DiffServ,
    Service overlays didn't take hold
  - Causes?
    - No business case
    - Bothed standardization
    - Naïve implementations
    - No need
- Future QoS
  - Look for fundamental insights
  - Develop design principles
  - Develop analytical tools
    - Network calculus

[Crowcroft,Hand,Mortier,Roscoe,Warfield]

- Old-style QoS is dead
  - X.25 too little, too early
  - ATM too much, too late
  - IntServ too much, too early
  - DiffServ too little, too late
  - IP QoS not there
  - MPLS too isolated
- QoS through overlays can't work
- Future QoS
  - Single bit differentiation
  - Edge-based admission control
  - Micropayment

- Companies do provide QoS
  - AT&T
    - MPLS
  - Equant
    - MPLS
  - Cable and Wireless
    - ATM
    - MPLS
  - TeliaSonera
    - SDH
    - WDM
    - ATM
  - Nortel
    - MPLS
    - SONET/SDH
    - WDM

Directic

[Liebeherr ... er,Roscoe,Warfield]

- Old-style Q ... is dead
  - ATM, IntServ, DiffServ, Service c... hold ... e, too early ... ch, too late ... much, too early ... little, too late ... here ... lated
  - Causes?
    - No bu
    - Bothe
    - Naïve
    - No ne ... overlays can't
- Future QoS
  - Look for ... ferentiation
  - Develop ... admission control
  - Develop ... nt
    - Netw

Summary

- Timely access to resources is important for multimedia application to guarantee QoS – reservation might be necessary

- Many protocols have tried to introduce QoS into the Internet, but no protocol has yet won the battle...
  - often NOT only technological problems, e.g.,
    - scalability
    - flexibility
    - ...
  - but also economical and legacy reasons, e.g.,
    - IP rules – everything must use IP to be useful
    - several administrative domains (how to make ISPs agree)
    - router manufacturers will not take the high costs (in amount of resources) for per-flow reservations
    - pricing
    - ...

Summary

- What does it means for performance in distributed applications?
  - QoS protocols
    - either not present
    - or used for traffic multiplexes

⇒ Applications must adapt to bandwidth competition
- either to generic competing traffic
- or to traffic within a multiplex

⇒ End-to-end QoS can be statistically guaranteed
- Overprovisioning in access networks
- Network calculus in long-distance networks