INF5071 – Performance in Distributed Systems

Protocols

October 01, 2010

On–demand Streaming Applications

Stable bandwidth problem

UDP

- The classical solution
  - Send data at playout speed
  - Write loss-tolerant audio-video codecs
  - Ignore all kinds of loss, or use FEC

- Problem
  - Does not back off at bandwidth bottlenecks
  - TCP connections suffer

⇒ Approach is no longer accepted

TCP Congestion Control

- TCP congestion control is based on the notion that the network is a "black box" – congestion indicated by a loss

- Sufficient for best-effort applications, but losses might severely hurt traffic like audio and video streams
  → congestion indication can enable features like quality adaptation

Comparison of Non-QoS Philosophies

Pro UDP	Pro TCP

Using Standard Protocols

<table>
<tr><th>Over UDP</th><th>Over TCP</th><th>Alternative Transport</th></tr>
<tr><td>RTP
Real Time Protocol
IETF std, supported by ITU-T & Industry</td><td>RTP in RTSP over TCP
standardized worst-case fallback
firewall-friendly</td><td>SCTP
Stream Control Transmission Protocol
IETF RFC, supported by telephone industry</td></tr>
<tr><td>RLM
TCP-friendly, needs fine-grained layered video</td><td rowspan="2">"Progressive Download" or "HTTP Streaming"
application-level prefetching and buffering
trivial, cheap, firewall-friendly</td><td rowspan="2">DCCP
Datagram Congestion Control Protocol
IETF RFC, driven by TCP-friendliness researchers</td></tr>
<tr><td>SR-RTP
TCP-friendly with RTP/UDP
needs special encoding (OpenDivX)</td></tr>
<tr><td>VDP
Video Datagram Protocol
Research, for Vosaic</td><td rowspan="2">Priority Progress Streaming
needs special encoding
needs special routers for ’multicast’</td><td rowspan="2">PRTP-ECN
Partially reliable transport protocol using ECN
Research, Univ. Karlstad</td></tr>
<tr><td>MSP
Media Streaming Protocol
Research, UIUC</td></tr>
</table>

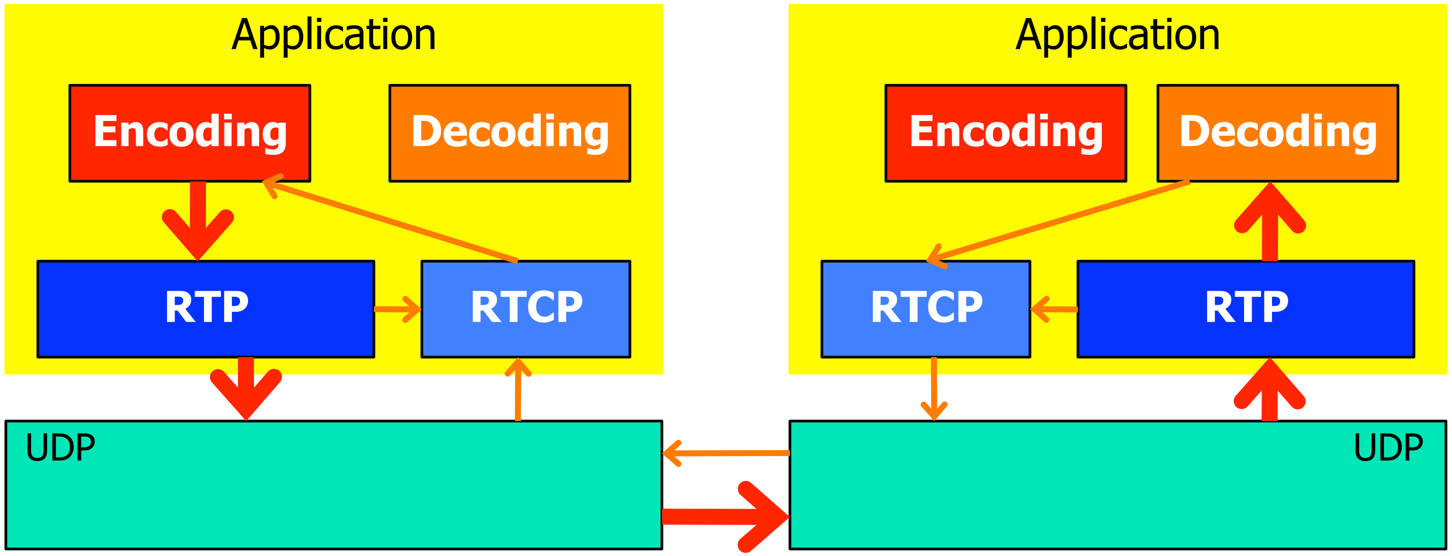
Real-time Transport Protocol (RTP)

- Real-time Transport Protocol (RTP)
  - RFC 1889
  - Designed for requirements of real-time data transport
  - **NOT** real-time
  - **NOT** a transport protocol

- Two Components:
  - Real-Time Transport Protocol (RTP)
  - RTP Control Protocol (RTCP)

- Provides end-to-end transport functions
  - Scalable in multicast scenarios
  - Media independent
  - Mixer and translator support
  - RTCP for QoS feedback and session information

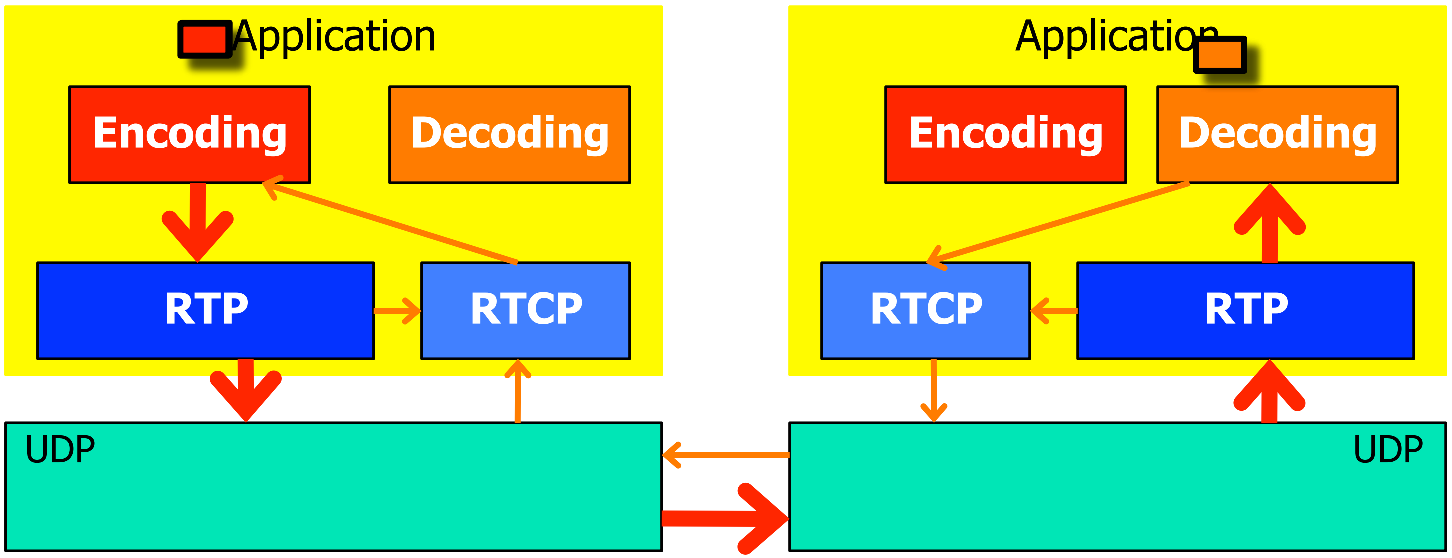
RTP Quality Adaptation



- Component interoperations for control of quality
- Evaluation of sender and receiver reports
- Modification of encoding schemes and parameters
- Adaptation of transmission rates
- Hook for possible retransmissions (outside RTP)

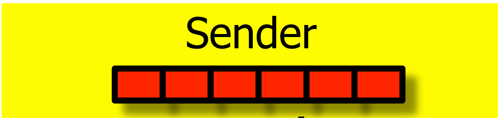
Loss-Delay Adjustment Algorithm

- LDA
  - An algorithm to stream with RTP in a TCP-friendly way
  - use RTCP receiver reports (RR)
    - RTCP sends RR periodically

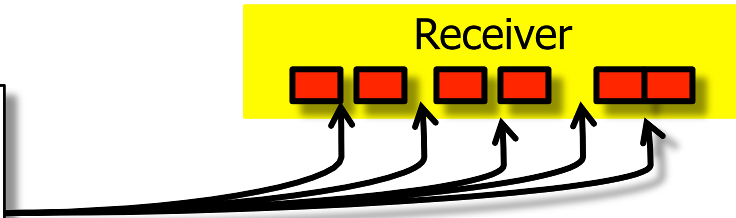


Loss-Delay Adjustment Algorithm

- LDA
  - An algorithm to stream with RTP in a TCP-friendly way
  - use RTCP receiver reports (RR)
    - RTCP sends RR periodically
  - works like TCP's AIMD
    - but RRs are rare
    - can't adapt every time
  - step one: find the bottleneck bandwidth b



  - use packet size and gaps size



$$b = \frac{packetsize}{gapsize}$$

Loss-Delay Adjustment Algorithm

- LDA
  - An algorithm to stream with RTP in a TCP-friendly way
  - use RTCP receiver reports (RR)
    - RTCP sends RR periodically
  - works like TCP's AIMD
    - but RRs are rare
    - can't adapt every time
  - no loss:
    - use "AIR" – additive increase *rate*
    - but never more than 1 packet/RTT
  - loss:
    - RTCP counts losses l
    - *guess* 3 of those losses in one RTT:

current rate

$$AIR_{i+1} = AIR_i * \left(1 - \frac{r_t}{b}\right)$$

$$r_{t+1} = r_t + AIR_{i+1}$$

newrate

$$r_{t+1} = r_t * (1 - l * 3)$$

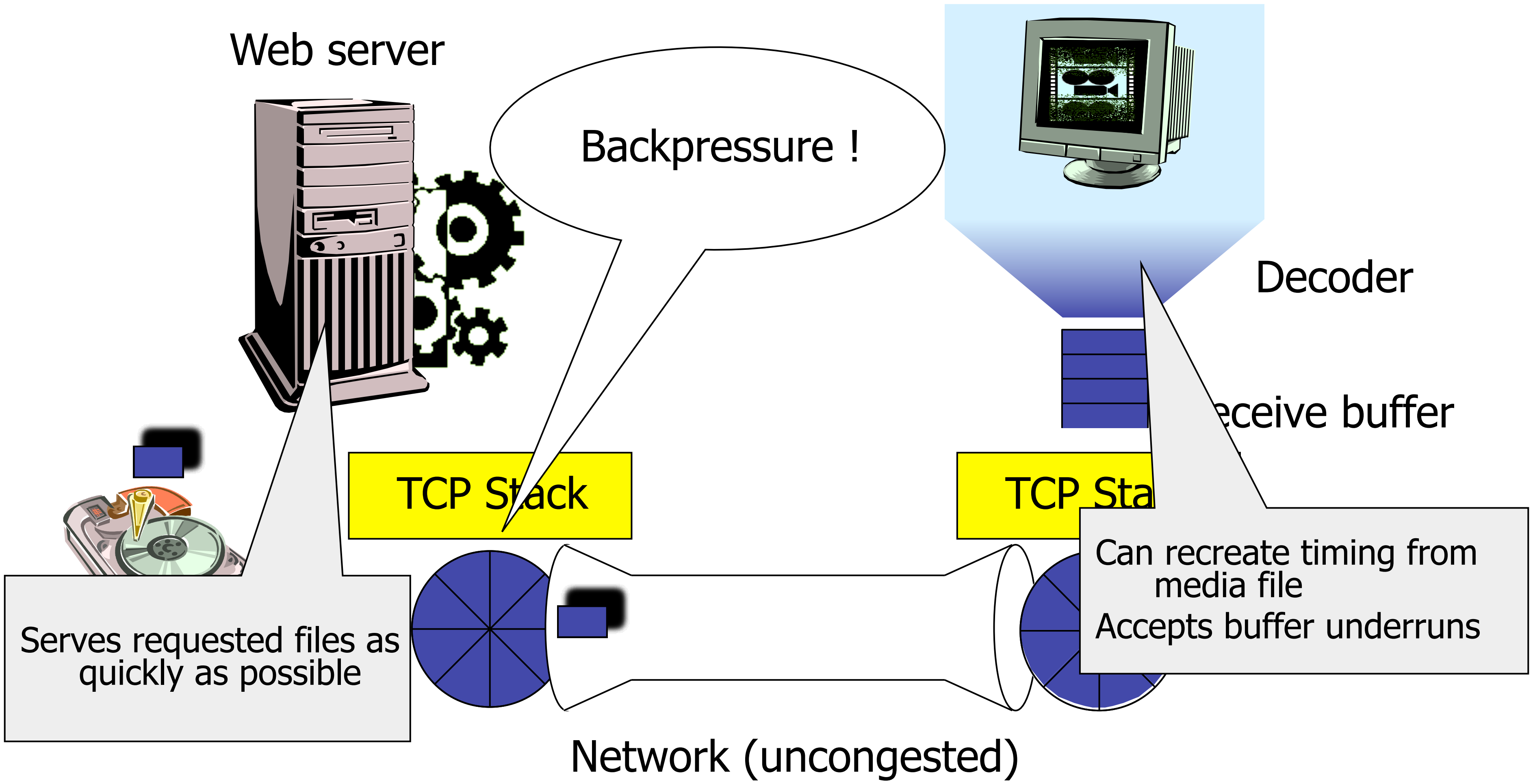
Progressive Download

- In-band in long-running HTTP response
  - Plain file for the web server
  - Even simpler than FTP
  - No user interactions – start, stop, ...

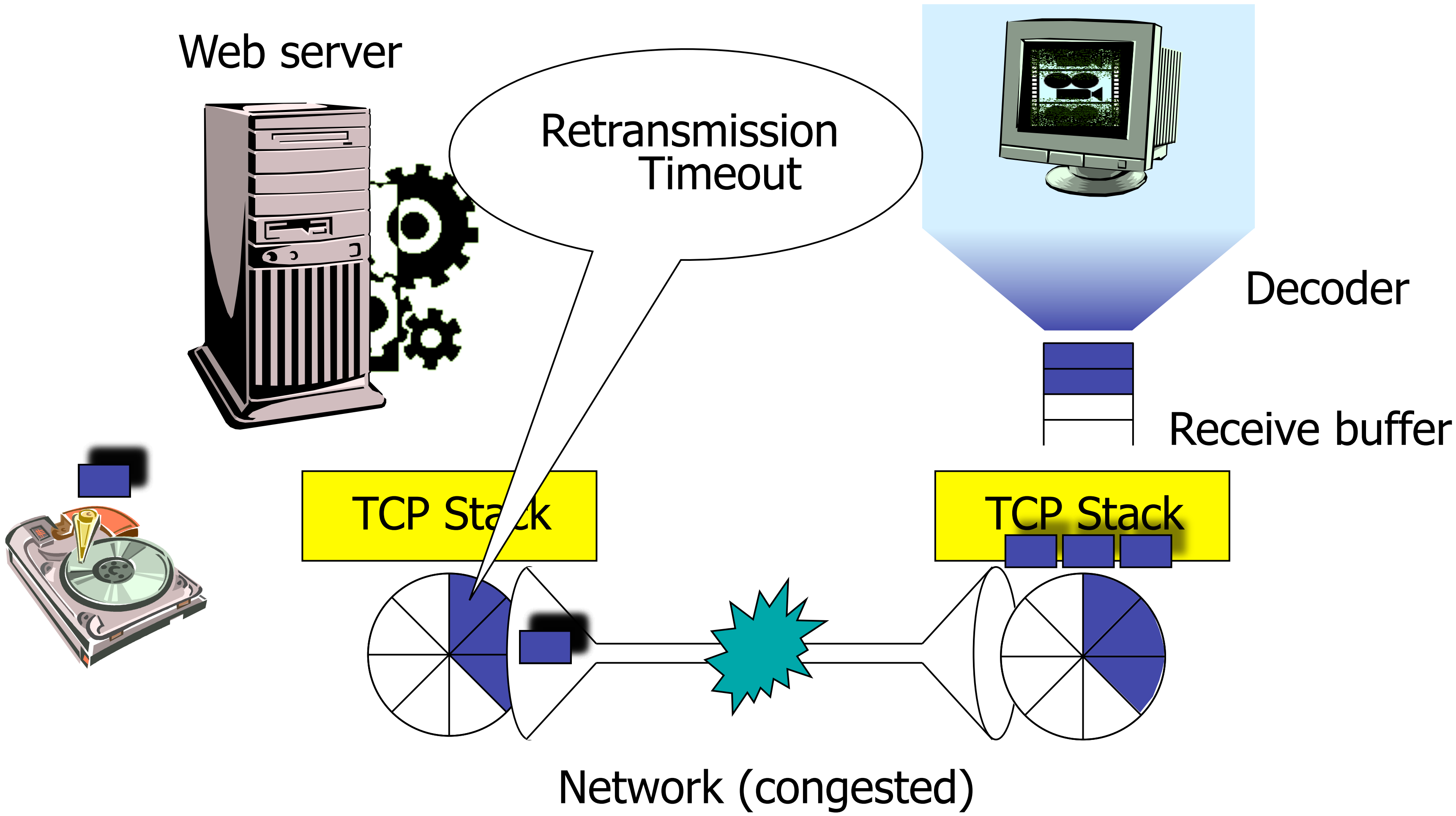
- If packet loss is ...
  - ... low – rate control by back-pressure from client
  - ... high – client's problem

- Applicability
  - Theoretical
    - For very low-bit-rate codecs
    - For very loss-intolerant codecs
  - Practical
    - All low-volume web servers

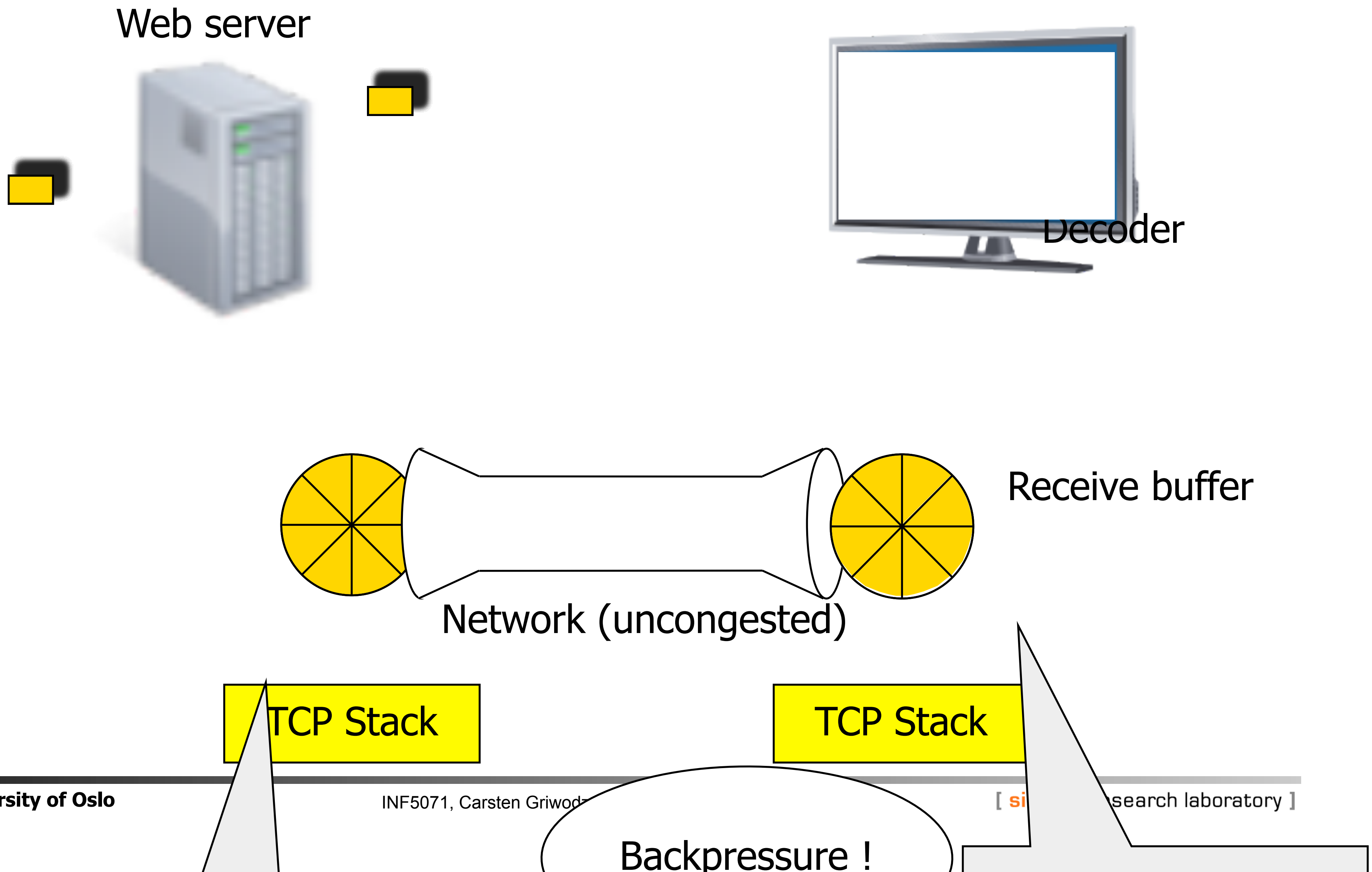
Progressive Download



Progressive Download

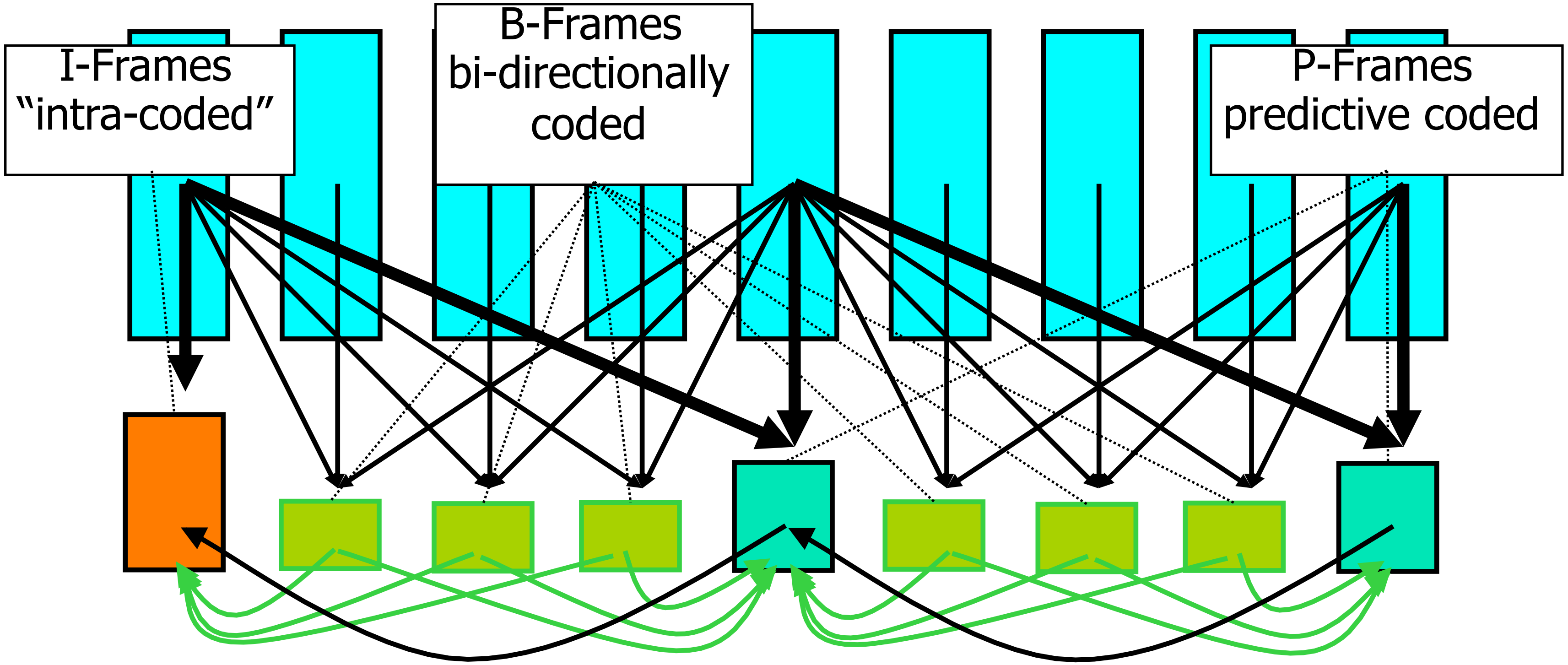


Progressive Download



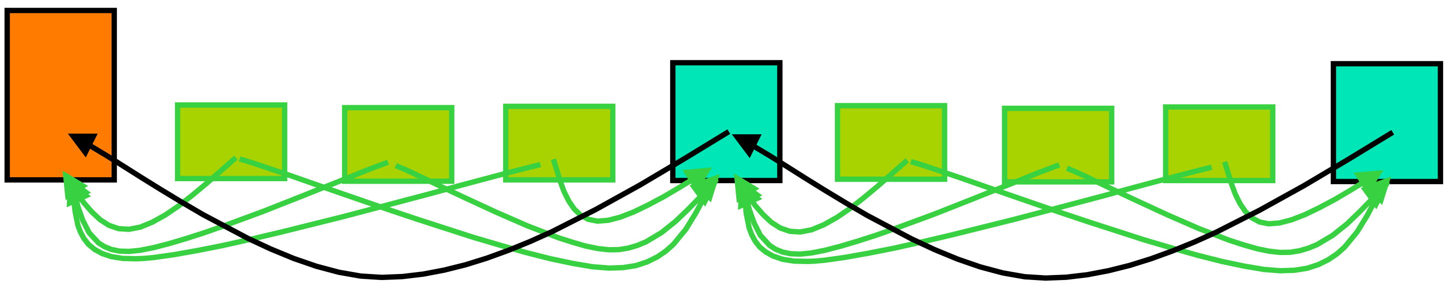
Coding for Adaptive Streaming: MPEG-1

- International Standard: Moving Pictures Expert Group
  - Compression of audio and video for playback (1.5 Mbit/s)
  - Real-time decoding
- Sequence of I-, P-, and B-Frames

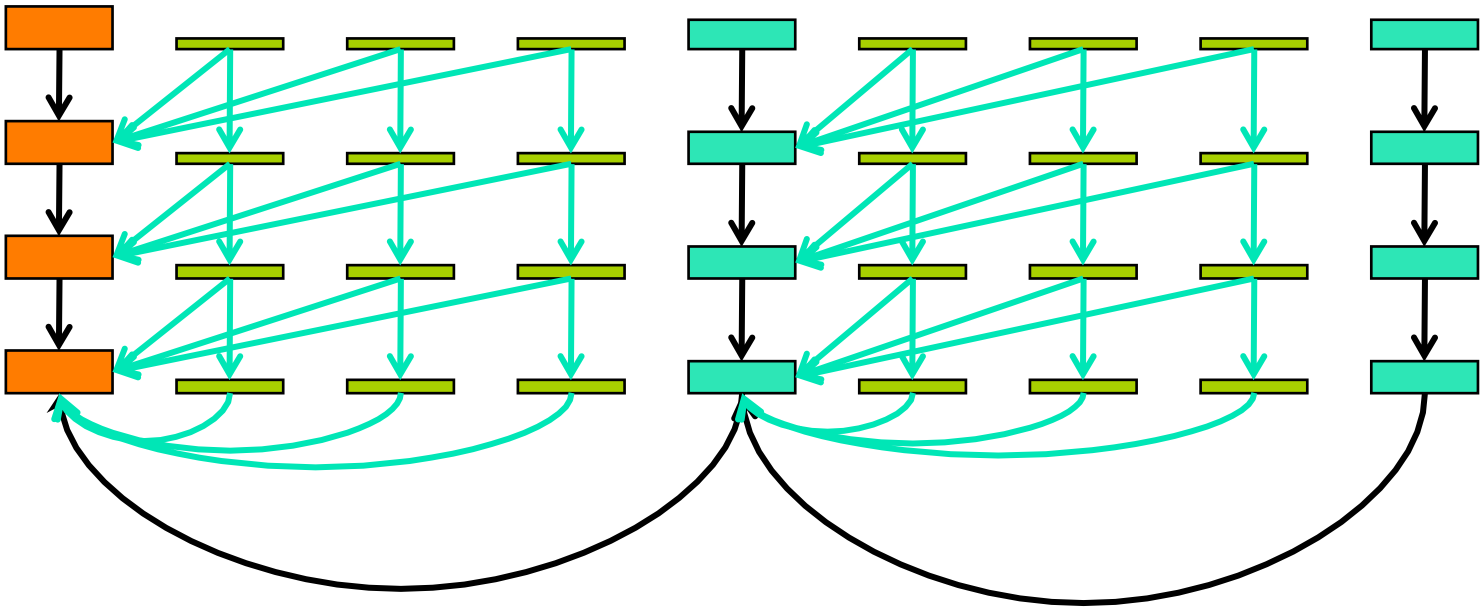


Coding ...: MPEG-1

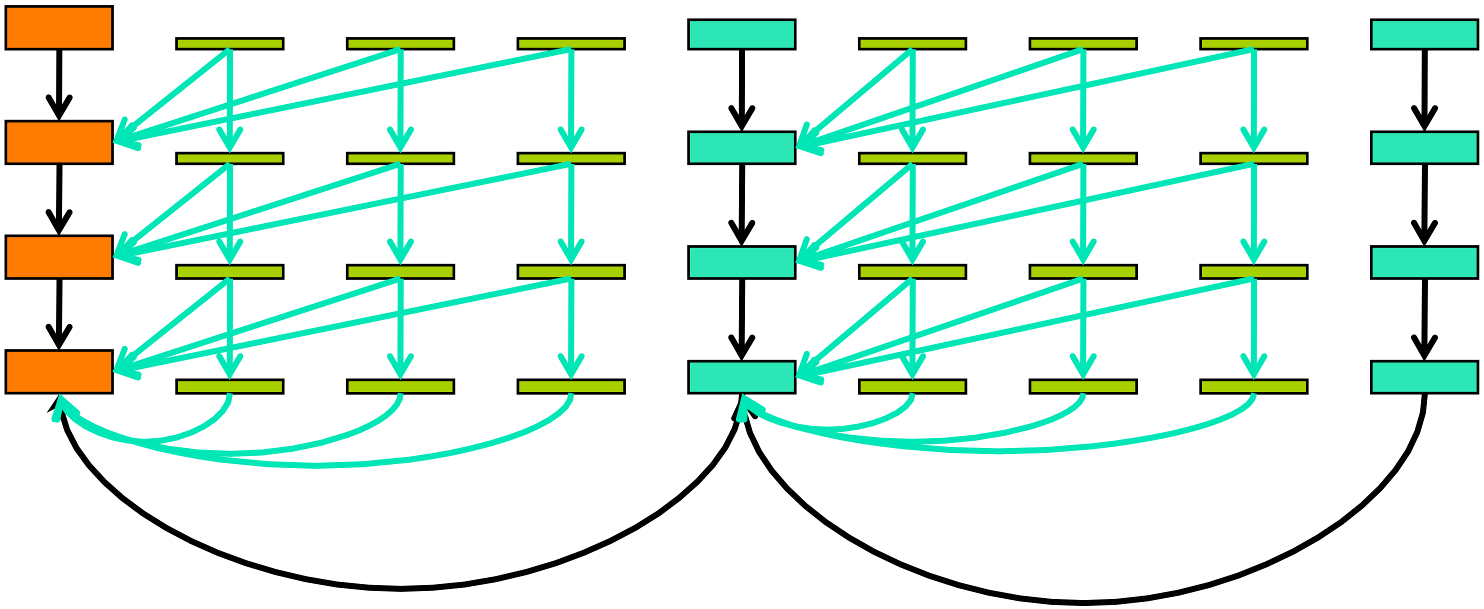
- Frames can be dropped
  - In a controlled manner
  - Frame dropping does not violate dependancies
  - Example: B-frame dropping in MPEG-1



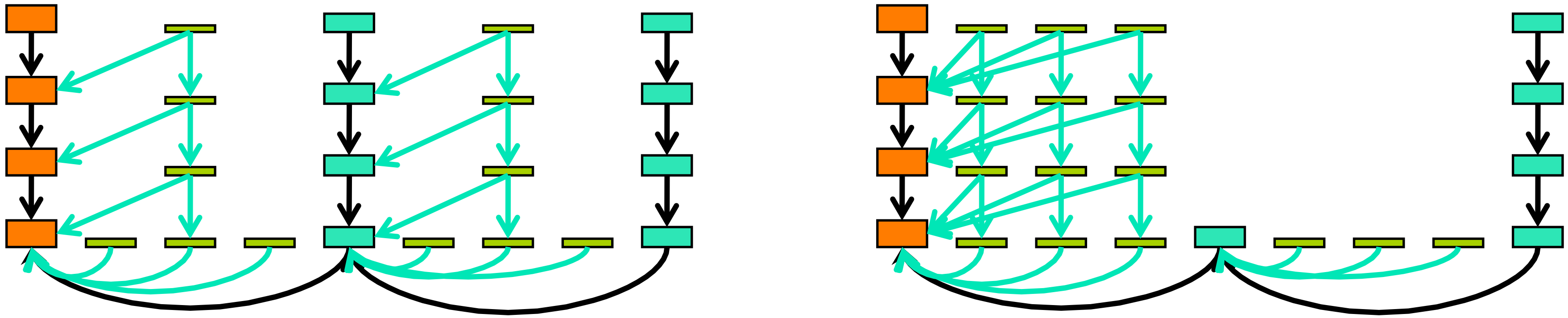
Coding ...: hierarchical layer coding



Coding ...: hierarchical layer coding



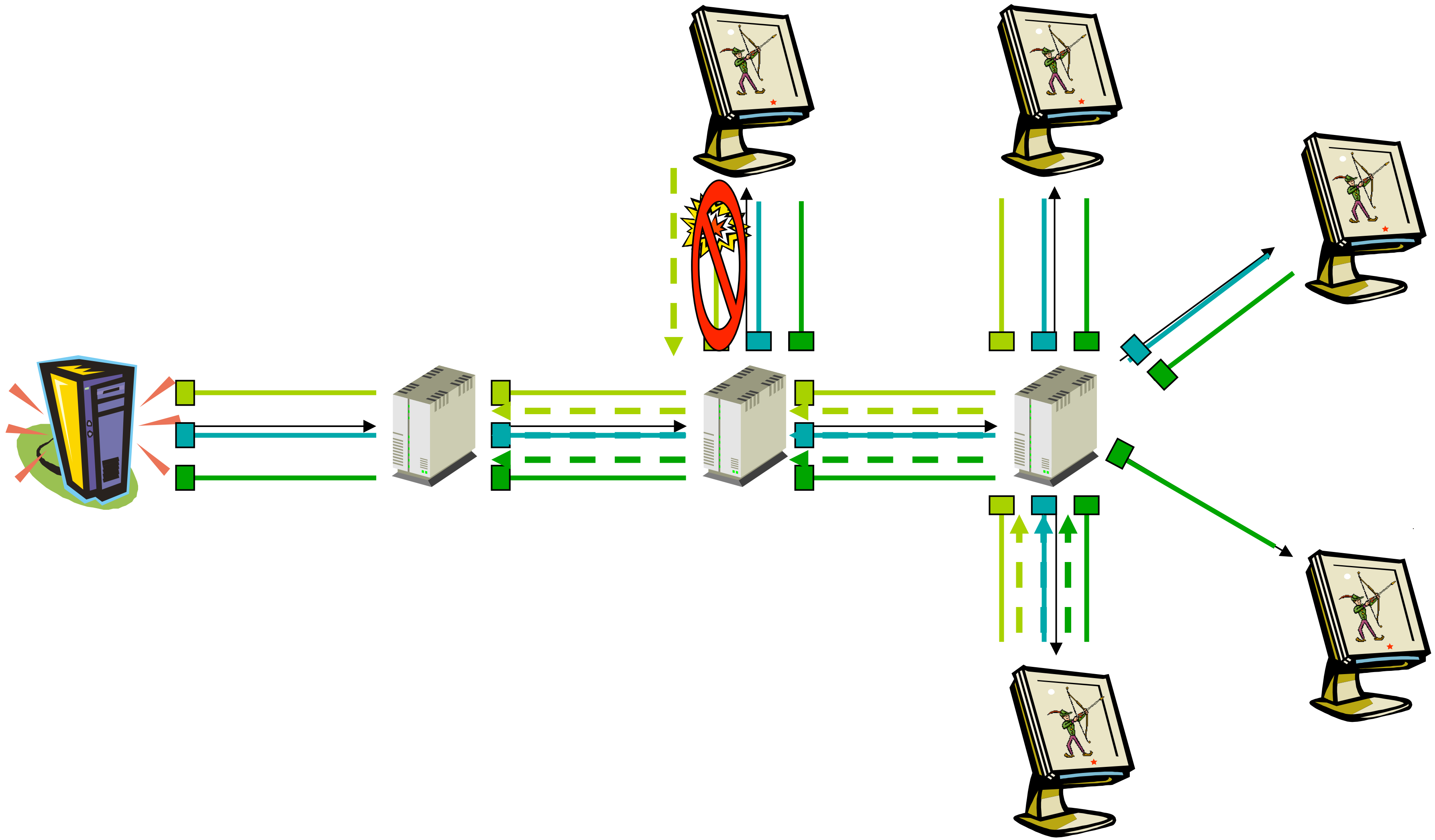
Coding ...: hierarchical layer coding



Receiver-driven Layered Multicast (RLM)

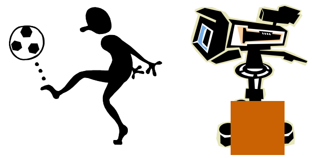
- Requires
  - IP multicast
  - layered video codec
- Operation
  - Each video layer is one IP multicast group
  - Receivers join the base layer and extension layers
  - If they experience loss, they drop layers (leave IP multicast groups)
  - To add layers, they perform "join experiments‟
- Advantages
  - Receiver-only decision
  - Congestion affects only sub-tree quality
  - Multicast trees are pruned, sub-trees have only necessary traffic

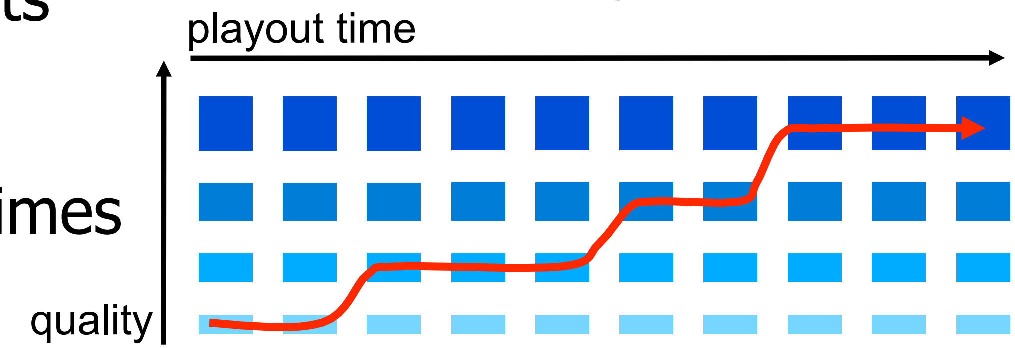
Receiver-driven Layered Multicast (RLM)



DAVVI

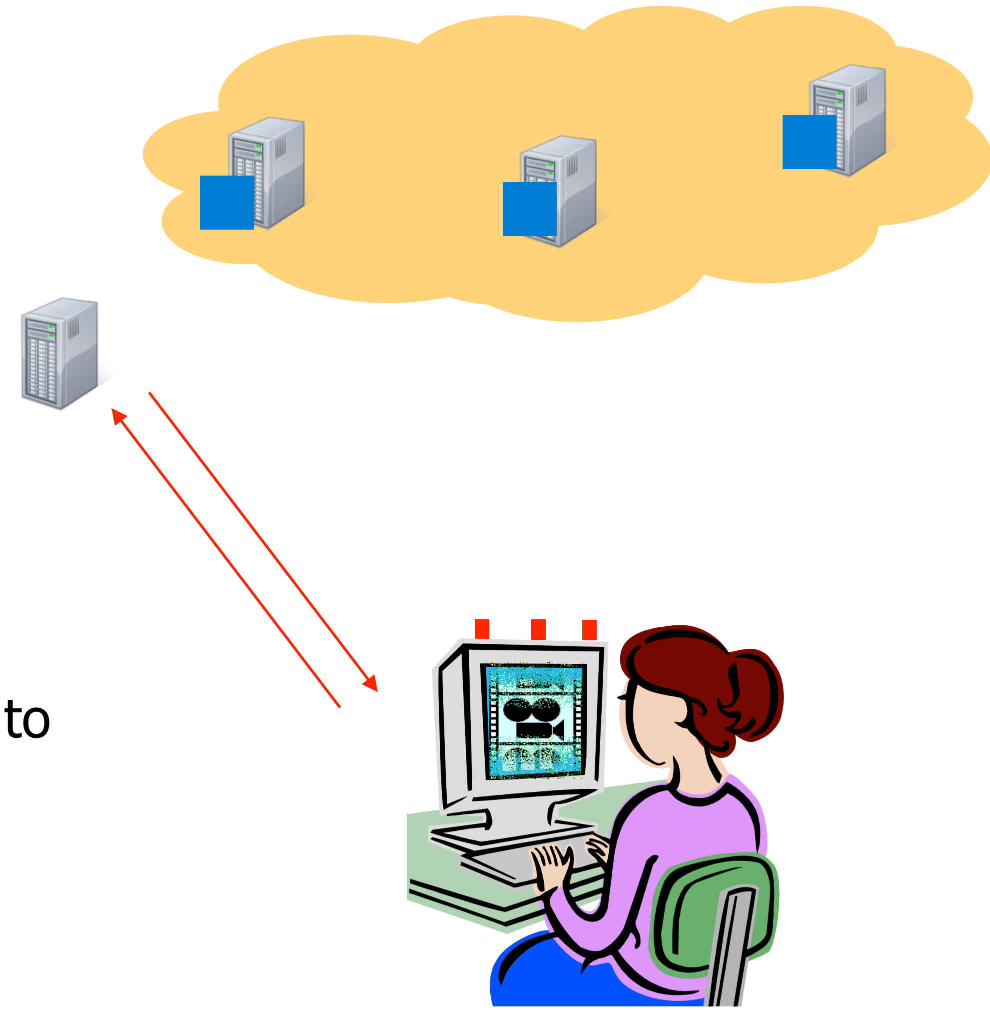
- Unmodified TCP
- All modern codecs possible
  - Have used MPEG-2, H.264+MP3
  - Needs new container format
- Divide a video into segments
  - 2 seconds are good
- Encode segments several times
  - At different quality levels





DAVVI

- For load-balancing and scaling multiple servers
- Downloads segments
- A tracker manages information about segment locations
- The user contacts the tracker for segment locations
- User sends **HTTP GET requests** to download **video segments**
- **Not so unlike Move Networks and Microsoft SmoothHD**
  - **Just faster ☺**



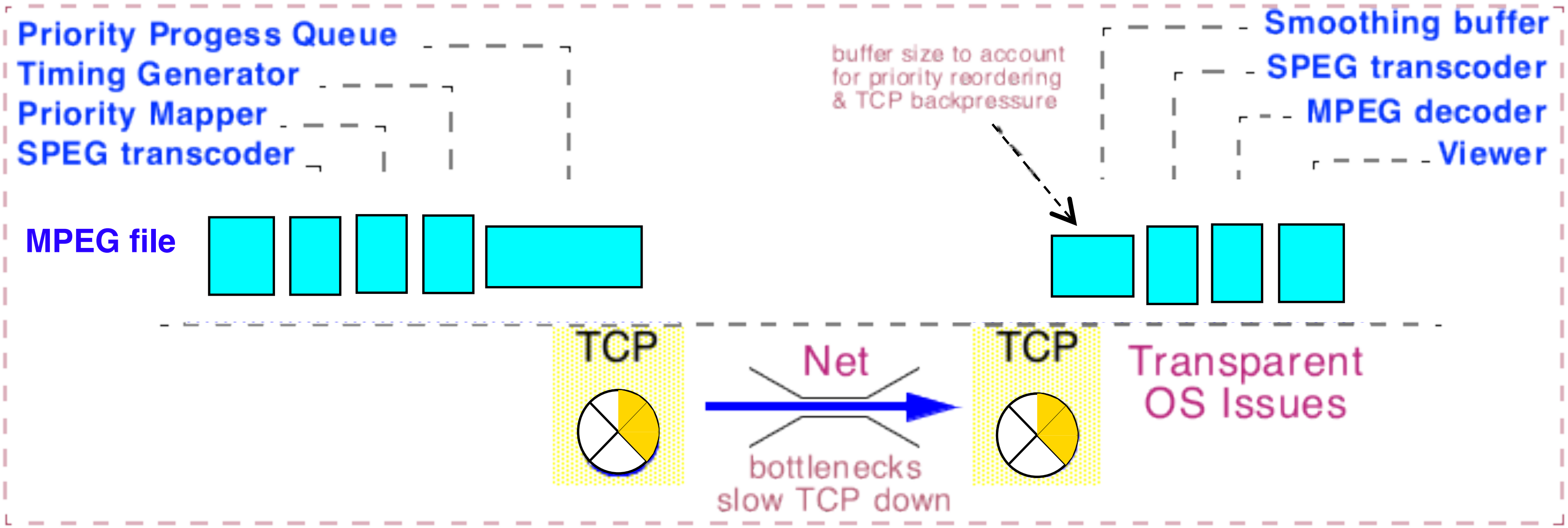
Priority Progress Streaming

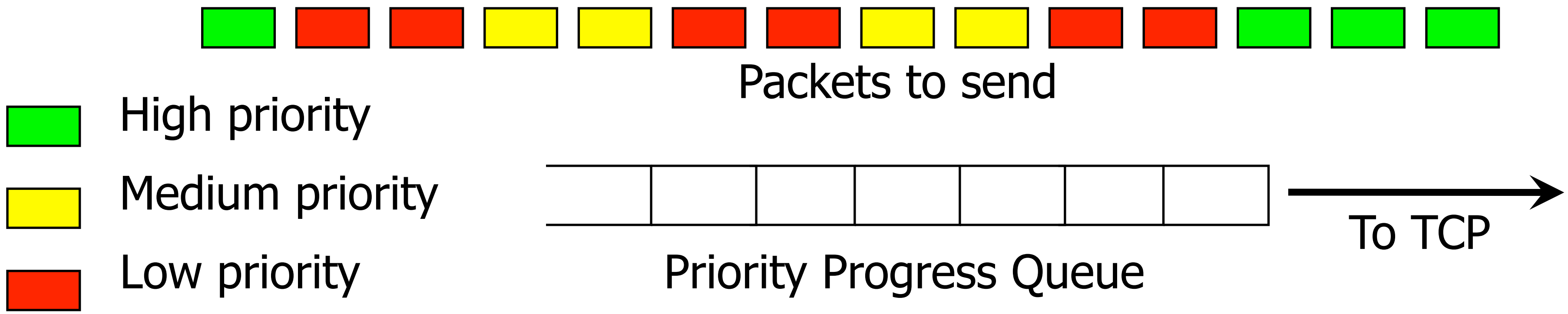
- Unmodified TCP (other transports conceivable)
- Unmodified MPEG-1 video-in (other encoding formats conceivable)

- Real-time video processing
  - Convert MPEG to Spatially Scalable MPEG (SPEG) – 10-25% overhead
  - Packetize SPEG to separate by frame and by SNR quality step
    - More variations conceivable: color, resolution
  - Assign priorities to SPEG packets
    - Dynamic utility curves indicate preference for frame or SNR dropping
  - Write SPEG packets in real-time into reordering priority progress queue

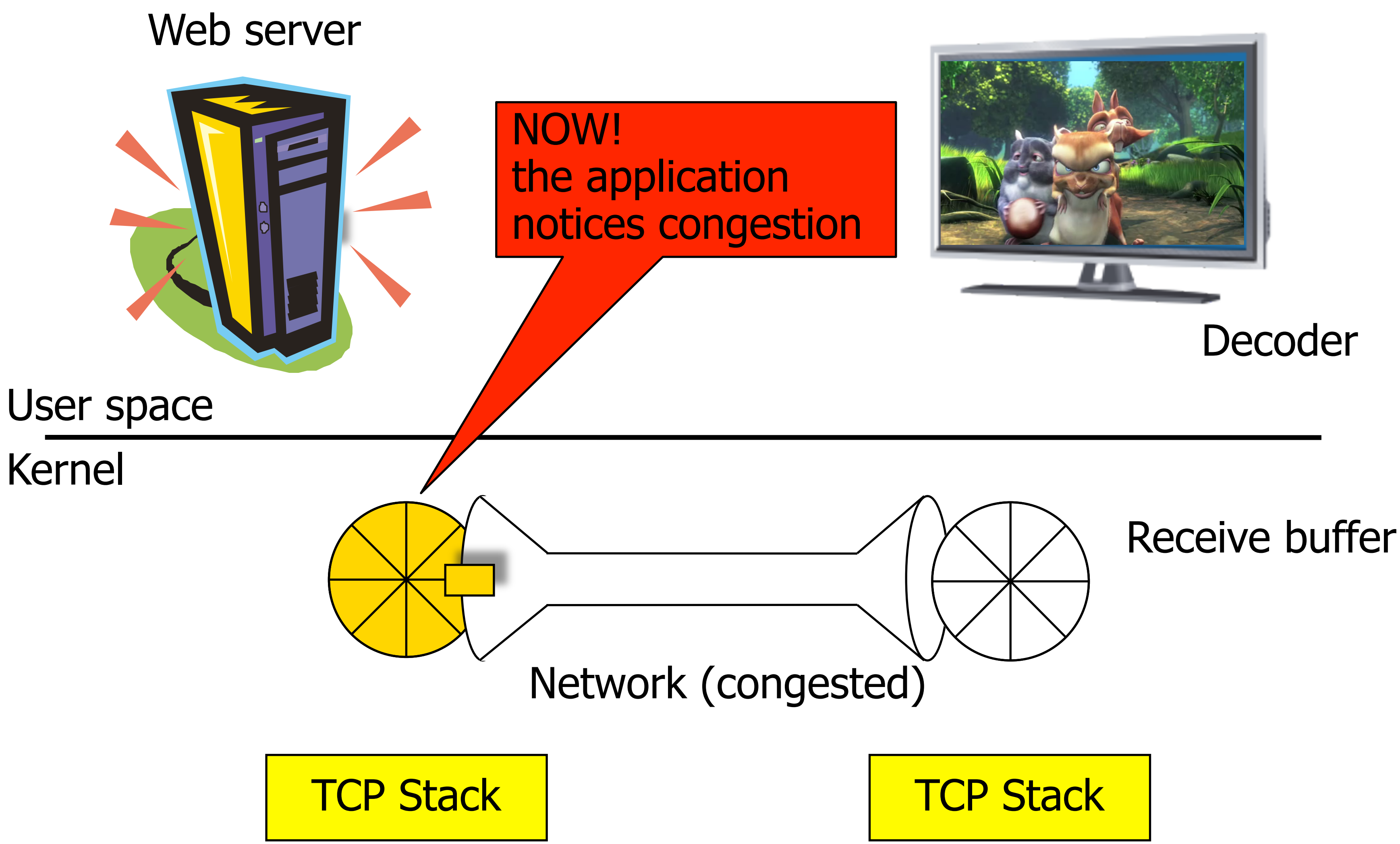
- Queues are long
  - Much longer than TCP max window
  - Dynamic adjustment allows fast start and dynamic growth
  - With longer queues
    - Total delay is increased
    - High priority packets win more often

Priority Progress Streaming





Paceline



Backpresure !

Paceline

- Try to estimate how full the TCP buffer is

- consider
  - number of bytes in flight at each RTT: $cwnd$
  - so: cwnd must be approx. $bandwidth/RTT$

- approach
  - recreate TCP ACK-mechanism in user space
  - send application-layer ACKs (P-ACKs)
  - estimate RTT
  - estimate bandwidth
  - don't feed TCP faster than $bandwidth/RTT$
  - slow down rate adaptation by computing $pressure = \dfrac{last - bw}{2 * long - term - avg - bw}$
  - estimate $cwnd$ development with pressure

$$cwnd = (1 - pressure) * last\_rtt * avg\_bw + pressure * avg\_rtt * avg\_bw$$

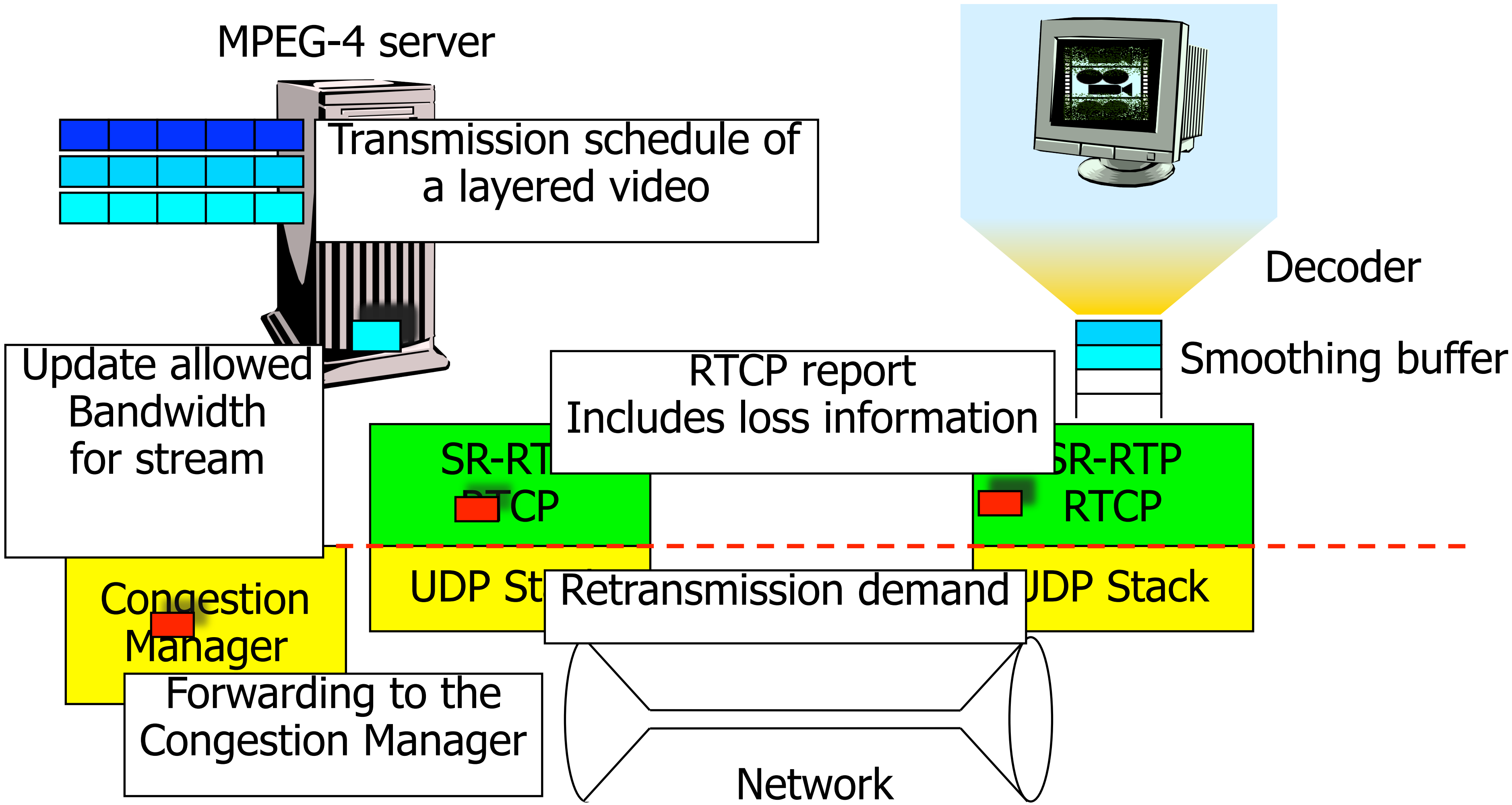
Selective Retransmission–RTP (SR–RTP)

- Features
  - Relies on a layered video codec
  - Supports selective retransmission
  - Uses congestion control to choose number of video layers

- Congestion Manager
  - Determines the permitted send rate at the sender
  - Uses TCP-friendly algorithm for rate computation

- Knowledge about encoding
  - Required at sender to select video layers to send
  - Required at receiver to
    - decode at correct rate
    - send NACKs

Selective Retransmission–RTP (SR–RTP)



Selective Retransmission–RTP (SR–RTP)

- Binomial Congestion Control
  - Provides a generalization of TCP AIMD

Increase

$$w_{t+RTT} = w_t + \alpha / w_t^k, \alpha > 0$$

Decrease

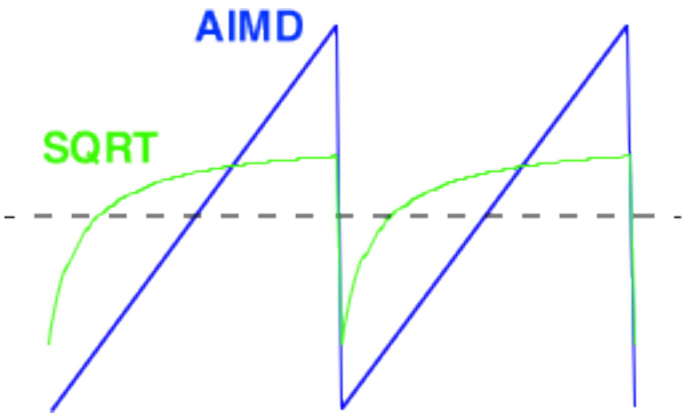
$$w_{t+RTT} = \beta \times w_t^l, 0 < \beta < 1$$

  - Congestion window size w_t depends on losses per RTT
  - TCP's AIMD: α = 1, β = .5, k = 0 and l = 1
  - k + l = 1: binomial congestion control is TCP friendly

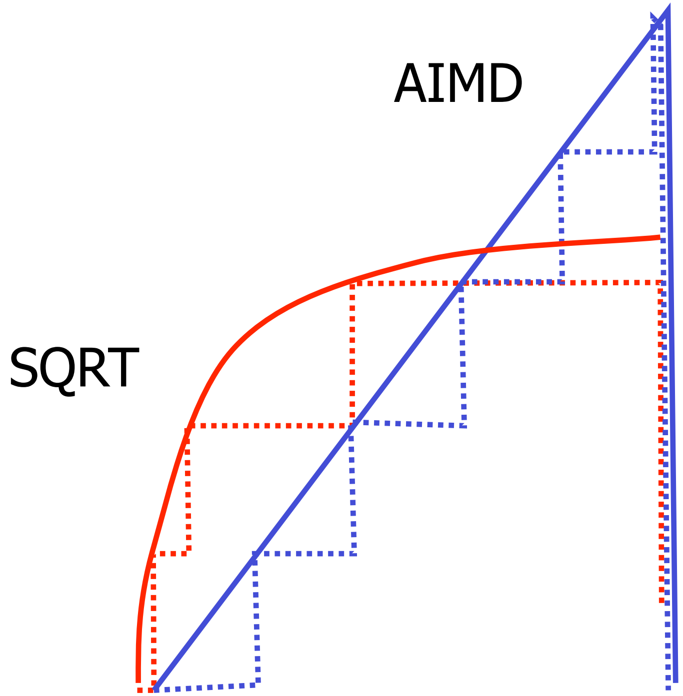
Nick Feamster and Hari Balakrishnan

Selective Retransmission–RTP (SR–RTP)

- **SQRT**
  - Special case of binomial congestion control
  - k=0.5, l=0.5
  - Name because $w^{0.5} = \text{sqrt}(w)$



- **Effect of SQRT**
  - Average bandwidth is like TCP's
  - Maximum is lower
  - SQRT covers a step function with less steps



On-demand streaming applications

- Smoothness is key
  - Use a lot of buffering
  - Don't surprise the application
  - Consume a limited amount of buffers
  - Try to make congestion control as smooth as possible

- Adaptive applications
  - Can by improved by this

- Next time: Interactive applications and QoS

Some References

1. Dorgham Sisalem, Henning Schulzrinne: "The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme", Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), July 1998
2. Charles Krasic, Jonathan Walpole, Wu-chi Feng: "Quality-Adaptive Media Streaming by Priority Drop", Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), June 2003
3. Charles Krasic, Jonathan Walpole: "Priority-Progress Streaming for Quality-Adaptive Multimedia", ACM Multimedia Doctoral Symposium, Ottawa, Canada, October 2001
4. Kurose, J.F., Ross, K.W.: "Computer Networking – A Top-Down Approach Featuring the Internet", 2nd ed. Addison-Wesley, 2003

- The RFC repository maintained by the IETF Secretariat can be found at http://www.ietf.org/rfc.html
  The following RFCs might be interesting with respect to this lecture:
  - RFC 793: Transmission Control Protocol
  - RFC 2988: Computing TCP's Retransmission Timer
  - RFC 768: User Datagram Protocol
  - RFC 2481: A Proposal to add Explicit Congestion Notification (ECN) to IP
  - RFC 1889: RTP: A Transport Protocol for Real-Time Applications
  - RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Control
  - RFC 2960: Stream Control Transmission Protocol
  - RFC 2326: Real Time Streaming Protocol
  - …