Z24: IP-based Transport

Mark Handley

IP-based Transport

The Internet is by far the largest network, so it's natural to want to transport multimedia over it.

Three main classes of application:

- **File transfer** (ftp, http, Kazaa, etc)
- **Streaming media** (RealPlayer, Windows Media Player, etc)
- **Interactive:** telephony (VoIP), videoconferencing, online gaming.

Application 1: File Transfer

Reliably transport a media file across the net.

Minimal delay guarantees:

- Start playing when file has completely arrived.
- Perhaps start playing when enough of the file has arrived that we think we'll be able to transfer the rest of it before we need to play it.

Reliability much more important than delay.

Application 2: Streaming Media

Both timeliness and reliability constraints.

- Some seconds delay before playout starts is acceptable.
- Once playout starts, we want it to be continuous.

Application 3: Interactive: Telephony, Conferencing, Gaming

Timeliness is critical.

- Low delay necessary for interactivity.
- 200ms round-trip is upper bound.

Reliability is secondary.

- Needs to be “good enough”.

IP Performance Guarantees

By default, the Internet provides “**best effort**” service.

It will try hard to get your data to the destination, but there are no guarantees.

- Packets can be *queued*, and hence delayed.
- Packets can be *lost* - usually when a queue overflows.
- Packets can be *re-ordered*.
- Packets can be *duplicated*.

***Conclusion:* internet multimedia is a non-starter without guaranteed quality of service?**

IP Performance - the reality

Typically the net is pretty good.

The web runs on TCP, and an ISP's customers complain if TCP doesn't work well enough.

- Packet loss rates of more than about 10% provide unacceptable web service too.
- Serious reordering or duplication mess up TCP too.
- Queuing delays of more than 1 second are rare: usually only on over-buffered tail circuits (eg DSL, modem).
  - Almost no queuing delays in the backbone.

Classic Transport Protocols

TCP (Transmission Control Protocol, RFC 793)

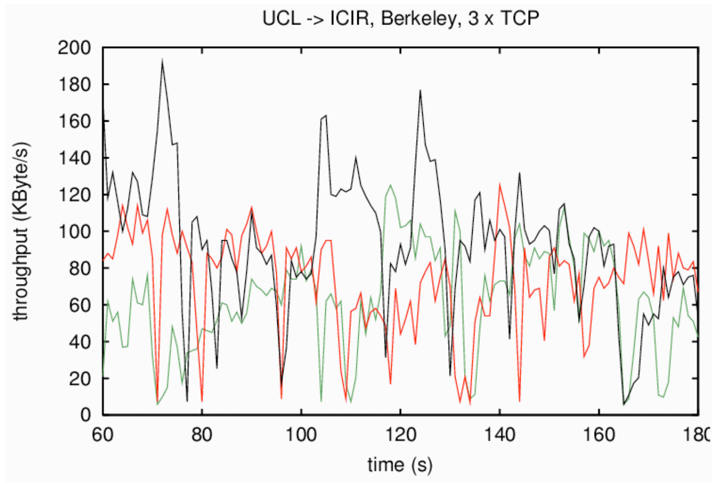
- provides a *reliable in-order bytestream* service.
- provides *congestion control*, so variable throughput.
- uses acknowledgements and retransmission of lost packets.

UDP (User Datagram Protocol, RFC 768)

- provides a *pure datagram service*.
- demultiplexing via ports, payload checksum.
- no protection against loss, reordering or duplication.

Note: http://www.ietf.org/rfc.html to obtain RFCs.

TCP Throughput



TCP vs UDP

- TCP gives you *reliability*, at the expense of having little control over when a particular part of the dataflow arrives.
- UDP gives you no protection, but gives you *precise control* over what gets sent when.

Which protocol for which application?

File Transfer

- Use TCP.

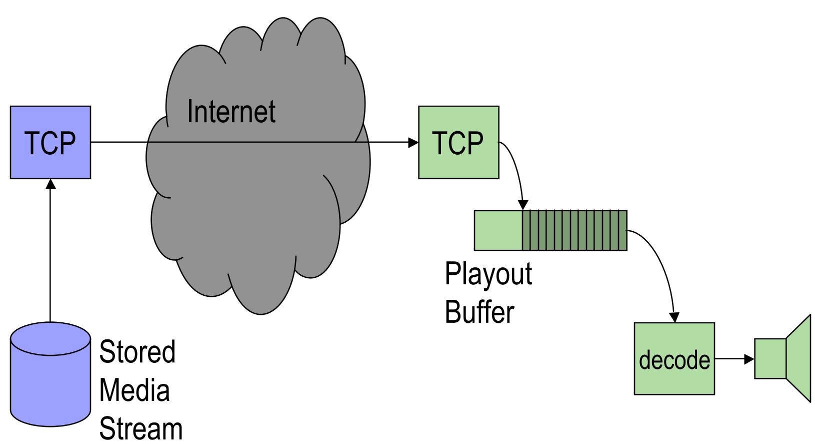
Streaming Media

- Can use TCP or UDP.
- Generally prefer UDP.

Telephony/Conferencing

- Must use UDP.

Streaming Media with TCP



Streaming Media with TCP

- Assume you've got a stored multimedia bitstream S bytes long, coded at R kbit/s.
  - Start the TCP transfer, measure achieved rate: r kbit/s.
  - Can start playout when:

    Remaining transfer time < playout time

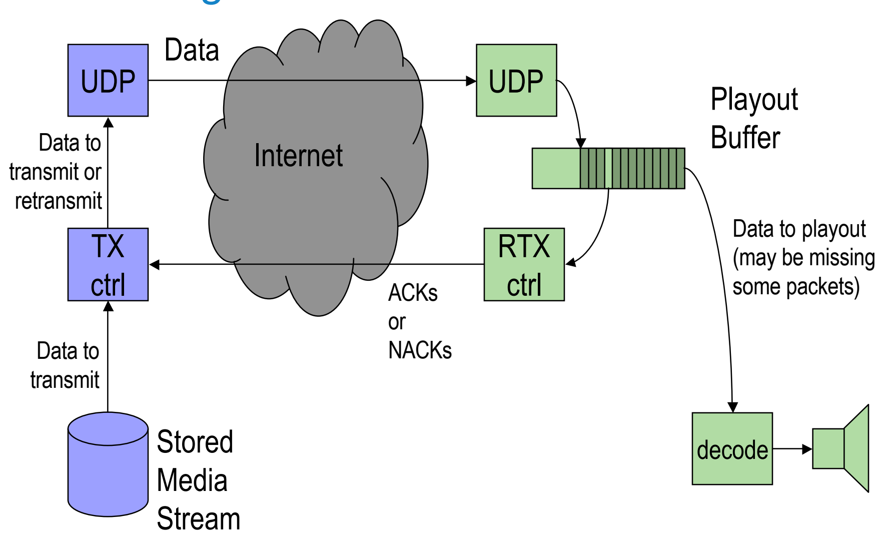
    ie. $8(S\text{-}T)/r \; < \; 8S/R$

    where T is the bytes that have already been transfered.
- Sometimes network performance will drop significantly.

  If the receiver playout buffer runs dry, need to stop playout and rebuffer more data until:

  remaining transfer time < remaining playout time

Streaming Media with UDP



Streaming Media with UDP

Assume (for now) that network has sufficient mean capacity.

Divide data into packets, add headers providing sequence numbers and/or data timestamps.

Transmit using UDP at the same rate the data will be played out at the receiver. At receiver, start playout after a small delay.

- Receiver sends NACKs indicating which packets were lost, and where its playout point is.
- Sender retransmits packets if there is still time to get them to the receiver before they need to be played out.
- Receiver attempts to mask any remaining loss.

TCP vs UDP for streaming media

- Delay before playout start with TCP is generally significantly larger.
  - TCP: absorb variation in throughput, arbitrary number of retransmissions.
  - UDP: absorb variation in latency, limited number of retransmissions.
- Usually stopping playout while “rebuffering” with TDP is more disturbing than slight degradation with UDP due to residual uncorrected packet loss.

TCP vs UDP for streaming media

- Practical issue:
  - TCP is usually easier to get through firewalls and NATs.
- General strategy (eg RealPlayer):
  - Try UDP first.
  - Fall back to TCP if UDP doesn't work.

Summary

- Three main types of multimedia application:
  - File transfer
  - Playback (streaming media)
  - Interactive (telephony/conferencing/gaming)
- Internet "best-effort" service isn't ideal for multimedia:
  - Loss, variable delay, variable capacity.
- Two transport protocols:
  - TCP when reliability is most important.
  - UDP when timeliness is most important.