COBALT®

Error Resilient Internet Video Transmission

Ciro A. Noronha, Ph.D.
Director of Technology, Compression Systems
Juliana W. Noronha
University of California, Davis

Motivation

- There are a number of protocols in use today to transport Video over IP.
- Since the “I” in IP stands for “Internet”, the Internet can (potentially) be used to transport Video over IP.
  - Low-cost contribution links!!
- However, not all Video over IP protocols are suitable for transporting Video on the Internet because:
  - **The Internet drops packets**
  - Video over IP is compressed and needs every bit
  - **Video over IP cannot take packet drops**
  - The Video over IP **protocol** has to handle this issue

Outline

"The nice thing about standards is that you have so many to choose from."

Andrew Tanenbaum, *Computer Networks*, 2nd ed., p 254

- Where does packet loss happen?
- How much packet loss is acceptable?
- What can we do about packet loss?
  - Protocol options
  - Theoretical analysis
  - Measurement Results
- Conclusions and recommendations

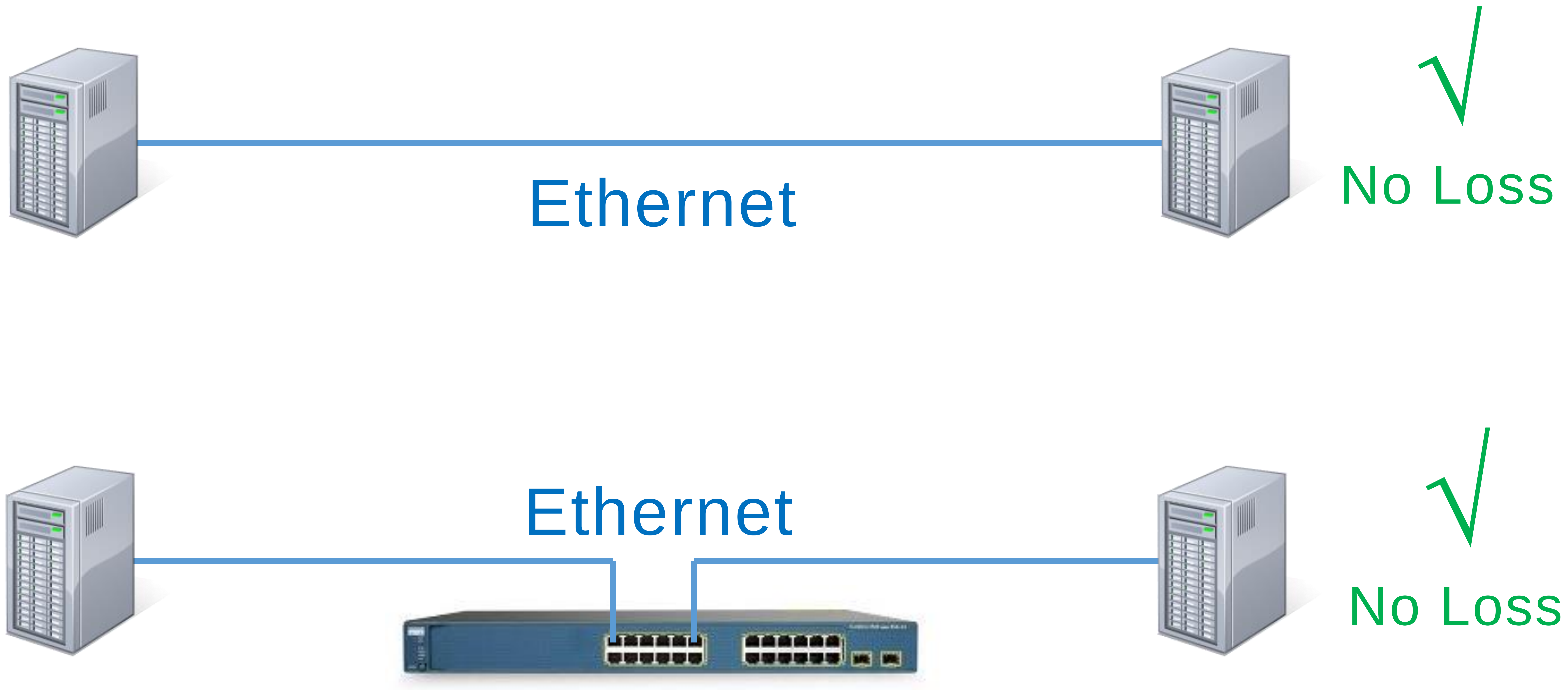
The Internet Protocol (IP)

The *Internet Protocol* defines an unreliable, connectionless, best-effort delivery mechanism for the Internet.

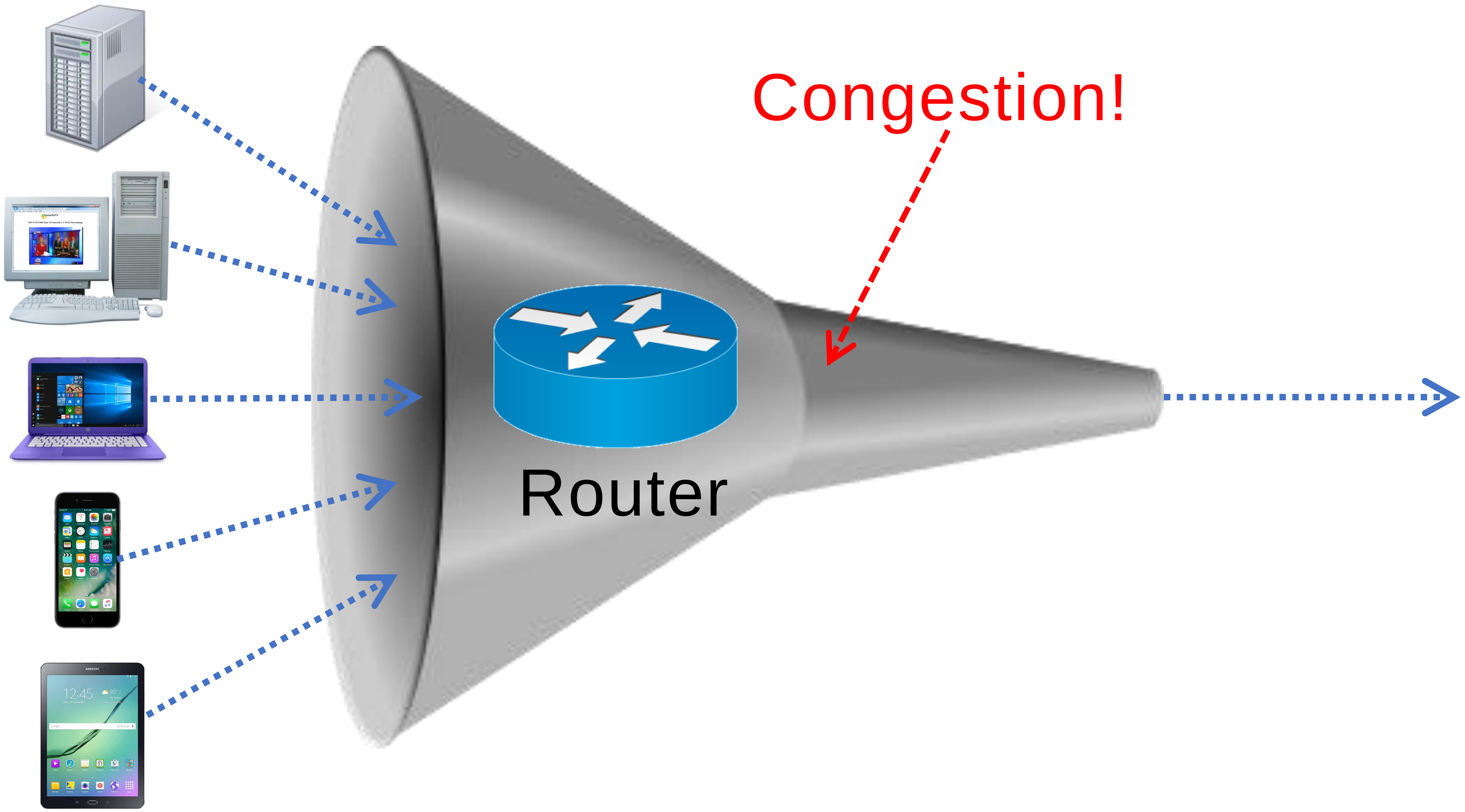
- **Unreliable**: packet delivery is not guaranteed
- **Connectionless**: packets are treated independently; multiple packets between two nodes may take different paths and arrive out-of-order
- **Best Effort**: packets are discarded when underlying networks fail or resources are exhausted

"I am going to try my best to deliver your packet, but if I cannot, no hard feelings."

Where are packets lost?



So, where are packets really lost?



Congestion!

- Congestion happens when the traffic wanting to go out a link exceeds the capacity of that link
- Routers have buffers that will accommodate small fluctuations
- Once the buffer is full, packets are dropped
  - A packet will traverse multiple routers and links – this can happen anywhere in the path
  - Normally, packets are dropped in bursts or blocks
  - This is the "best-effort" aspect of the Internet

Can't this be fixed with traffic priorities?

- **In theory, yes.**
  - Video traffic can be "marked" so it is recognizable at the router
  - Router can be configured to give priority to video packets
    - If there is congestion, other traffic is dropped
- **However:**
  - You can do this if you own and control all the routers in the path
  - You don't own and control the routers in the Internet
  - Internet will ignore all packet markings

What is an "acceptable" packet loss?

- Video compression works by removing redundancy from the content
  - Every bit of compressed video is very important
- There is a simple way to look at the effect of packet loss:
  - Assume that every packet that is dropped by the network causes a noticeable glitch in the video
    - A block of packets dropped together causes one glitch
  - Decide how many glitches per (day/hour/minute) is acceptable to you

Some numbers

Assume a 4 Mb/s stream, with 1316-byte packets

Dropping one packet in	Produces a glitch every
1,000	2.6 seconds
10,000	26 seconds
100,000	4 minutes 23 seconds
1,000,000	44 minutes
10,000,000	7 hours 19 minutes

In order to achieve reliable operation on the Internet, a network protocol is needed to "recover" in some way the packets that have been lost.

The Network Protocol Tradeoff

- Fundamentally, there is a tradeoff between LATENCY and PACKET LOSS RESILIENCY:
  - Decoders cannot “wait forever” – packets have expiration dates
  - You can give yourself time to deal with packet loss by pre-buffering before the decoder – the more time you give yourself, the better job you can do to recover from lost packets
  - However, many applications (e.g., contribution) have latency limits

The Network Protocol Tradeoff

Protocol Latency

Gives you time to recover from lost packets – the more time you have, the better job you can do!

Protocol Basics

End-to-end IP applications run on top of one of two protocols:

- User Datagram Protocol (UDP)
  - "Raw" network service
  - Packets are delivered as fast as possible, but may be dropped
- Transmission Control Protocol (TCP)
  - "Reliable" network service
  - Flow control (bad for encoders, unless rate changes on the fly)
  - Unbounded latency

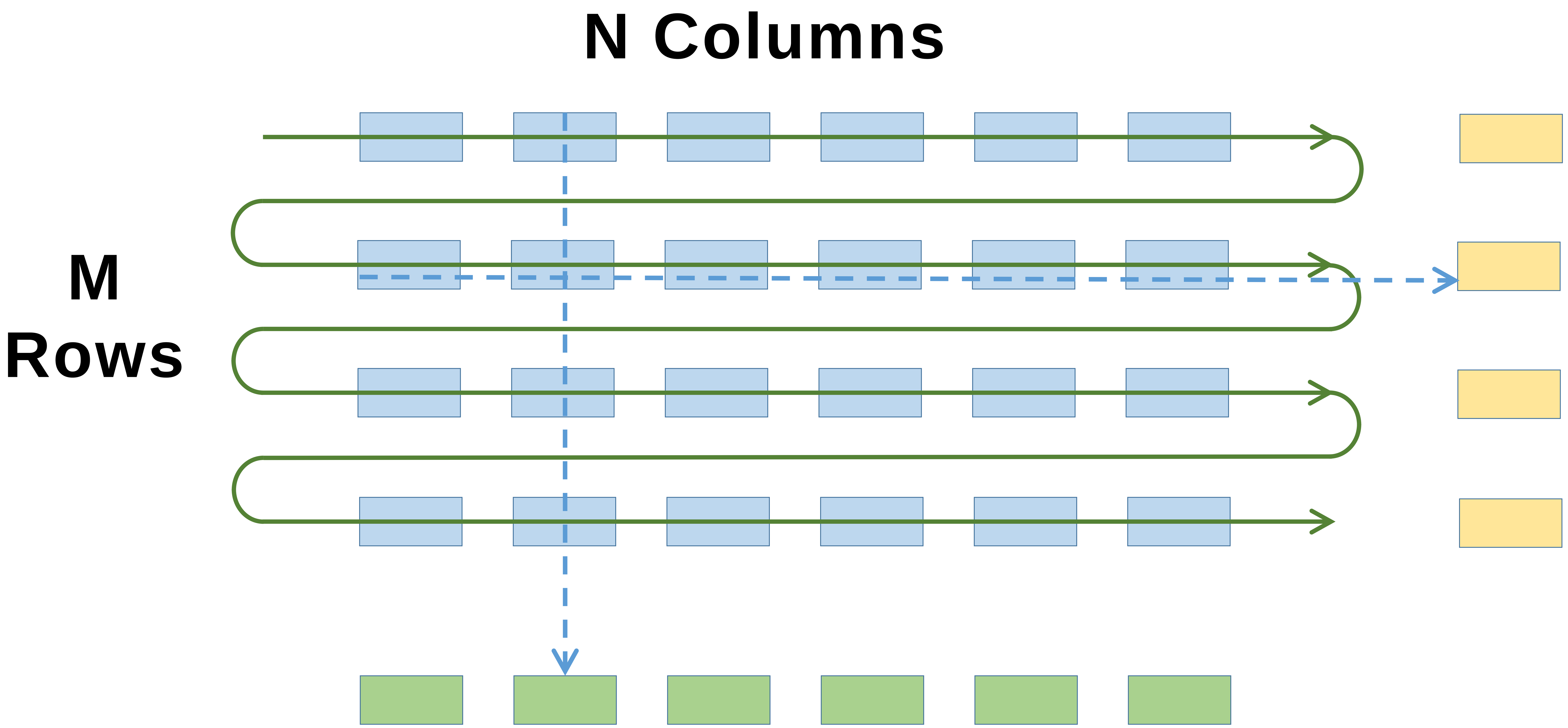
Protocol Roadmap

- We are limiting this discussion to protocols:
  - That will work over the Internet
  - That have no limitations on media transport
- Roadmap:
  - UDP based:
    - RTP plus SMPTE-2022 FEC
    - ARQ
  - TCP based:
    - HLS and similar variants

RTP plus SMPTE-2022 FEC

- Basic idea:
  - Transmit the video using RTP
    - That gets you timestamps and sequence numbers
    - Sequence numbers let you know when packets were dropped
  - Transmit “extra” FEC packets
  - If packets are lost in the network, it ***may*** be possible to rebuild them from the received packets and FEC packets:
    - For each N packets send 1 FEC packets
    - If there is one loss in this set of N+1 packets, it can be corrected
  - Use a matrix arrangement to deal with burst losses

FEC Illustration



Row FEC (optional): can recover single packet losses in each row

Overhead:

- N column packets
- M row packets

For each NxM video packets

Column FEC: can recover a burst of up to N successive lost packets every NxM packets

Some FEC Numbers

Columns	Rows	Recovery Capability	Overhead	Latency @ 2 Mb/s	Latency @ 10 Mb/s
5	5	5 pkts every 25	20%	263 ms	53 ms
10	5	10 pkts every 50	20%	526 ms	105 ms
20	5	20 pkts every 100	20%	1052 ms	211 ms
10	10	10 pkts every 100	10%	1052 ms	211 ms

ARQ

- ARQ stands for:
  - **A**utomatic **R**epeat re**Q**uest
  - **A**utomatic **R**epeat **Q**uery
- This is the generic name for a number of retransmission strategies in the face of packet loss
  - Standard TCP uses a couple of ARQ variants
- In video transmission, the most useful variant is “Selective Retransmission” (NACK-based)
  - If you don’t hear from me, everything is OK
  - If I miss anything, I let you know and you resend just that
- ARQ implementations in industry today do not interoperate due to lack of standards

Cobalt RTP/ARQ

- Use RTP as the base video transmission layer
  - Compatible with all professional IRDs (minus the packet loss correction)
  - Packet losses are detected using sequence numbers
- Use the RTCP NACK message from RFC-4585 to request retransmission of lost packets
  - One NACK message can request up to 17 packets
- It is possible to build a complete ARQ solution using only published standards with no proprietary methods

ARQ Illustration

ARQ Notes

- The ARQ latency is at least one network round trip delay
  - Allowing multiple round trip delays allows the receiver to retry a packet multiple times
  - Usual Latency x Reliability tradeoff
- The ARQ overhead is a function of the packet loss
  - If there is no packet loss, there is zero overhead
  - Overhead increases with packet loss, as lost packets are retransmitted

HTTP Live Streaming (HLS)

- HLS is a protocol designed by Apple to provide streaming using a standard (unmodified) web server
- Implemented primarily on mobile devices
- The video stream is divided into “chunks” of a few seconds each
- The decoder downloads the chunks as files from the web server with standard HTTP transactions, using a playlist
- Protocol supports adaptive streaming (multiple bit rates/resolutions)

HLS Illustration – single stream/profile

HLS Details

- Characteristics:
  - Very high latency: 3-4 times the chunk size (which varies from 2 to 30 seconds)
  - Extremely robust (uses TCP and HTTP)
  - Can potentially survive short network outages
  - Can easily scale to a large number of destinations
- Probably one of the most robust transports available if you can afford the latency
- Supported as a transport mechanism in the Cobalt encoder/decoder series

A little probability and statistics…

- Assume independent loss probability for each transmitted packet (binomial distribution)
- Calculate the rate of packets still lost *after* correction with statistical analysis
- This allows us to theoretically compare the performance of the various protocols and settings
- Our variables are:

R = number of requests (ARQ)

N = number of packets per row (FEC)

M = number of packets per column (FEC)

Percentage Packet Loss (p_c) for ARQ and Column FEC
100.0000000000
0.0100000000
0.0000010000
0.0000000001
Residual Packet Loss Percentage (p_c) after Correction
0.1
1.0
10.0
Percentage Packet Drop (p)
Number of Requests/Packets
N = 10
N = 20
N = 5
R = 1
R = 2
R = 4

Percentage Packet Loss (p_c) for ARQ and Row-Column FEC

Overhead

- In both scenarios, additional packets are transmitted:
  - FEC sends additional FEC packets
  - ARQ sends retransmissions
- We can also model the overhead of the various protocols and settings

$$overhead = \frac{number\ of\ extra\ packets\ sent}{total\ number\ of\ original\ packets}$$

Overhead for ARQ and FEC
Percent Overhead
10.0
1.0
0.1
0.1
1.0
10.0
Percentage Packet Drop (p)
Number of Requests/Packets
M = 10, N = 10
M = 5, N = 20
M = 5, N = 5
N = 10
N = 20
N = 5
R = 1
R = 2
R = 4

Field Test Data

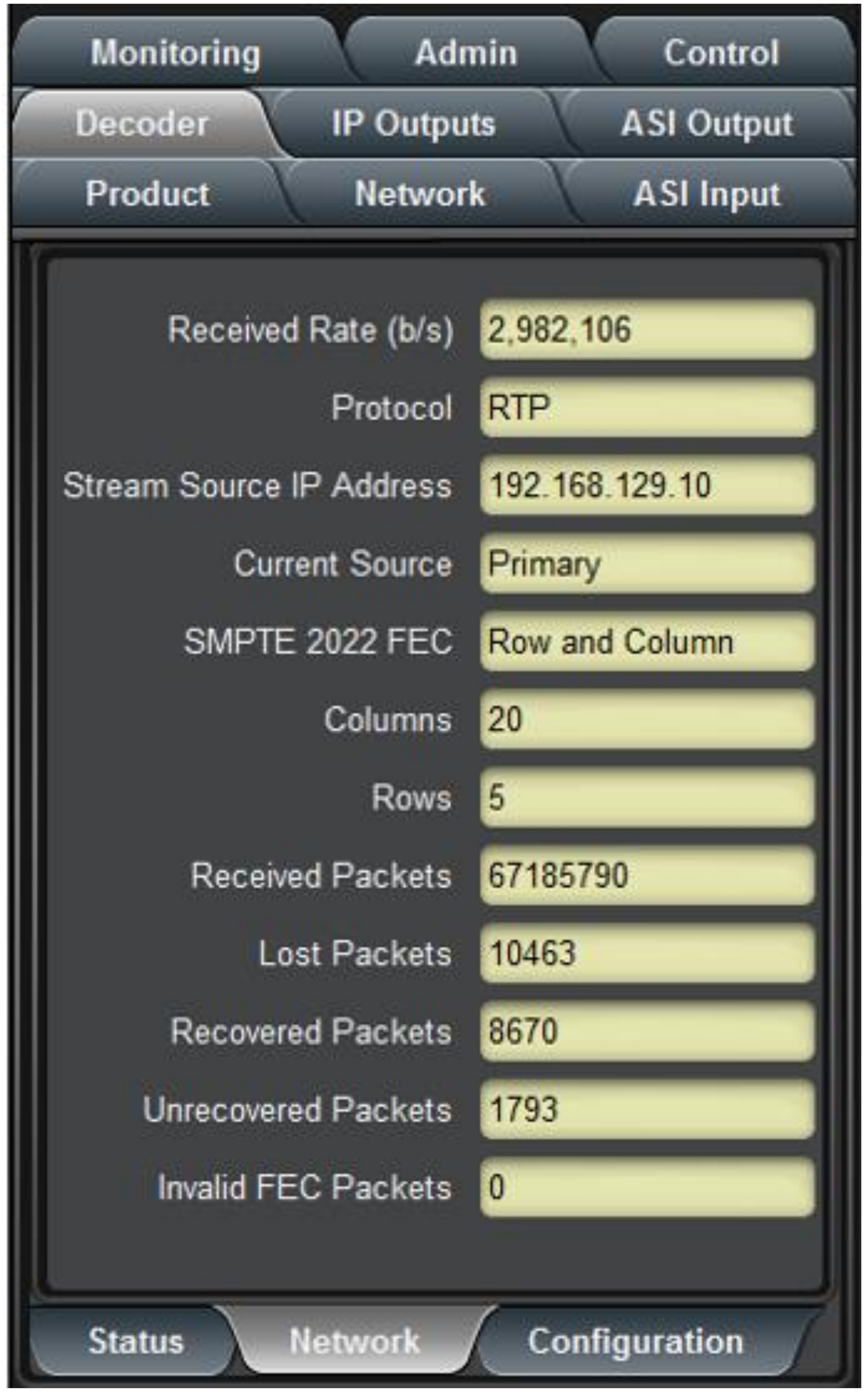
- Locations:
  - Santa Clara, CA
  - Champaign, IL
- ISP: Comcast
- Network Round Trip Time: 75 ms
- Number of hops: 12
- Target bit rate: 3 Mb/s
- Equipment:
  - 9223 Encoder
  - 9990-DEC Decoder

Initial Link Characterization

Custom software was used to receive the video stream and measure statistics

Test Duration	25 hours 42 minutes
Total Packets	26,381,219
Dropped Packets	8,187
Network Packet Loss	0.031%
Packet Loss Instances	2,464
Average Packet Drop	3.3 packets
Max Packet Drop	169 packets
Network Glitch Interval	37.5 seconds

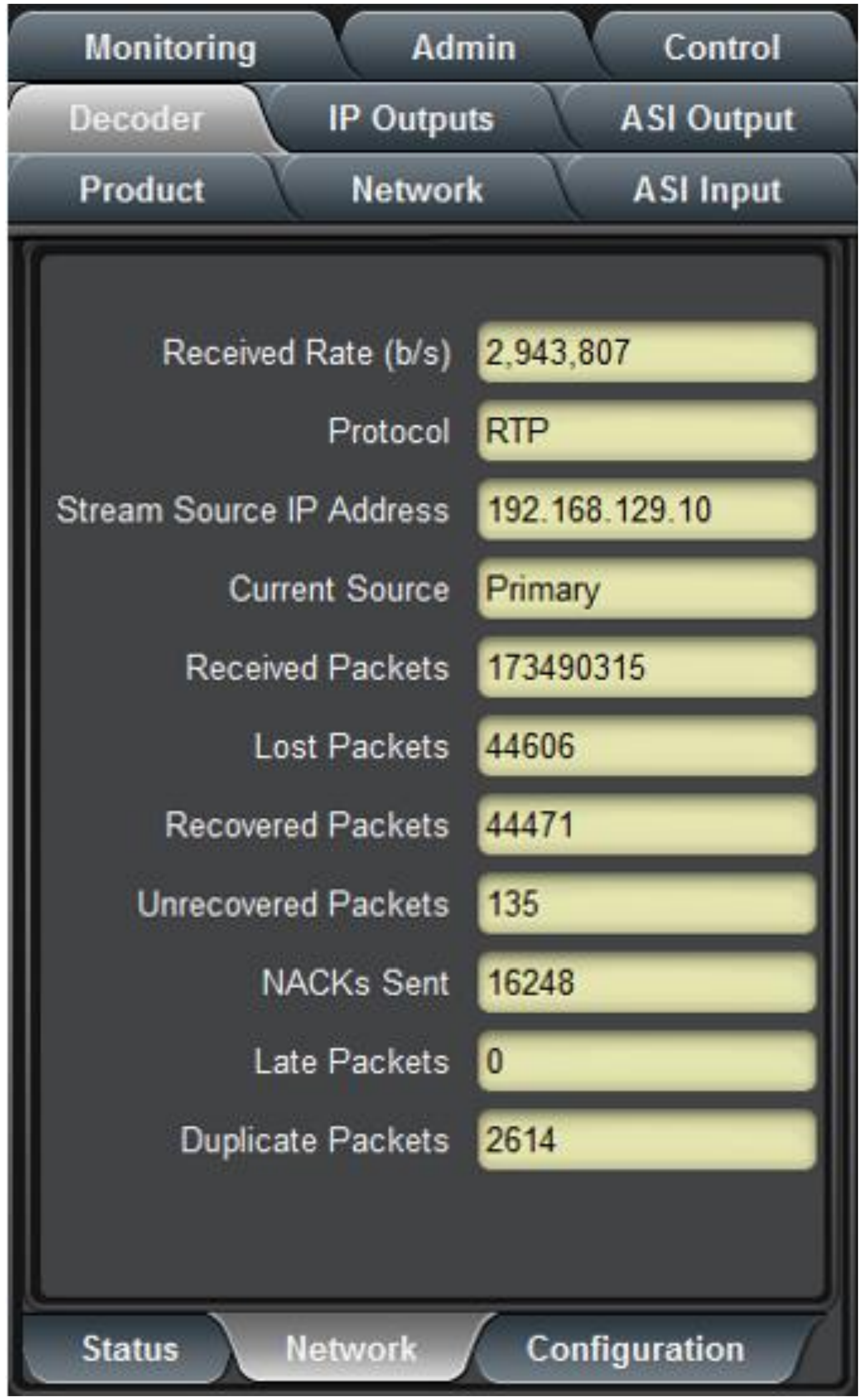
RTP/SMPTE-2022 Test Data



Parameters: 20x5 matrix, row and column

Test Duration	65 hours
Test Start Date	05/19/17, 3:50PM
Network Packet Loss	0.0158%
Corrected Packet Loss	0.0027%
Correction Ratio	83%
Bandwidth Overhead	25%
Network Glitch Interval	1 minute 13 seconds
Corrected Glitch Interval	7 minutes 12 seconds
Protocol Latency	702 ms

RTP/ARQ Test Data



Parameters: up to 4 retries allowed

Test Duration	169 hours
Test Start Date	05/24/17, 12:30PM
Network Packet Loss	0.0257%
Corrected Packet Loss	0.000078%
Correction Ratio	99.7%
Bandwidth Overhead	0.027%
Network Glitch Interval	46 seconds
Corrected Glitch Interval	4 hours 7 minutes
Protocol Latency	400 ms

FEC/ARQ Comparison

Scaling:

- Latency
  - ARQ latency is constant
  - FEC latency decreases with increasing bit rate
- Overhead
  - ARQ overhead will increase with packet loss
  - FEC overhead is constant

Parameter	2022 FEC	ARQ
Network Packet Loss	0.0158%	0.0257%
Corrected Packet Loss	0.0027%	0.000078%
Correction Ratio	83%	99.7%
Bandwidth Overhead	25%	0.027%
Network Glitch Interval	1 minute 13 seconds	46 seconds
Corrected Glitch Interval	7 minutes 12 seconds	4 hours 7 minutes
Protocol Latency	702 ms	400 ms

So, which one do I choose?

	RTP plus SMPTE 2022 FEC	RTP plus ARQ	HTTP Live Streaming
Latency	Moderate (less than 1 sec)	Moderate (less than 1 sec)	Very High (multiple seconds)
Overhead	High	Very Low	Very Low (uses TCP)
Correction Capability	Poor	Very Good	Excellent
Standard?	Yes	No (*)	Yes
Ease of setup	Simple	Moderate	Trivial

Standardization Efforts

- The standard solutions today are limited:
  - SMTPE-2022 does not work well on the Internet
  - HLS works well but has very high latency
- The **Video Services Forum** (VSF) has started a group called **RIST** (Reliable Internet Stream Transport) to come up with a protocol
  - Group launched during the Feb 2017 VSF meeting
  - Face-to-face organizational meeting during NAB 2017
  - Target is to have a standard by NAB 2018
  - Cobalt Digital is active in this group

THANK YOU!