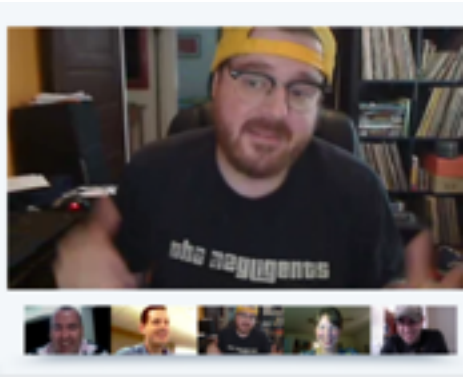
Video Telephony for End-consumers: Measurement Study of Google+, iChat, and Skype

Yang Xu, Chenguang Yu, Jingjiang Li and Yong Liu
ECE Department
Polytechnic Institute of New York University

- Video Conferencing becomes more and more popular among end-consumers
  - Skype, Google+ Hangout, iChat
- Challenges of realtime voice/video transmission over Internet:
  - high-rate (>500kbps)
  - low-delay (<350ms)
  - users highly sensitive to voice/video quality degradation



p Key Questions to be answered

- System Architecture
- Video Generation and Adaption
- User perceived Voice/Video Delay Performance
- Loss Recovery

p Challenges

- all three systems use proprietary protocols,
- encrypt data and signaling,
- very little public information about their design choices

p Methodology:

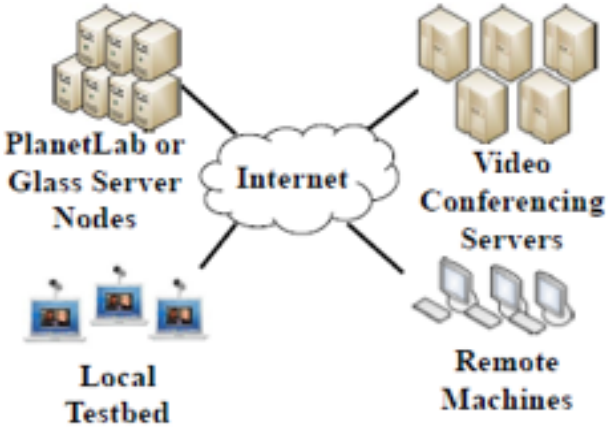
- Measure each system as black-box through controlled experiments

1. probe the black-box under different network conditions,
2. record those systems' behaviors,
3. analyze recorded traces to infer their designs.

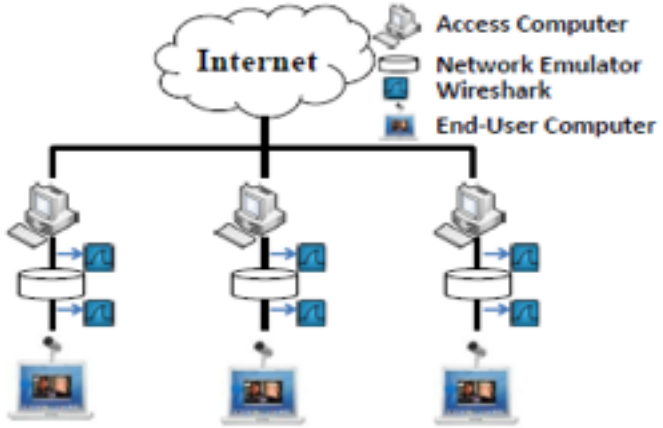
- Agenda
  - Skype Two-party video calls (Jan 2011 – June 2011)
  - Multi-party Video Conferencing (June 2011 – May 2012)

pTestbed

- Network emulation:
  - Network emulator – BW capacity, propagation delay, packet loss.
- Data collection:
  - TCPDump for packet-level information
  - Store application display window for video-level information
- Video call emulation:
  - Repeatable video sequence – Akiyo (So that the experiment results could be comparable)
  - Inject through virtual video camera tool



(a) Overall Testbed



(b) Local Testbed

pVideo Conferencing Topology

- iChat: Central P2P
- Google+: Server-centric
  - each participator connects to his local server.
- Skype: Hybrid
  - two-party call: voice and video using P2P
  - multi-party: voice using central P2P, video relayed by servers
  - video relay servers all located somewhere close to NJ/NY area

(a) iChat

(b) Google+ Hangout

(c) Skype multi-party

➢Conferencing Server Placement (1)

pGoogle+: Locates servers around the world

✓Table 1 shows RTT between measurement locations and their corresponding servers.

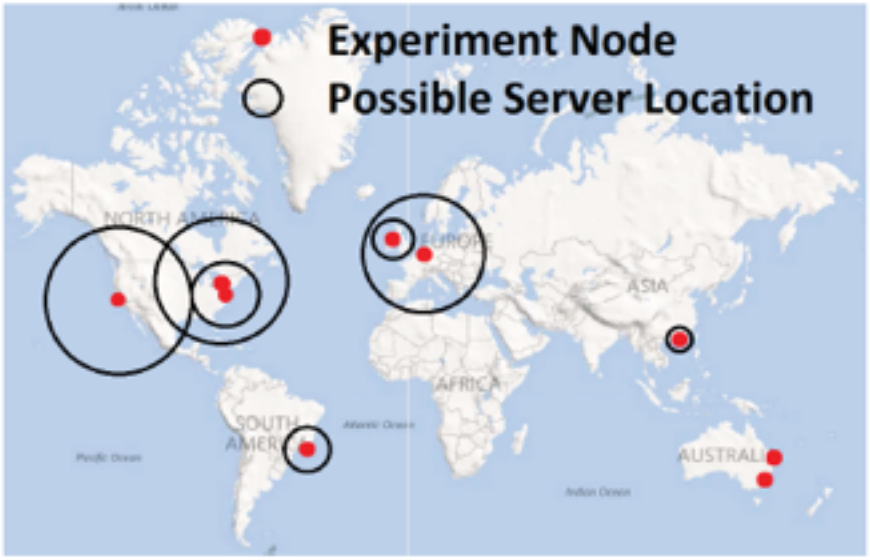


Figure 3: Google+ Experiment Nodes and Corresponding Server Locations

Table 1: Google+ Hangout Server IP Addresses

Friend Location	Server IP	RTT (ms)
Hong Kong, CHN	74.125.71.127	3.49
Armagh, UK	173.194.78.127	8.88
Rio de Janeiro, BRA	64.233.163.127	9.02
New York, USA	173.194.76.127	14.2
Aachen, DE	173.194.70.127	20.00
Toronto, CA	209.85.145.127	26.76
San Jose, USA	173.194.79.127	28.89
Brisbane, AU	72.14.203.127	147
Canberra, AU	74.125.31.127	147

➢Conferencing Server Placement (2)

pSkype: All servers' IP addresses in the subnet 208.88.186.00/24, some place around New Jersey/New York area like Table 2 shows.

Table 2: RTT to Skype Video Relay Servers

PlanetLab/Glass Server Locations	RTT (ms)
New Haven, USA	17.4
New York, USA	25.9
Washington, USA	31.3
Vancouver, CA	61.41
San Jose, USA	67.5
London, UK	99.2
Guayaquil, EC	111
Saarbrucken, DE	138
Oulu, FIN	151
Rio de Janeiro, BRA	176
Tel Aviv, IL	207
Canberra, AU	222
Hong Kong, CHN	226
Brisbane, AU	266
West Bengal, IN	324

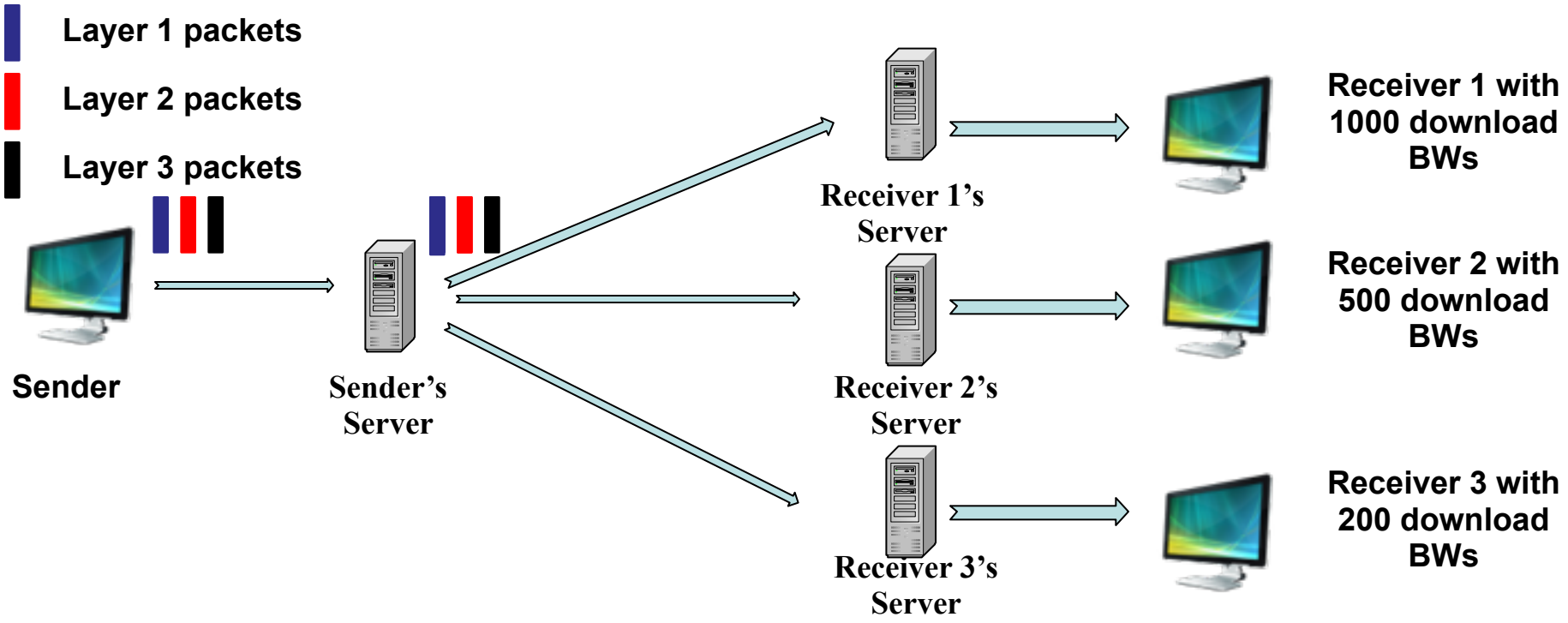
pVideo Generation and Adaptation For Receiver Heterogeneity

- Question:
  - If receivers' download BWs are different, what will sender do?

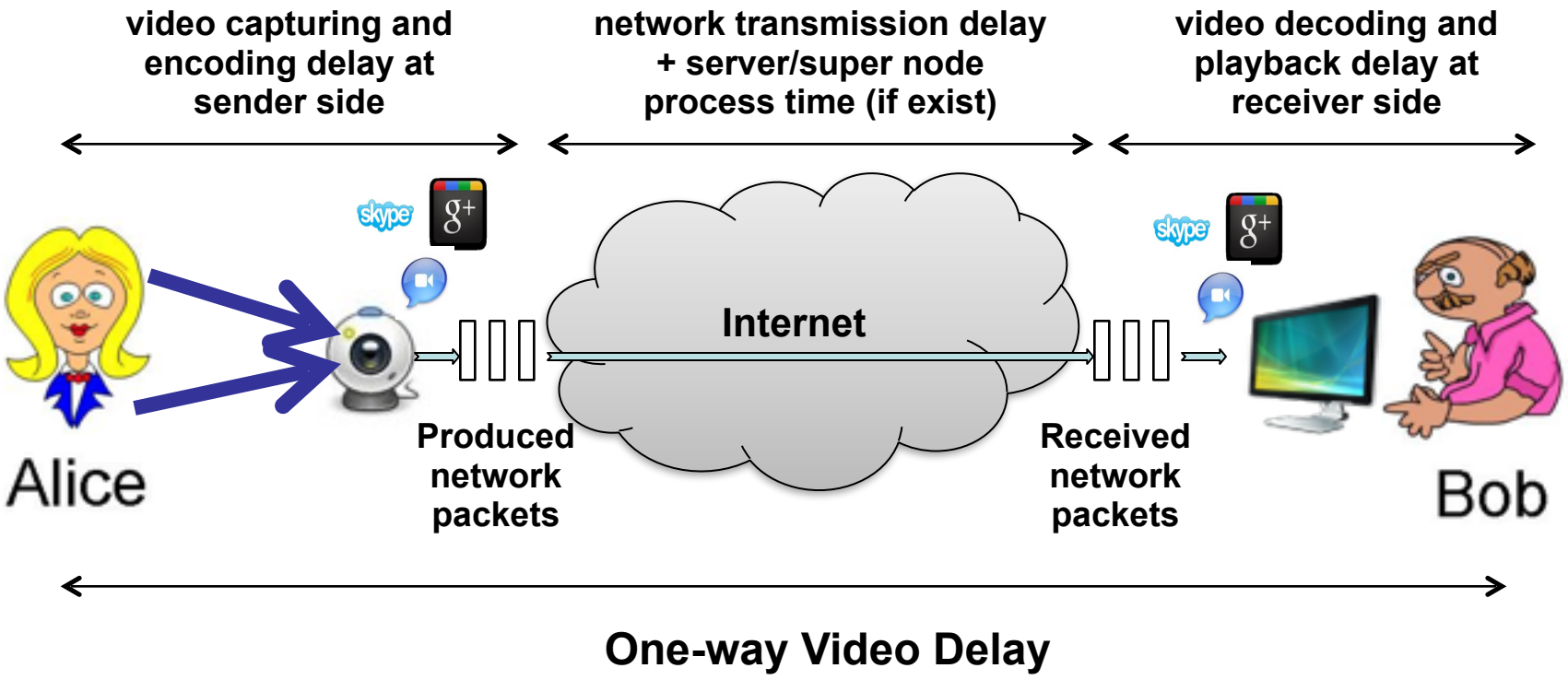
- Results:
  - iChat: one-version encoding
    - heterogeneous receivers always receive same video version (determined by the weakest receiver).
  - Skype: source-side multi-version encoding
    - source generates multiple video versions and send all of them to relay server → large source upload bandwidth overhead
    - a receiver downloads from relay server a version matching his BW

pVideo Generation and Adaptation For Receiver Heterogeneity

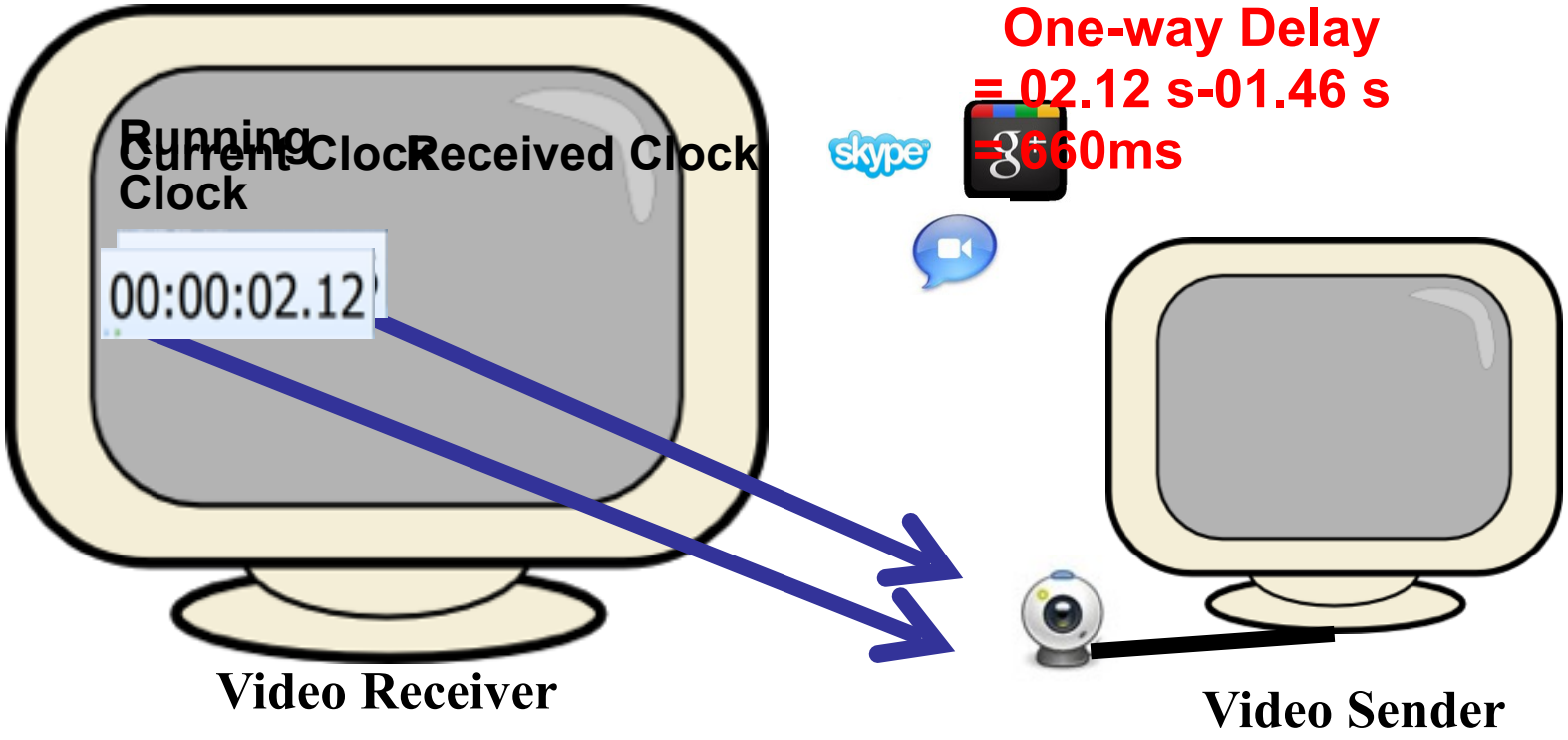
- Google+ employs layered video coding
  - source video is encoded into multiple layers;
  - high layers decodeable iff lower layers are received
  - receivers get different subsets of layers, more layers → better quality



- End-to-end voice/video delay perceived by users
  - Video Delay Components:
  - Question:
    - Which component accounts for a significant portion of delay?



- End-to-end Voice/Video Delay
  - One-way Video Delay Measurement:
    - Put screens of two computers side by side. Use a running stopwatch as the video source.
    - Difference of these two clocks are the one-way delay



pEnd-to-end voice/video Delay

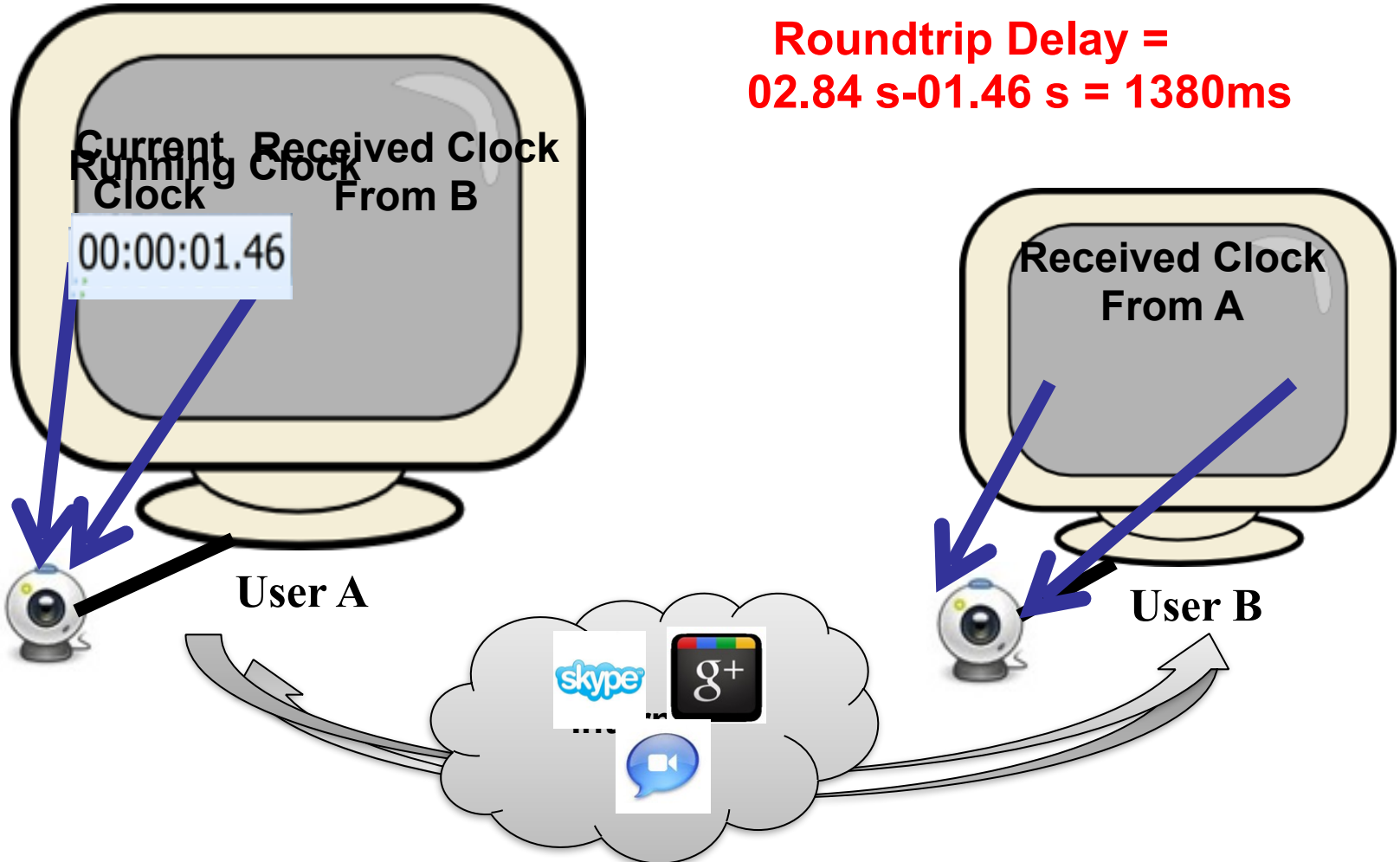
- One-way Video delay is around 200ms.
- Since network transmission delay<20ms, video processing delay accounts for about 180ms.

Table 7: One-way Delay Performance (ms)

Systems		Video	Voice
Google+		180	100
Skype Two-Party		156	110
Skype Multi-party	initiator to normal	230	130
	normal to normal	230	190
iChat Two-Party		220	220
iChat Multi-party	initiator to normal	220	220
	non-initi. to non-initi.	270	270

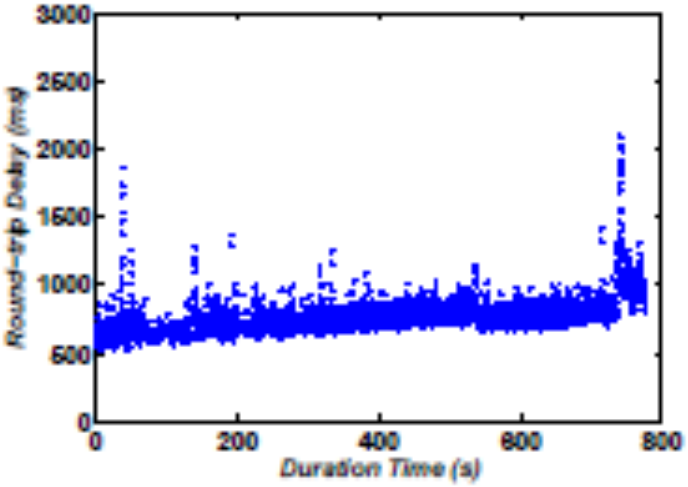
pDelay Performance over Internet

- Delay between two geographically distributed computers
- One-way video delay measurement method couldn't be used here.
- Round-trip Video Delay:

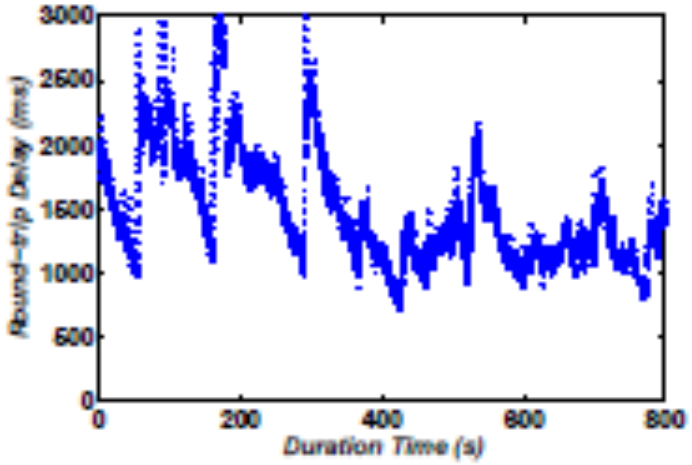


pDelay Performance over Internet

- Delay Experiments between Hong Kong and New York show that Skype multi-party's delay performance is much worse!
- A larger measured delay using our testbed could be due to:
  - ✓larger video packet delay -- propagation delay and queue delay
  - ✓ video packet loss → undecodeable frame -> long video delay



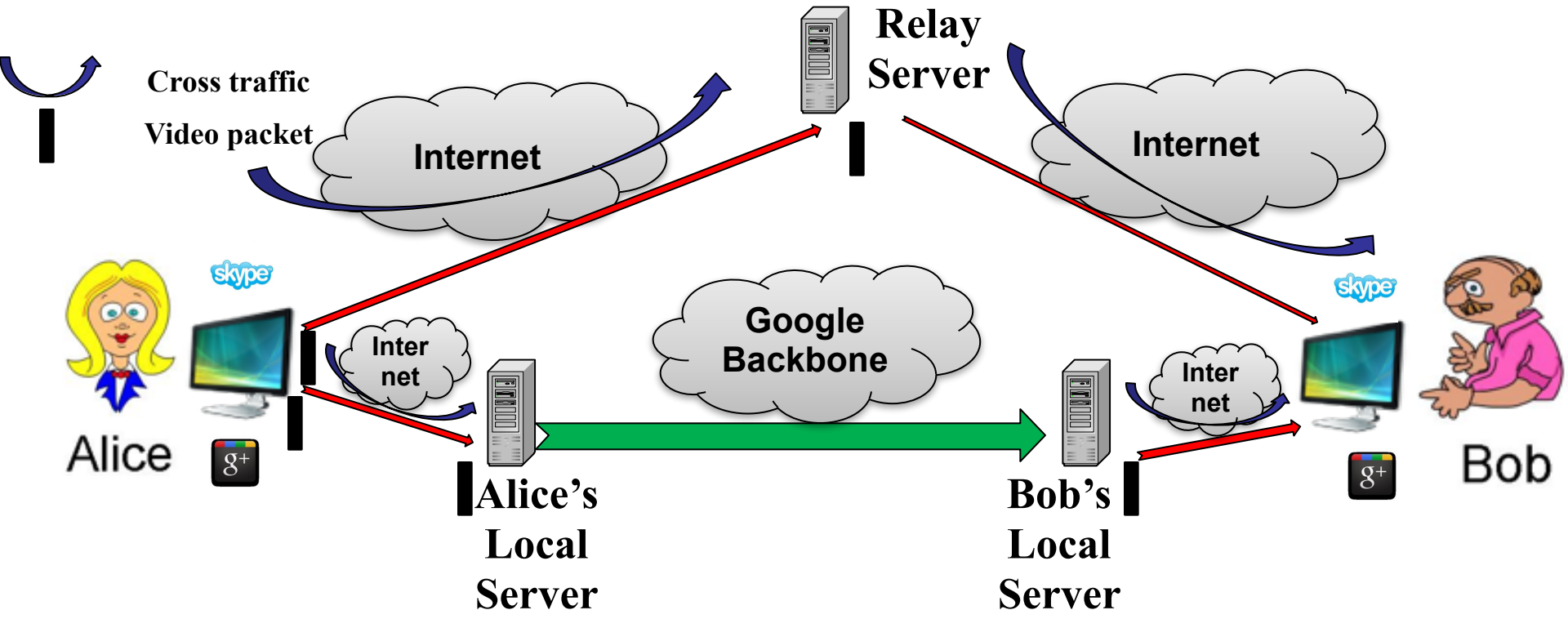
(a) Google+ Hangout



(b) Skype Multi-party

pDelay Performance in Internet(3)

- Architecture:
  - Skype: servers just do relay function, video flows need to compete with other unknown flows in Internet all the way.
  - Google+, transmission between their servers can guarantee good QoS.



pLoss Recovery Method

- Question: how these systems recover from packet losses under tight delay budget?
- Conventional Wisdom
  - Retransmission would incur long retransmission delays
  - Forward-Error-Correction (FEC): add redundancies to do protection

pLoss Recovery Method

- Skype uses FEC:
  - Profiling Skype Video Calls: Rate Control and Video Quality (INFOCOM 2012)
  - For upload link side, use very aggressive FEC approach. (FEC ratio > 0.45)
  - For download link side, FEC ratio would increase if loss ratio increases.
- Skype could endure random packet loss (< 10%).

Table 9: FEC Adaptation at Skype Sender Side

Video Rate (kbps)	Video Sender Side			Video Receiver 1		Video Receiver 2	
	Upload Loss Ratio	Upload Rate (kbps)	FEC Ratio ρ_s	Received Rate (kbps)	FEC Ratio ρ_r	Received Rate (kbps)	FEC Ratio ρ_r
513.48	0	974.00	0.47	542.62	0.05	542.56	0.05
474.26	0.02	1108.00	0.57	519.61	0.09	522.05	0.09
460.37	0.05	1019.50	0.55	487.84	0.06	488.19	0.06
225.05	0.08	496.57	0.55	241.80	0.07	241.32	0.07

Table 10: FEC Adaptation at Skype Relay Server Side

Video Rate (kbps)	Video Sender Side		Video Receiver 1			Video Receiver 2	
	Upload Rate (kbps)	FEC Ratio ρ_s	Download Loss Ratio	Relay Send Rate (kbps)	FEC Ratio ρ_r	Received Rate (kbps)	FEC Ratio ρ_r
513.48	974.00	0.47	0	542.62	0.05	542.56	0.05
478.52	1163.87	0.59	0.02	653.40	0.27	505.43	0.05
440.94	955.72	0.54	0.05	949.73	0.54	465.82	0.05
343.73	821.43	0.58	0.08	824.39	0.58	363.88	0.06

pLoss Recovery Method

- Google+ uses Selective Persistent Retransmission:
  - Selective – protect half of the lost packets. (likely lower layers).
  - Persistent – keep retransmitting a selected packet until it is received successfully (mostly 1-2 tries, up to 18 tries – observed batch retransmissions for a lost packet).
- Offer acceptable video quality under 40% random packet loss!

Table 11: Retransmission in the Upload Link of Google+ Hangout

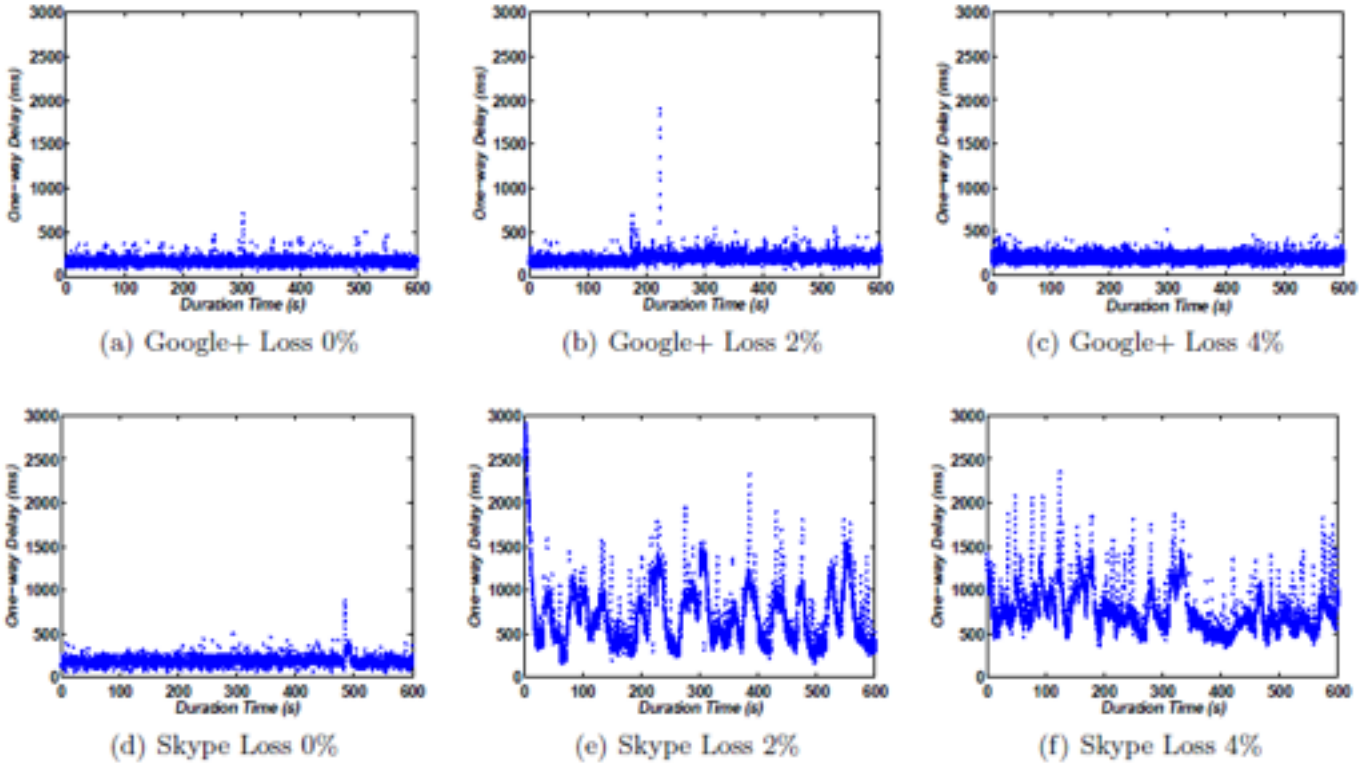
Loss	FPS	Total Rate (kbps)	Retrans Rate (kbps)	Recovery Ratio	Persistent Ratio
0	29.97	826.0	0.9	-	-
0.05	29.98	836.9	24.2	0.46	1.00
0.10	29.97	885.7	52.1	0.46	1.00
0.20	29.98	857.2	101.4	0.45	1.00

Table 12: Retransmission in the Download Link of Google Plus Hangout

Loss	FPS	Total Rate (kbps)	Retrans Rate (kbps)	Recovery Ratio	Persistent Ratio
0	27.91	810.8	4.4	-	-
0.05	27.07	827.3	35.2	0.51	1.00
0.10	20.158	744.3	67.4	0.60	1.00
0.20	19.231	677.7	116.4	0.64	1.00

pRobustness of Conferencing Quality

- Bursty Loss: 4-5 packet will be dropped in batch.
  - Skype's video delay's variance is too large (lots of frames couldn't be displayed). FEC approach is hard to recover lost packets when bursty loss happens.
  - Google+'s local retransmission is more robust.



(a) Google+ Loss 0% (b) Google+ Loss 2% (c) Google+ Loss 4%

(d) Skype Loss 0% (e) Skype Loss 2% (f) Skype Loss 4%

Figure 9: One-way Delay For Google+ and Skype under Different Bursty Loss Probabilities

pConclusion:

- Bandwidth-rich server infrastructure can be deployed to significantly improve user conferencing experiences.

- In extremely tight design space, video generation, protection, adaptation and distribution have to be jointly considered.

- Video and voice processing delay account for a significant portion of end-to-end delay.

- When relay servers are well-provisioned and selected, per-hop retransmission is more preferable than FEC .

- With layered video coding, prioritized selective retransmissions can further enhance the robustness against various network impairments.

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

NYU

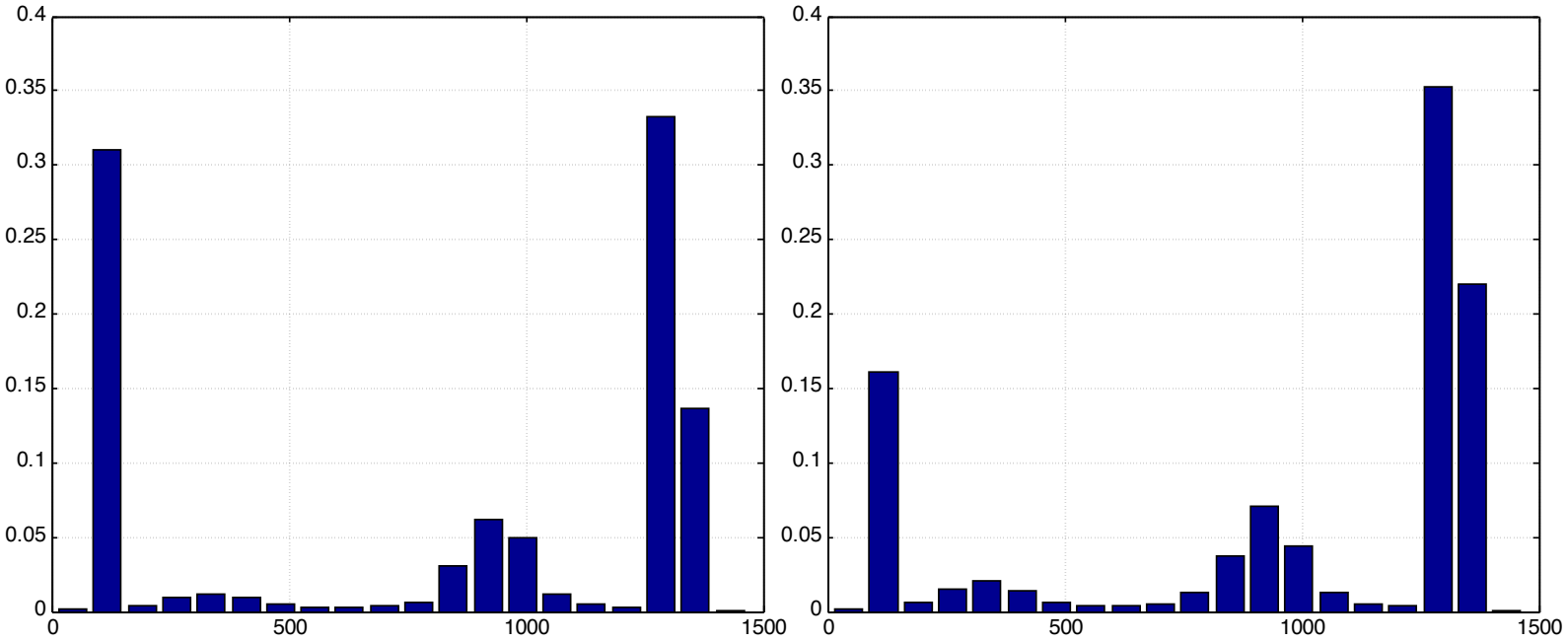
Thanks!

☺

Q && A ?

NYU WIRELESS

Packet Size Distribution(Sender)



- More large packets with increased loss ratio
  - FEC mechanism for error protection

pLoss Recovery Method

- FEC challenges in video conferencing:
  - how much redundancies? hard to accurately predict loss rate
  - long FEC block → long FEC encoding and decoding delay
  - short FEC block → high redundancies

➢Video Generation and Adaptation For Heterogeneity

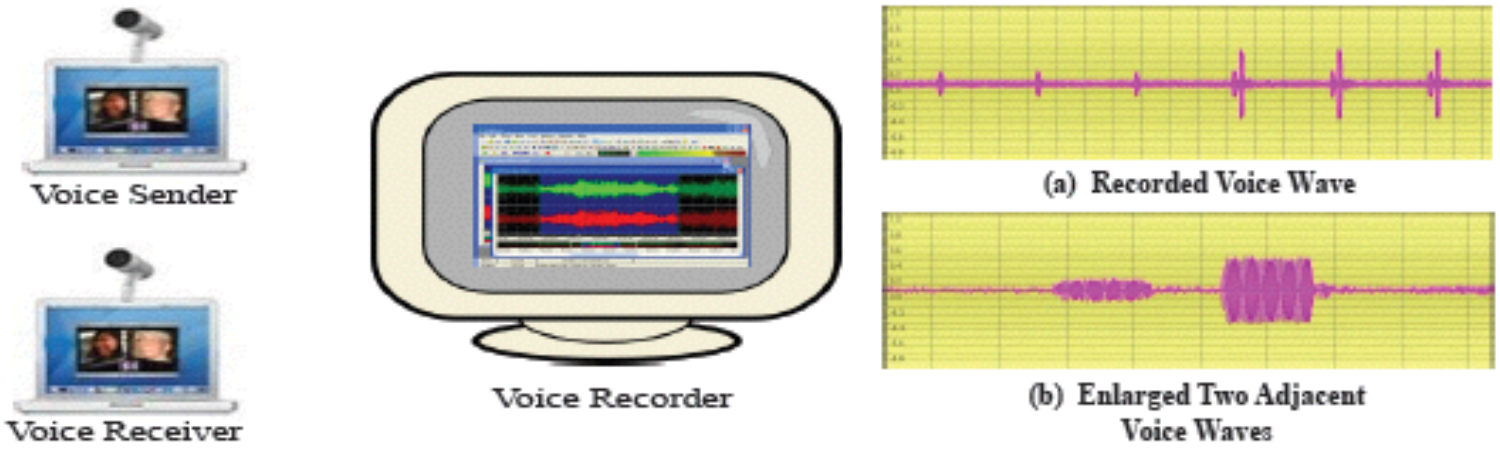
pGoogle+ Payload Analysis: (RTP header)

- Use “M”, “Timestamp” and “Length” fields to find the same packet in Sender, Receiver 1 and Receiver 2 sides
- Server decides to send some selected frames to Receiver 1, selected packets in some frames to Receiver 2 in Table 6.

Table 6: Packet Payloads in Google+

M	Timestamp	Length (bytes)	Sequence Number		
			Sender	Receiver 1	Receiver 2
0	2063696701	1269	61603	44445	52498
0	2063696701	1113	61604	44446	52499
0	2063696701	1278	61605	44447	
0	2063696701	1234	61606	44448	
0	2063696701	1283	61607	44449	
0	2063696701	1277	61608	44450	
0	2063696701	1077	61609	44451	
1	2063696701	989	61610	44452	
0	2063699269	621	61611		
1	2063699269	560	61612		
0	2063703362	1086	61613		
0	2063703362	485	61614		
0	2063703362	1167	61615		
1	2063703362	1048	61616		
0	2063706604	543	61617		
1	2063706604	914	61618		
0	2063709620	1276	61619	44453	52500
0	2063709620	1067	61620	44454	52501
0	2063709620	1272	61621	44455	
0	2063709620	1267	61622	44456	
0	2063709620	1279	61623	44457	
0	2063709620	1276	61624	44458	
1	2063709620	736	61625	44459	

pOne-way Voice Delay: Use a tool to inject repeated “Tick” sound (period time > 2s) into voice sender, take time difference between signal into voice sender and signal out of voice receiver as the one-way voice delay.

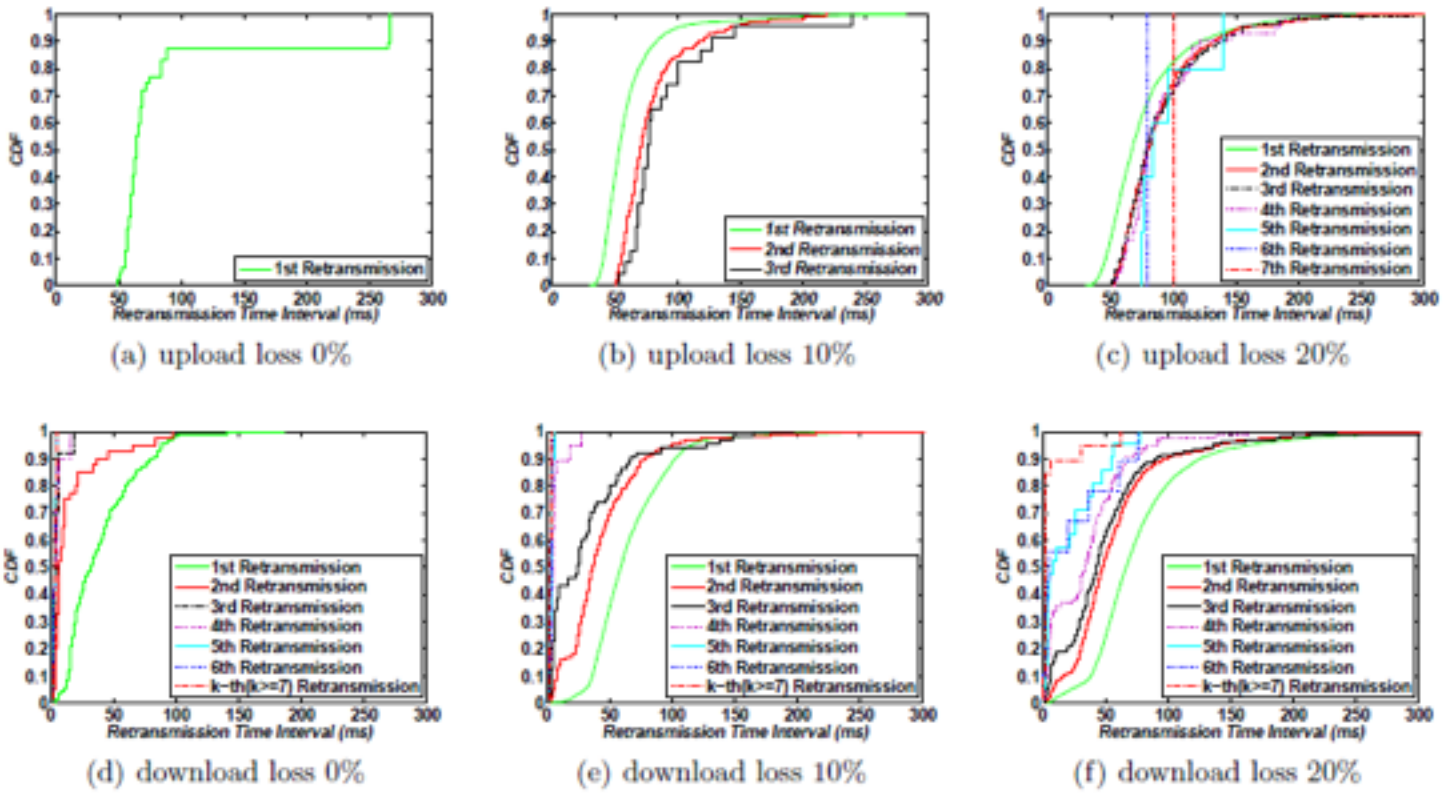


(a) One-way Voice Delay (b) Voice Waves Detail

➢Loss Recovery(1)

pGoogle+ uses Selective Persistent Retransmission:

- ✓60% of the first retransmission happens within 70 ms (RTT = 14ms) after the first transmission.
- ✓Uplink side -- Conservative way, always wait for about 70 ms before retransmission.
- ✓Downlink side – Aggressive way, use batch retransmissions before lost confirmation.



(a) upload loss 0% (b) upload loss 10% (c) upload loss 20%

(d) download loss 0% (e) download loss 10% (f) download loss 20%

Figure 7: Retransmission Time Interval under Loss Variation for Google Plus Hangout

➢Loss Recovery(2)

pGoogle+ uses Selective Persistent Retransmission (2):

✓Persistent -- retransmit selected packets until received successfully (most of the time 1-2 tries, highest up to 18 times!).

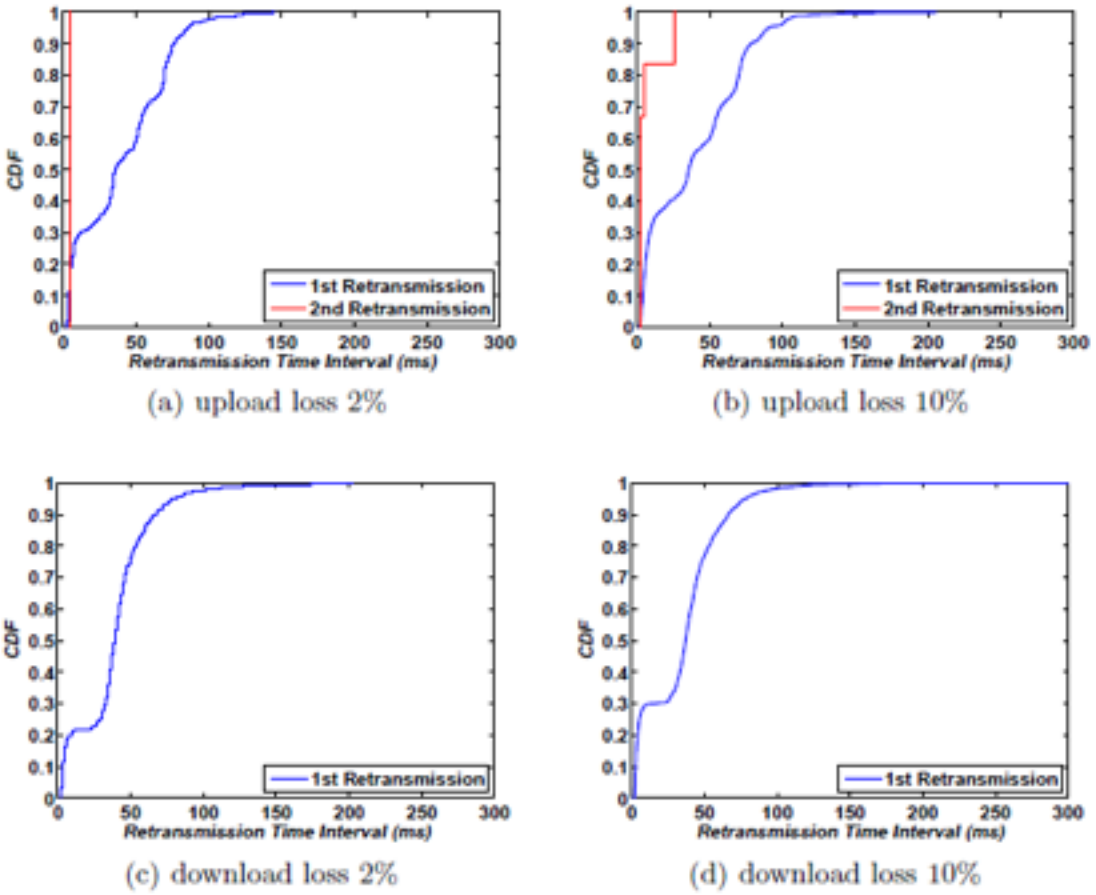
Table 13: Retransmission Probability in Google+

Case	1st	2nd	3rd	4th	5th	6th	k-th ($k \geq 7$)
Downlink loss 0	0.8058	0.1311	0.0146	0.0097	0.0049	0.0146	0.0194
Downlink loss 0.05	0.8845	0.0958	0.0101	0.0032	0.0032	0.0005	0.0027
Downlink loss 0.10	0.8677	0.1125	0.0140	0.0027	0.0003	0.0003	0.0024
Downlink loss 0.20	0.7691	0.1793	0.0376	0.0100	0.0023	0.0008	0.0010
Uplink loss 0	1.00						
Uplink loss 0.05	0.9492	0.0477	0.0023	0.0008			
Uplink loss 0.10	0.8963	0.0947	0.0090				
Uplink loss 0.20	0.7996	0.1620	0.0293	0.0080	0.0009	0	0.0002

➢Loss Recovery(3)

piChat's Retransmission Strategy:

- ✓Doesn't employ persistent retransmission, just tries to do retransmission once or twice.
- ✓iChat waits for 30+ ms (RTT = 2 - 4ms) for the first retransmission.



(a) upload loss 2%

(b) upload loss 10%

(c) download loss 2%

(d) download loss 10%

Figure 8: Retransmission Time Interval under Loss Variation for iChat

➢Loss Recovery Method

piChat uses Simple Retransmission:

- ✓When loss happens, just tries to do retransmission once or twice, then give up.
- ✓A lost video packet may prevent decoding of multiple subsequent video frames. (If a packet is lost in I frame, many P frames or B frames depending on that I frame couldn't be decoded at all !)
- ✓iChat's video quality under losses is much worse than Google+ and Skype (show later)

➢Robustness of Conferencing Quality (1)

pCase 1 – Bursty Loss (1):

✓Introduce bursty losses to download link. For each bursty loss, 4-5 packet will be dropped in batch.

✓Measure One-way video delay performances of these systems under different bursty loss ratios.

✓Digital Clock Picture Recognition Probability(using OCR) shows that:

- iChat's video quality is too bad

Table 19: Digital Clock Recognition Probability

Case	iChat	Google+	Skype
No Loss	0.90	0.86	0.88
2% Bursty loss	0.46	0.90	0.90
4% Bursty loss	0.10	0.76	0.92

➢Robustness of Conferencing Quality (3)

pCase 2 – Varying RTT:

✓Retransmission approach is sensitive to RTT.

✓We vary RTT to compare one-way video delay performances of Skype and Google+, Two curves of Skype and the curve of Google + with no loss are almost the same. When slight loss happens and RTT is large, Google +'s delay performance becomes worse than Skype.

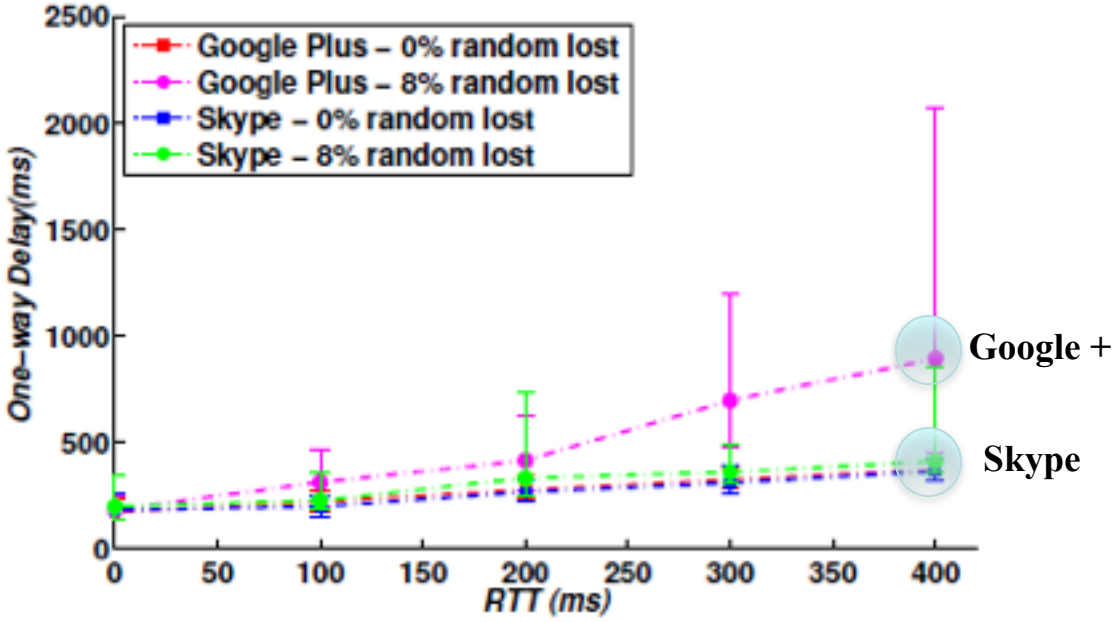


Figure 10: Google+ and Skype Delay Performances under RTT variation