Quantifying QoS Requirements of Network Services:

A *Cheat-Proof* *Framework*

Kuan-Ta Chen Academia Sinica
Chen-Chi Wu National Taiwan University
Yu-Chun Chang National Taiwan University
Chin-Laung Lei National Taiwan University

Presented by Cheng-Chun Tu (Stony Brook University)

The Long-Lasting Question

For a real-time interactive network service, what is the *minimum level of network QoS* required to provide *satisfactory user experience*?

More Concretely…

What is the

Minimum network bandwidth
Maximum packet loss rate
Maximum network delay

for a smooth

Skype
MSN Messenger
AIM
Google Talk
Lineage
World of Warcraft
Unreal Tournament

user experience ?

Motivation

Understanding QoS requirements can enable …

- ***Network planning***
  - E.g., how to place game servers if we know the maximum acceptable RTT of certain online game.

- ***Resource arbitration***
  - E.g., guarantee network bandwidth for conferencing calls at home gateway

Our Contributions

- A *general, cheat-proof* framework for quantifying the minimum QoS needs

- *cross-application* comparative analysis of applications' minimum network QoS need
  - E.g., Skype vs. Google Talk

- *cross-service* comparative analysis of network services' resource demands
  - E.g., VoIP vs. online games

Properties of Our Framework

- *Simple experiment procedure*
  - even inexperienced participants can make consistent judgments easily

- Cheating detection → enabling *crowdsourcing*

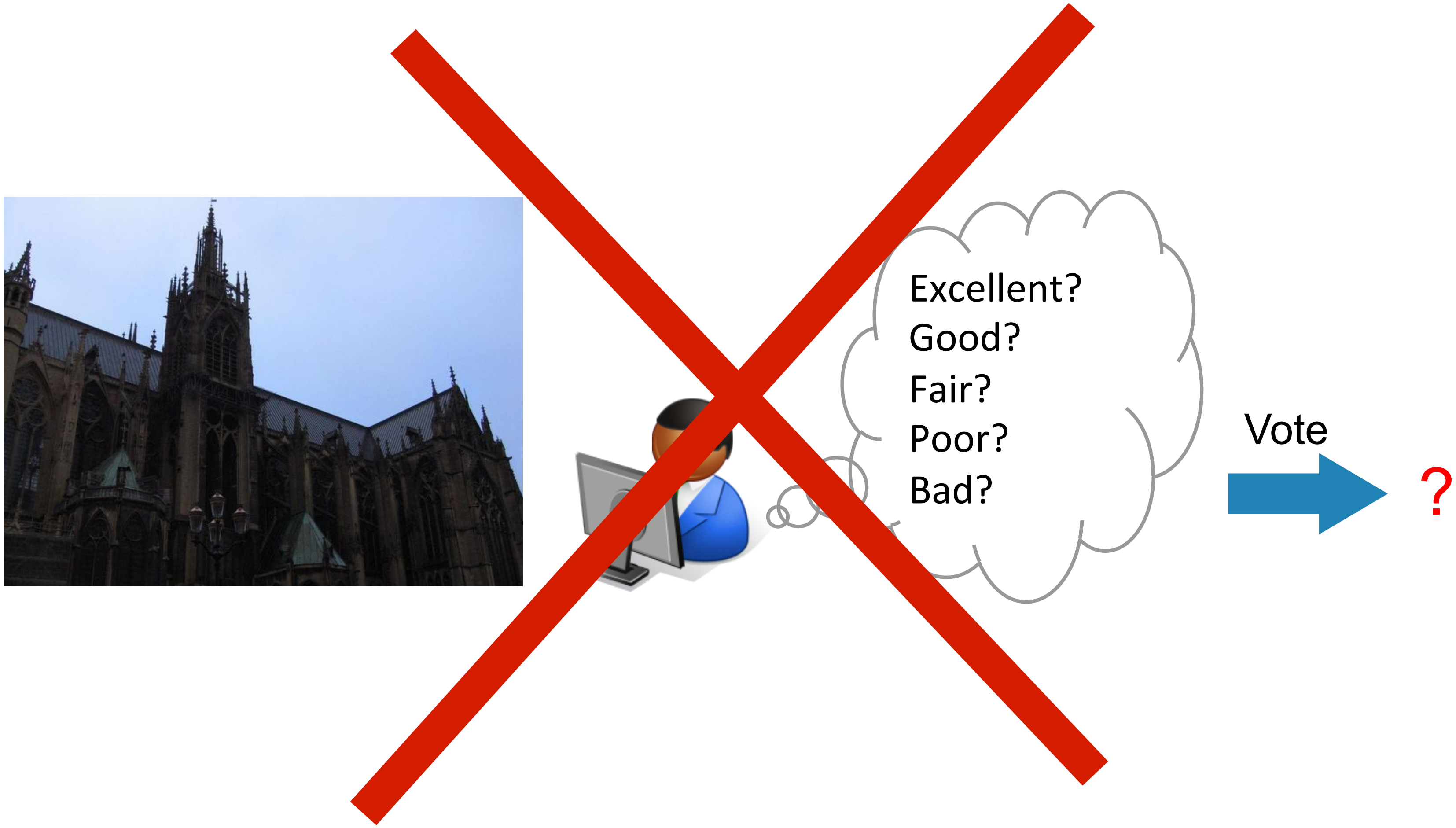
- *No artificial thresholds* are used/defined

Our Ambition

A simple, cost-effective and cheat-proofing way

to measure QoS requirements of a network service

Intuitive Solution: MOS Rating



Excellent?
Good?
Fair?
Poor?
Bad?

Why Not MOS: Reasons

- *Slow in scoring* (think/interpretation time)
- *Not cheat-proof*
- *No justifiable threshold* representing "barely acceptable" level

Talk Progress

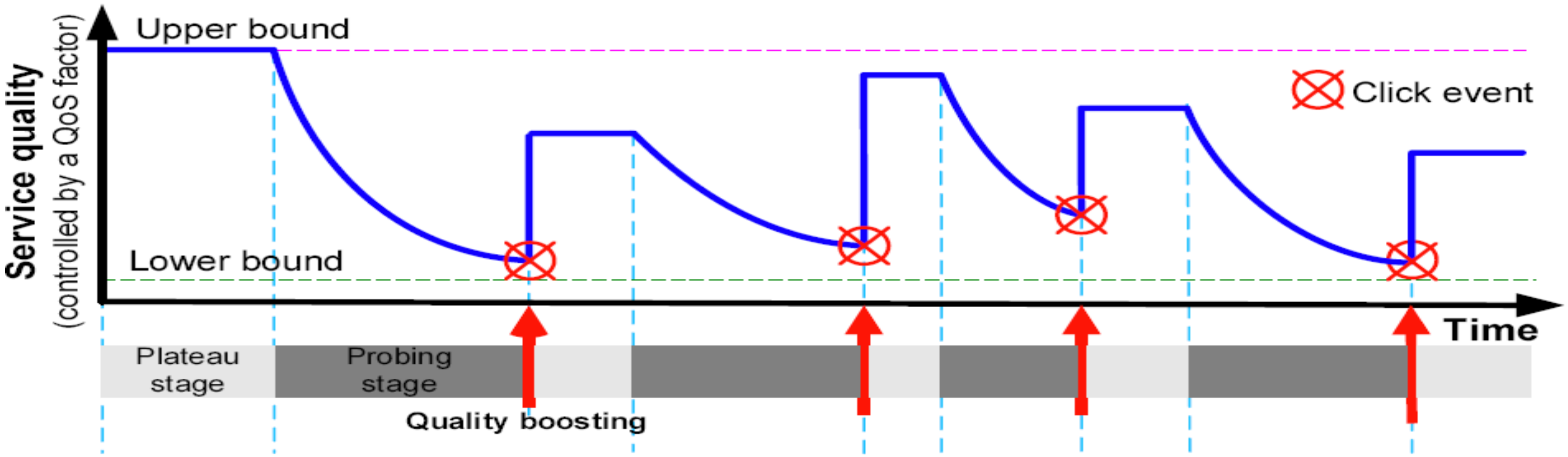
- Overview
- Methodology
  - Experiment Design
  - Cheat Proof Mechanism
- Pilot Study
  - Setup
  - Consistency Check
  - Intra-Service Comparison
  - Inter-Service Comparison
- Conclusion

Method Overview

- "Method of Limits" approach from Psychophysics
- *Repeat measurements:*
  Gradually decrease application quality until the quality becomes not acceptable
- We record the network QoS that correspond to minimum acceptable application quality as *"intolerance threshold"*

Intolerance Threshold Measurement

- ***Plateau stage***
  - Remind users the “normal” service quality
- ***Probing***
  - Explore users’ intolerance thresholds
- ***Quality boosting***
  - Enter the next round of measurement

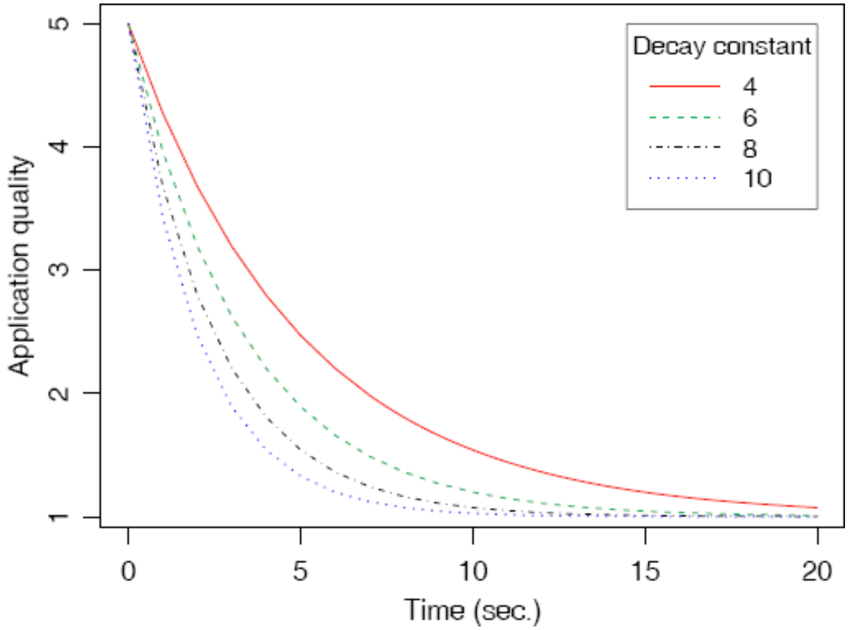


Probing Stage

- Explore users' intolerance thresholds in a graceful way
- Basically following the exponential decay function

$$N(t) = N_0 e^{-\lambda t}$$

- Conceptually like "slow start" + "congestion avoidance"



Cheating Detection

- Detect participants who do not pay attention to experiments even in laboratory

- Also, enable ***crowdsourcing***
  - Crowdsourcing = Crowd + Outsourcing
  - To reduce experiment cost

Not every Internet user is trustworthy!

- Users may give erroneous feedback perfunctorily, carelessly, or dishonestly
- Dishonest users have more incentives to perform tasks

Randomness in Measurement Procedures

- To prevent users from ***guessing*** the current service quality based on time, run-time parameters are ***randomly*** decided

- Plateau stage
  - Duration: 2 – 6 seconds
- Probing stage
  - Duration: 15 – 25 seconds
- Quality boosting
  - Increased to a random service quality

Cheat Proof Mechanism

- A measured intolerance threshold ➔ an intolerance threshold sample (ITS)
- *Basic idea:* ***If a user's ITS samples are statistically self-consistent over time***

- Assume a user made *n* ITS samples $\mathbf{v} = (v_1, v_2, \ldots, v_n)$
- Repeat *m* times: randomly divide $\mathbf{v}$ into $\mathbf{v_a}$ and $\mathbf{v_b}$ test if $\mathbf{v_a} \sim \mathbf{v_b}$ using *Wilcoxon rank-sum hypothesis test*
- The p-value of hypothesis tests is adjusted using the *Bonferroni method* (significance level = α/m)
- See if all the p-values are higher than α/m

Talk Progress

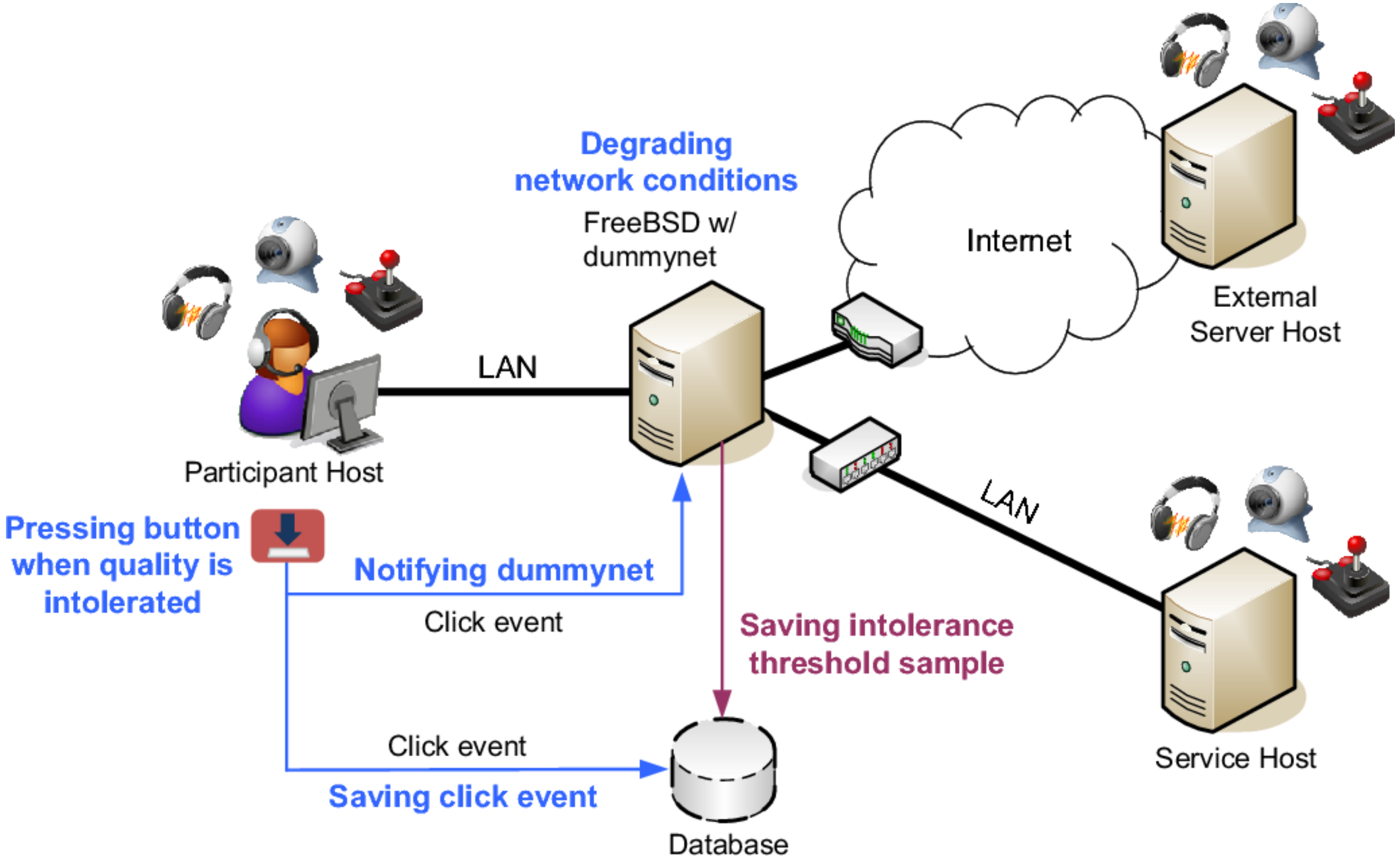
- Overview
- Methodology
  - Experiment Design
  - Cheat Proof Mechanism
- Pilot Study
  - Setup
  - Consistency Checks
  - Intra-Service Comparison
  - Inter-Service Comparison
- Conclusion

Pilot Study

- To verify the efficiency and effectiveness of the our framework

- QoS factors
  - Network bandwidth, packet loss rate, network delay
- Applications chosen in the study
  - VoIP: AIM, MSN Messenger, Skype, Google Talk
  - Conferencing: AIM, MSN Messenger, Skype
  - Games
    - FPS: Unreal Tournament
    - RPG: Lineage, World of Warcraft

Experiment Setup



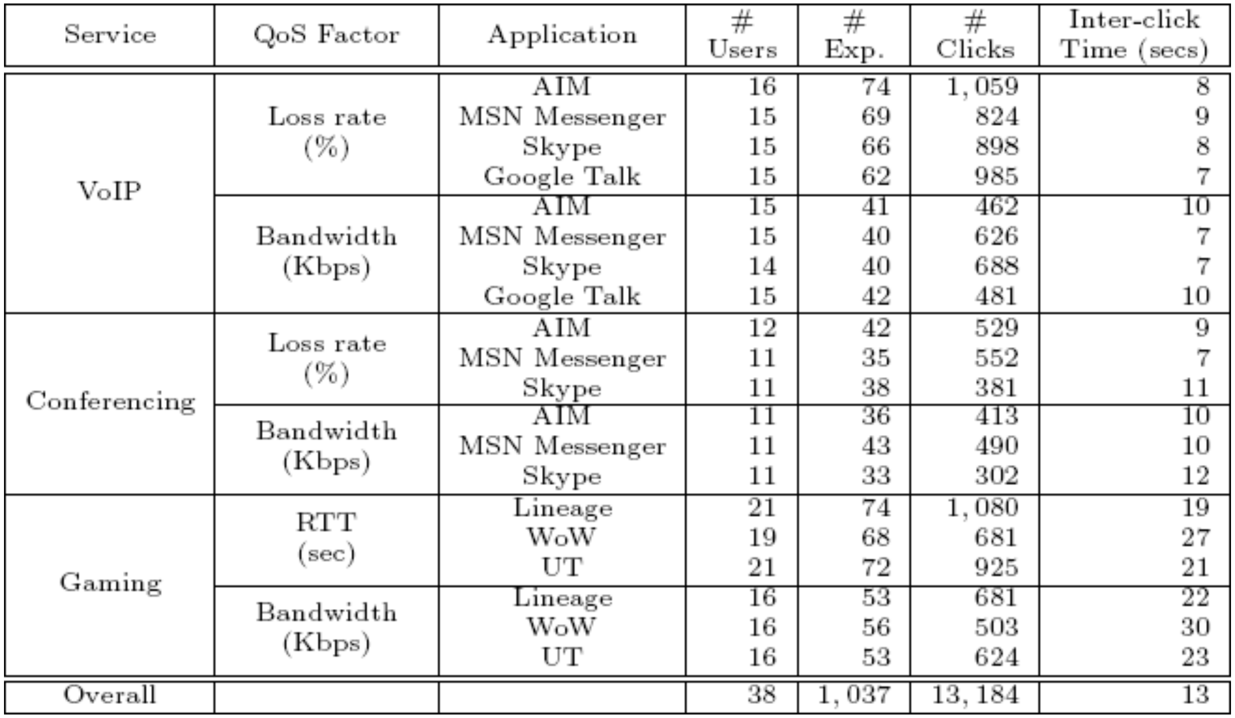
Participants Instruction

The only guideline given was

"Click the SPACE key whenever you find the service quality intolerable."

Summary of Experiment Results

- **38** part-time employees
- **20** service-application-QoS factor combinations
- **1,037** experiment (**47.6** hours)
- **13,184** click actions



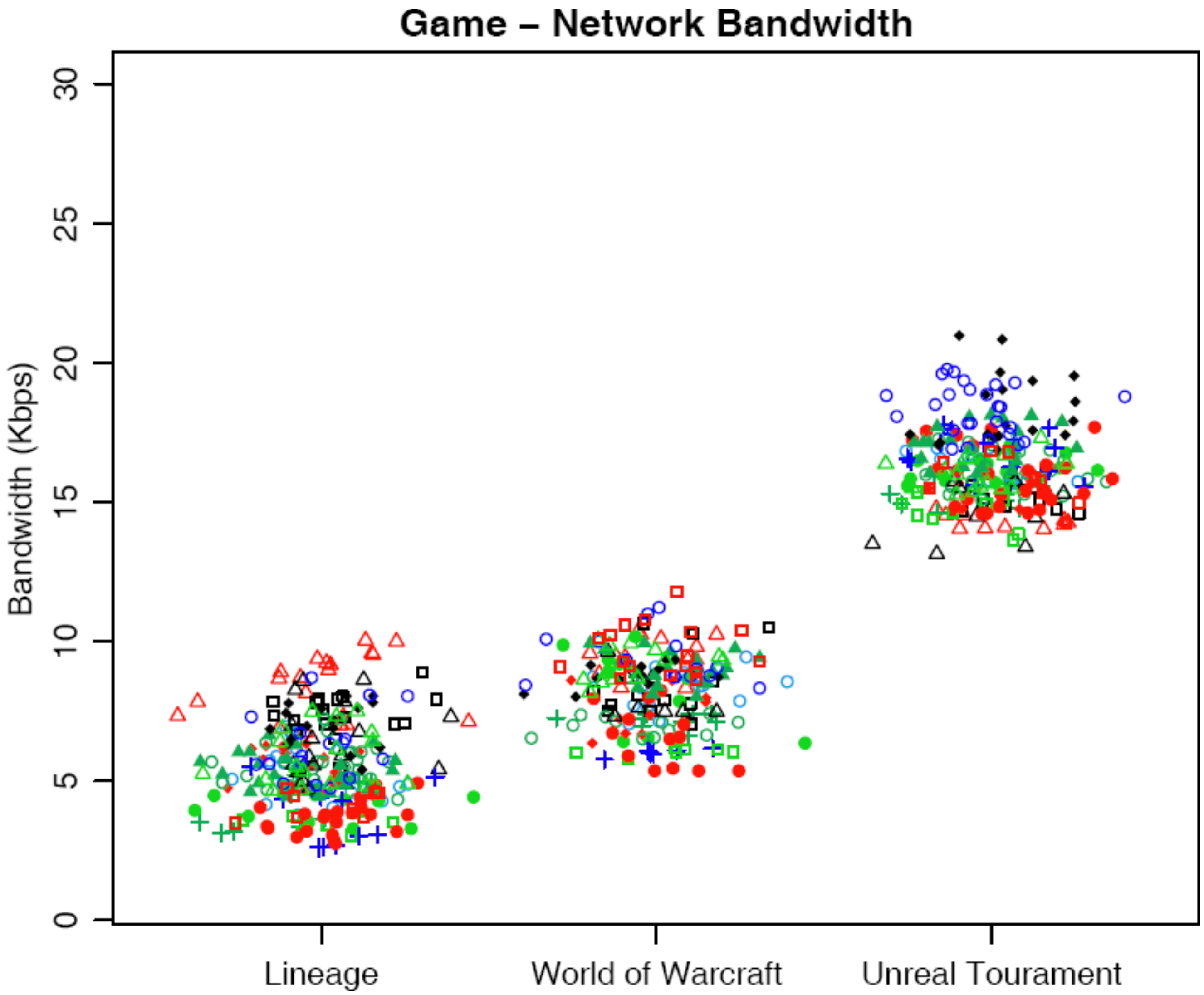
Service	QoS Factor	Application	# Users	# Exp.	# Clicks	Inter-click Time (secs)
VoIP	Loss rate (%)	AIM	16	74	1,059	8
		MSN Messenger	15	69	824	9
		Skype	15	66	898	8
		Google Talk	15	62	985	7
	Bandwidth (Kbps)	AIM	15	41	462	10
		MSN Messenger	15	40	626	7
		Skype	14	40	688	7
		Google Talk	15	42	481	10
Conferencing	Loss rate (%)	AIM	12	42	529	9
		MSN Messenger	11	35	552	7
		Skype	11	38	381	11
	Bandwidth (Kbps)	AIM	11	36	413	10
		MSN Messenger	11	43	490	10
		Skype	11	33	302	12
Gaming	RTT (sec)	Lineage	21	74	1,080	19
		WoW	19	68	681	27
		UT	21	72	925	21
	Bandwidth (Kbps)	Lineage	16	53	681	22
		WoW	16	56	503	30
		UT	16	53	624	23
Overall			38	1,037	13,184	13

Consistency Checks

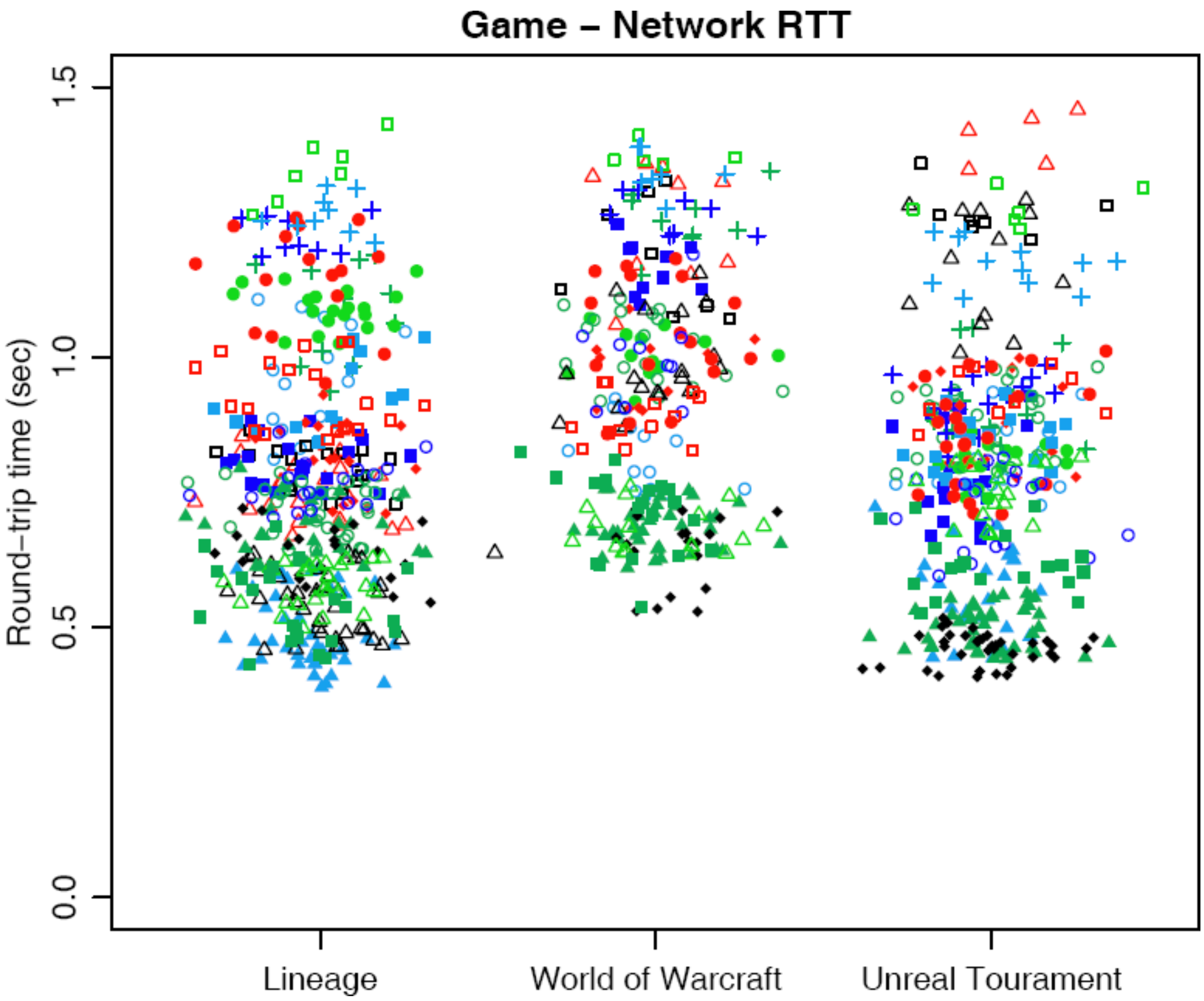
- Consistency of individual participants
  - 97% passed the cheat-proofing test with signif. level 0.05

- Consistency of overall Inputs
  - Generally consistent
  - ITS for some application-factor pairs are more variable than others
    ➔ "service quality may not be dentical even if the network conditions are exactly the same" (due to different workload)

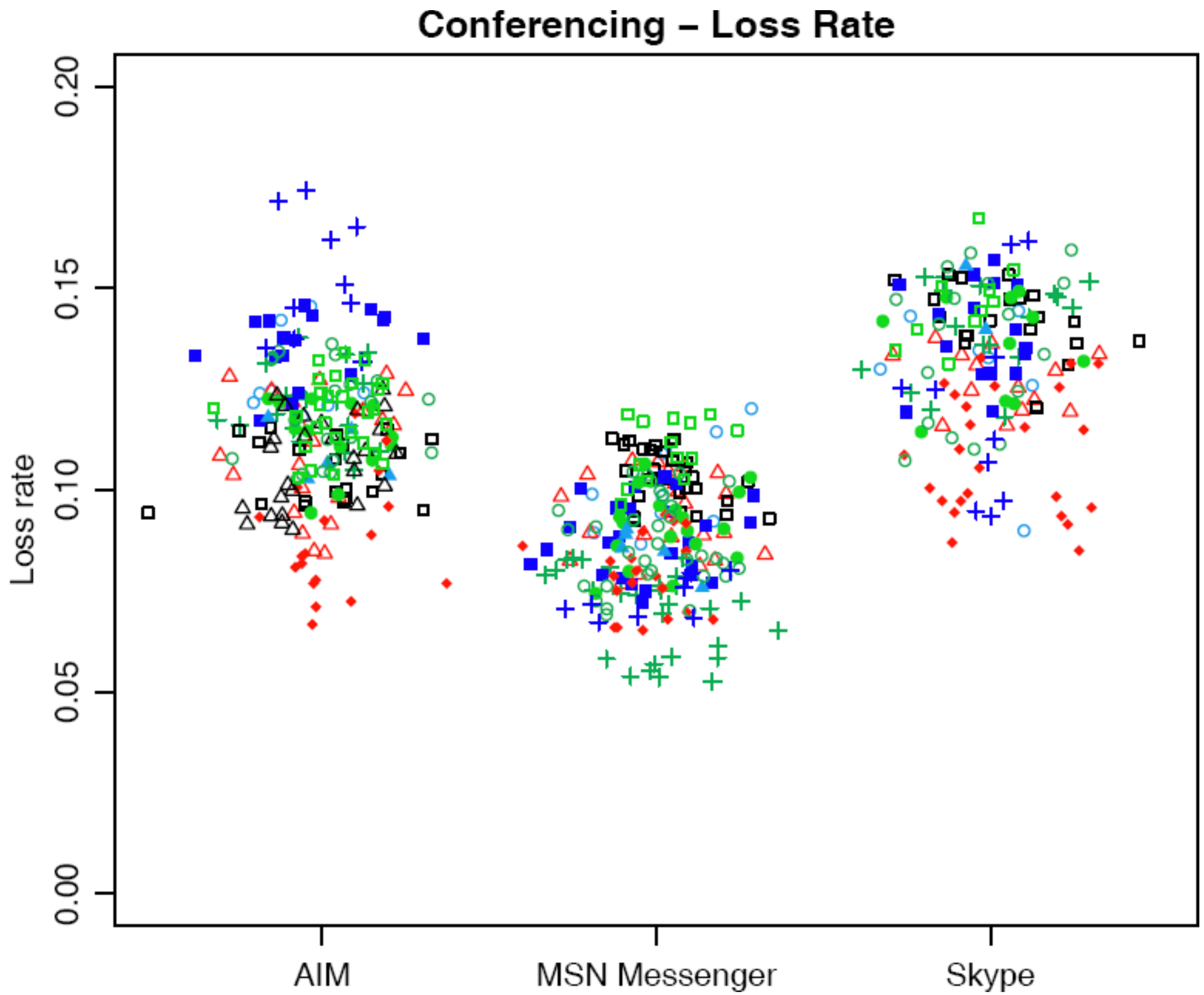
ITS Consistency Check (1)



ITS Consistency Check (2)

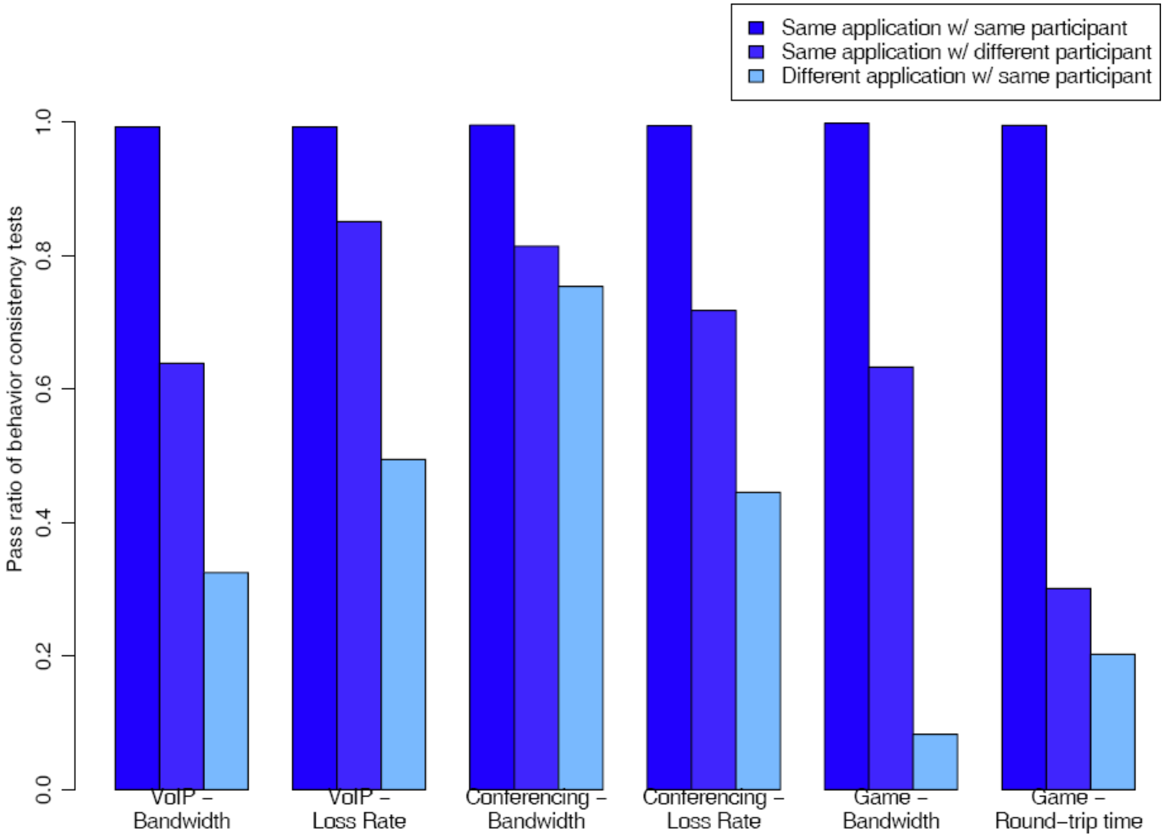


ITS Consistency Check (3)

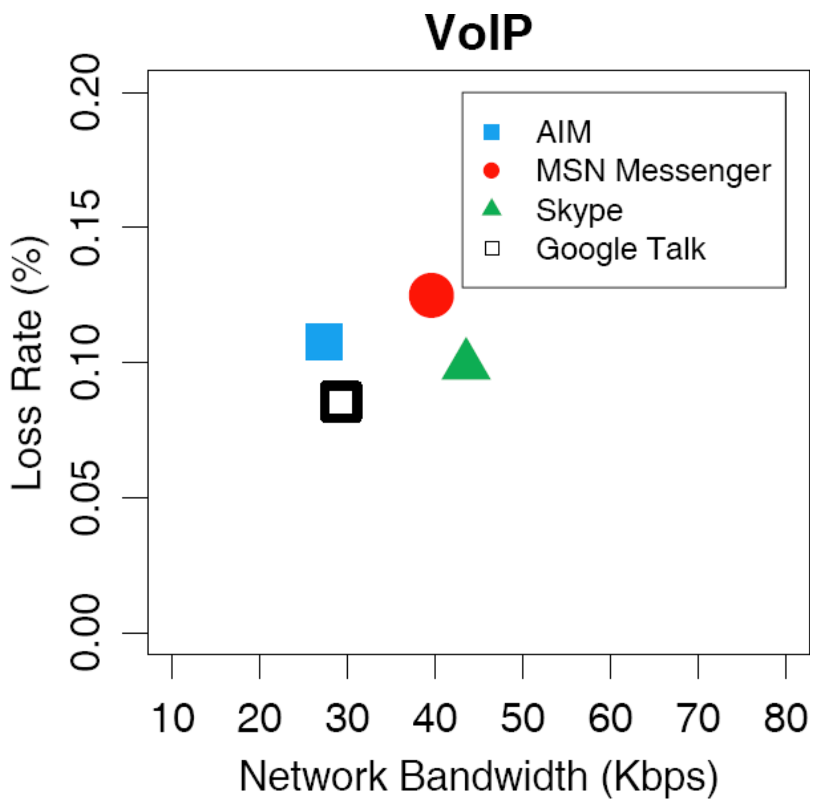


Consistency Check across Participants

- Different participants agree ***more*** on the threshold of an application than the same participant agrees on the thresholds of different applications

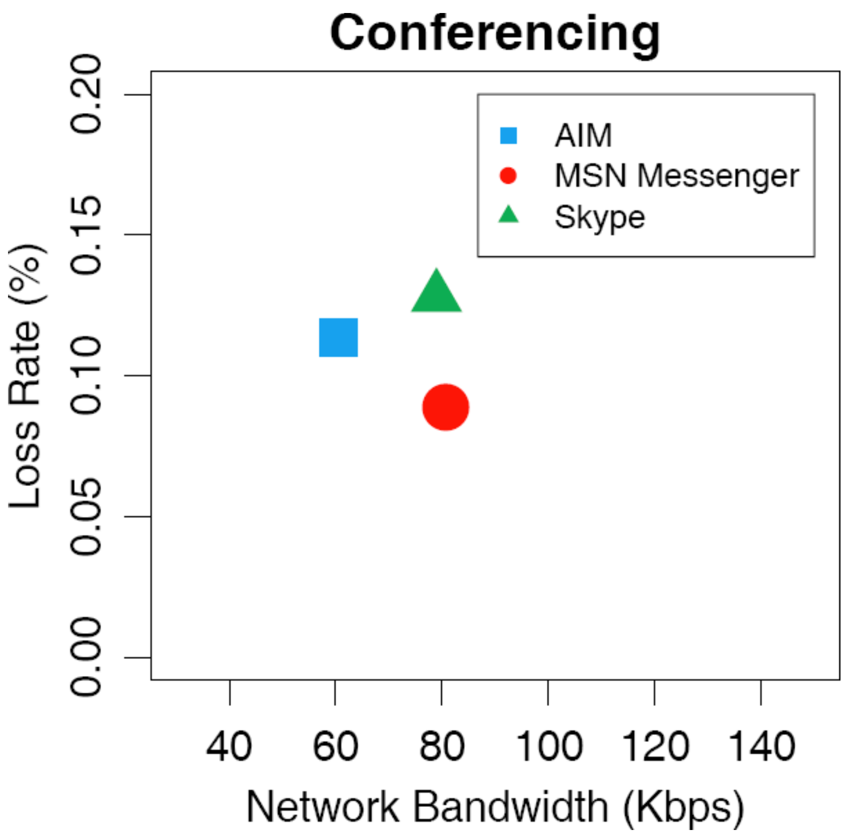


Intra-Service Comparative Assessment (VoIP)



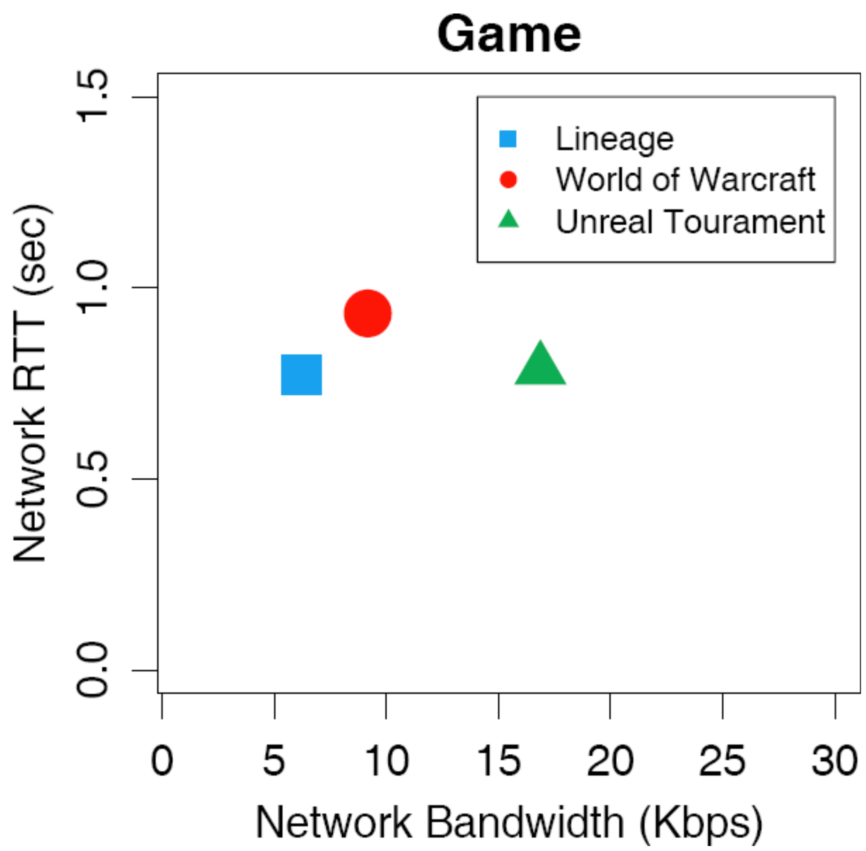
- Skype is most demanding in bandwidth (in contrast to AIM and Google talk)
- Google talk is least robust to packet loss (in contrast to Skype)
- ***Can see easily the strength and weakness of each application***

Intra-Service Comparative Assessment (Conferencing)



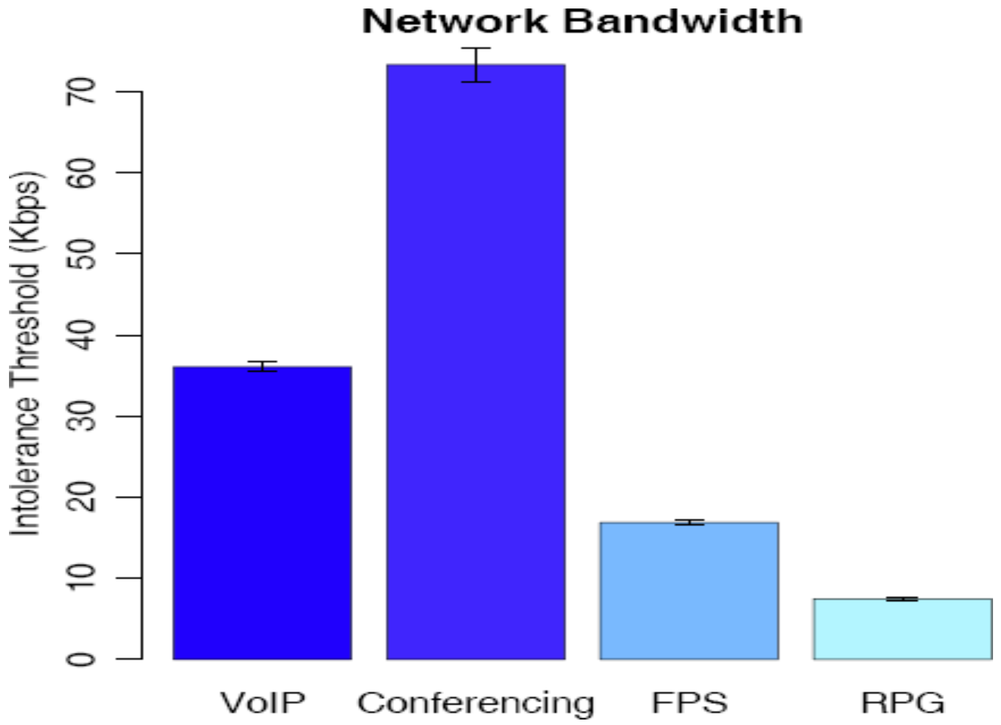
- Skype and MSN Messenger are more demanding in bandwidth
- Skype is most resilient to packet loss, but MSN Messenger is not
- ***None of the applications excel in all aspects***

Intra-Service Comparative Assessment (Games)



- UT, the only FPS game, is most demanding in bandwidth
- World of Warcraft is more bandwidth demanding, and more resilient to packet loss than Lineage

Inter-Service Comparison



- Coincidentally, the relative bandwidth needs of conferencing, VoIP, FPS, and RPG is approx. 8:4:2:1 (70, 35, 17, 8 Kbps)

Talk Progress

- Overview
- Methodology
  - Experiment Design
  - Cheat Proof Mechanism
- Pilot Study
  - Setup
  - Consistency Check
  - Intra-Service Comparison
  - Inter-Service Comparison
- Conclusion

Conclusion

- A ***general, cheat-proof*** framework for quantifying QoS requirements of network services
  - ***Simple***
  - ***Even untrained participants can produce consistent inputs***
  - ***Crowd-sourcing possible!***

- We hope the framework will be helpful for
  - Evaluation of competing applications
  - Application recommendation
  - Network planning
  - Resource arbitration
  - …

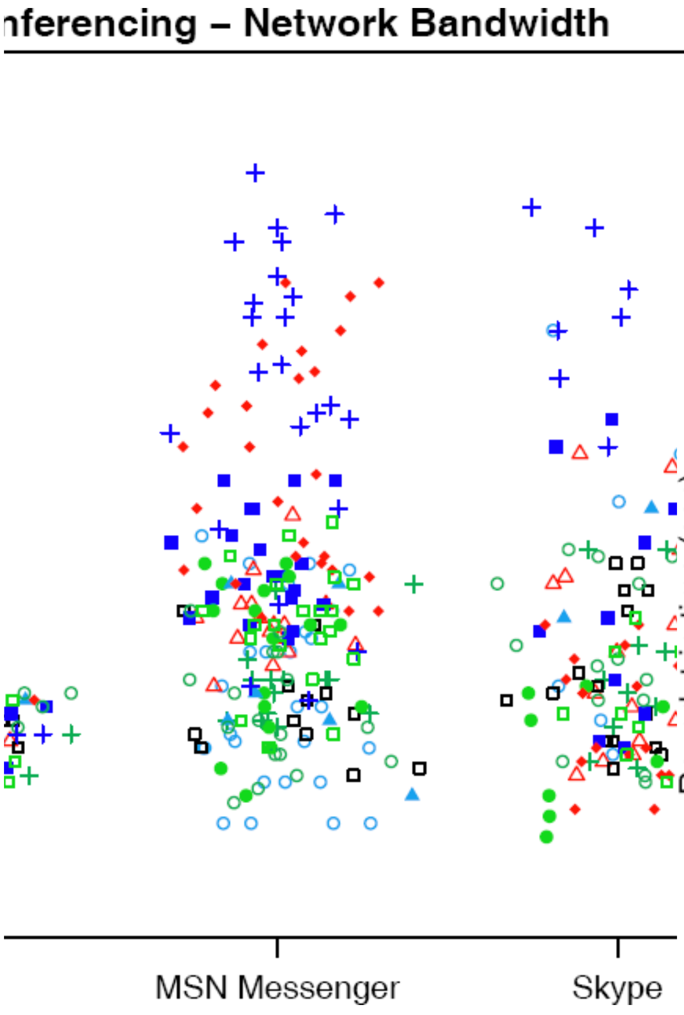
Thank You!

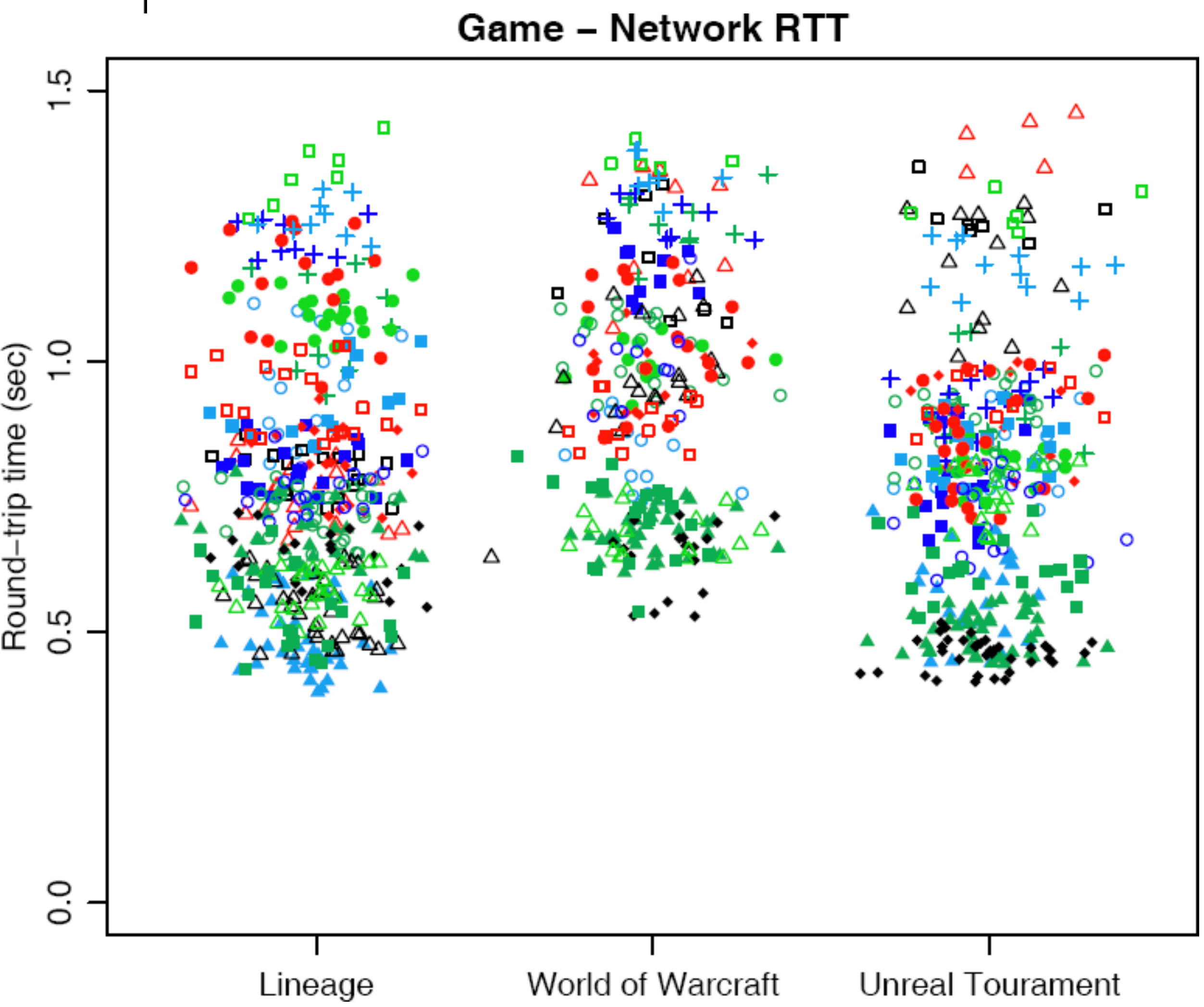
Kuan-Ta Chen	Academia Sinica
Chen-Chi Wu	National Taiwan University
Yu-Chun Chang	National Taiwan University
Chin-Laung Lei	National Taiwan University

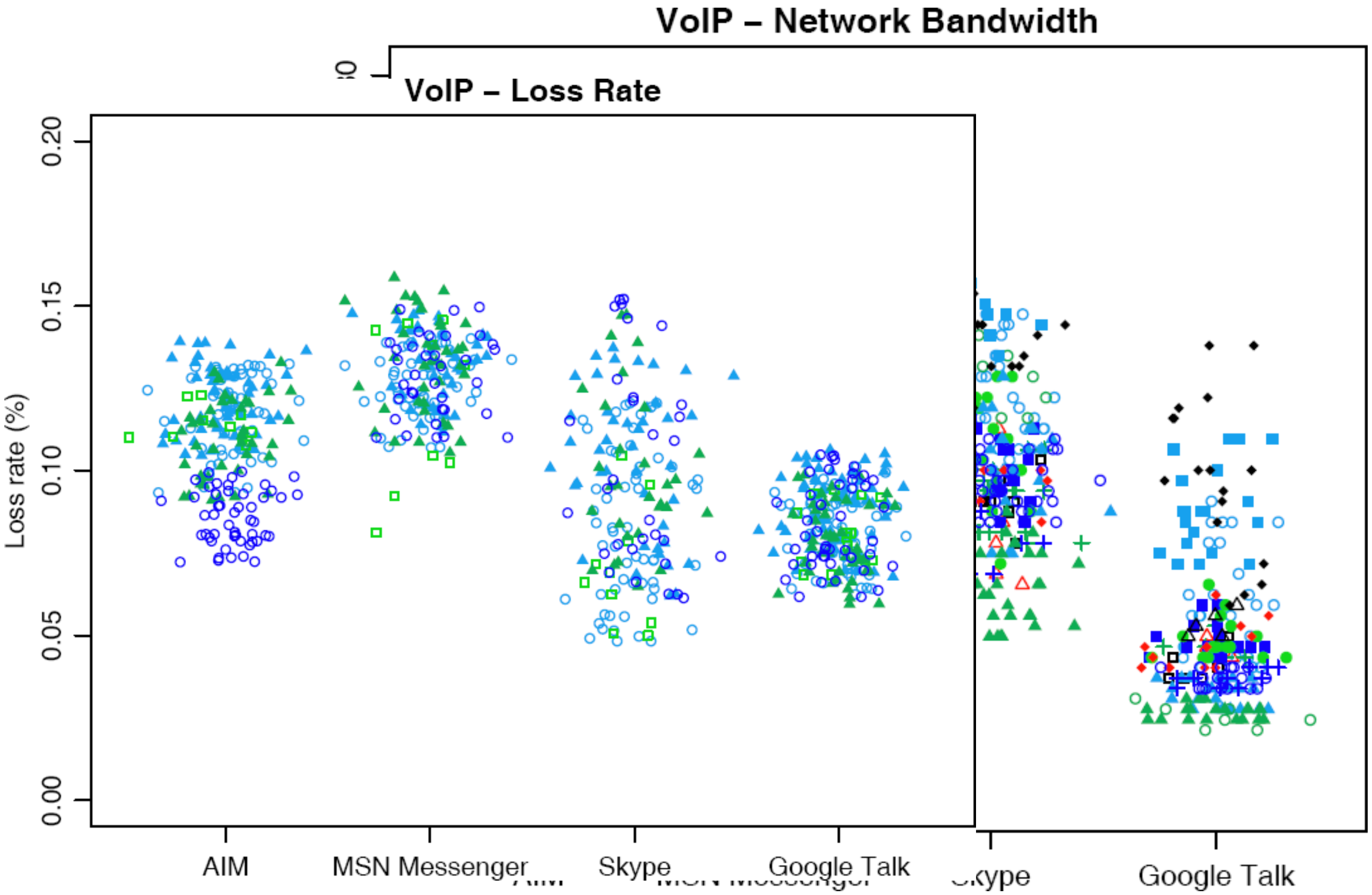
Presented by Cheng-Chu Tu (Stony Brook University)

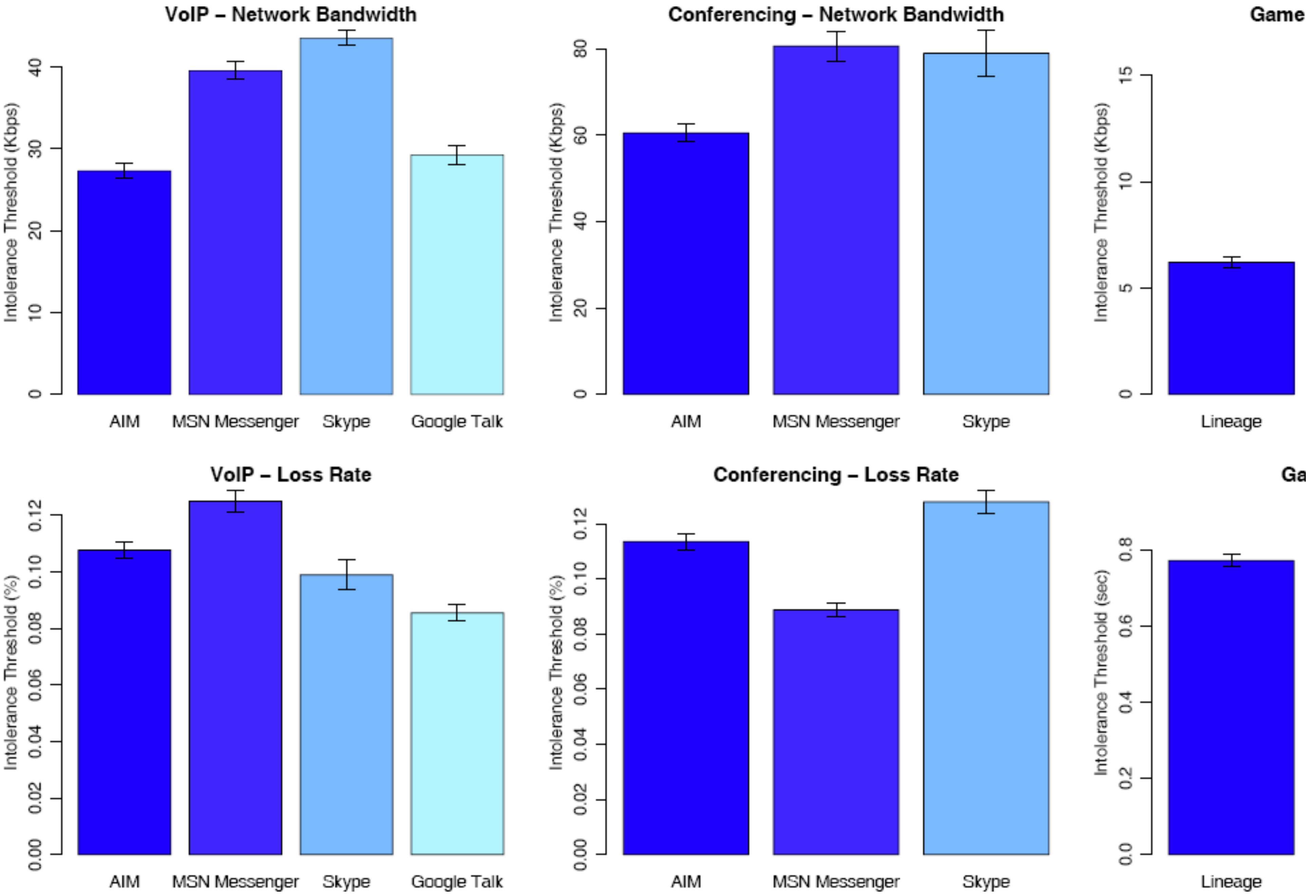
Future Work

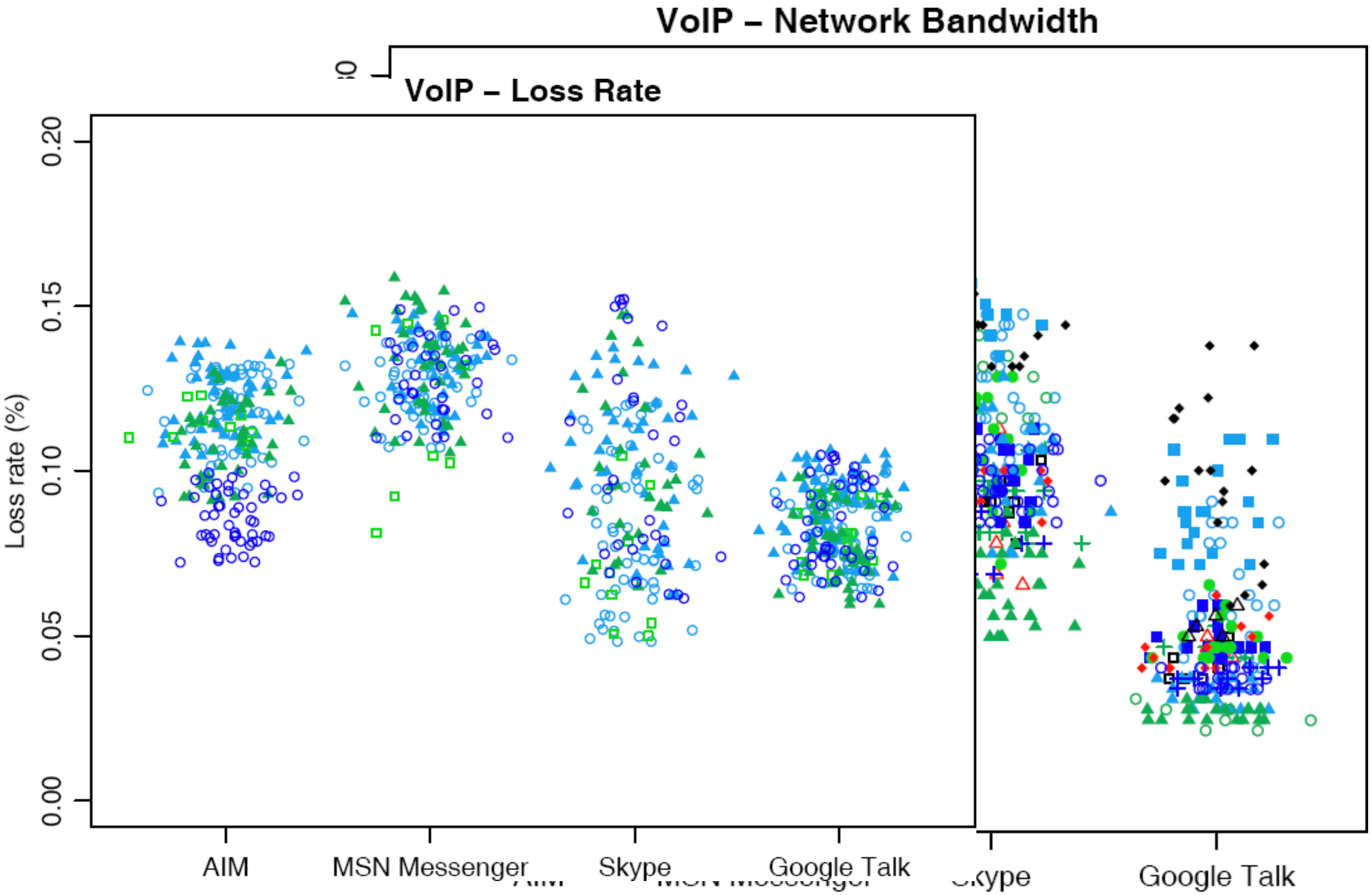
- More sophisticated input verification mechanism
  - per user (in addition to per experiment)
  - reducing false alarm
- Incentives provisioning
  - pricing
  - gaming
- More validation traces
  - user behavior analysis


nferencing – Network Bandwidth
MSN Messenger
Skype


Game – Network RTT
Round−trip time (sec)
0.0
0.5
1.0
1.5
Lineage
World of Warcraft
Unreal Tourament


VoIP – Network Bandwidth
VoIP – Loss Rate
Loss rate (%)
0.00
0.05
0.10
0.15
0.20
AIM
MSN Messenger
Skype
Google Talk
Skype
Google Talk


VoIP – Network Bandwidth
Intolerance Threshold (Kbps)
0
10
20
30
40
AIM
MSN Messenger
Skype
Google Talk
Conferencing – Network Bandwidth
Intolerance Threshold (Kbps)
0
20
40
60
80
AIM
MSN Messenger
Skype
Game
Intolerance Threshold (Kbps)
0
5
10
15
Lineage
VoIP – Loss Rate
Intolerance Threshold (%)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
AIM
MSN Messenger
Skype
Google Talk
Conferencing – Loss Rate
Intolerance Threshold (%)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
AIM
MSN Messenger
Skype
Ga
Intolerance Threshold (sec)
0.0
0.2
0.4
0.6
0.8
Lineage


VoIP – Network Bandwidth
VoIP – Loss Rate
Loss rate (%)
0.00
0.05
0.10
0.15
0.20
AIM
MSN Messenger
Skype
Google Talk
Skype
Google Talk