RTP Scaling for Very Large Groups

Henning Schulzrinne and Jonathan Rosenberg
Dept. of Computer Science and Dept. of Electrical Engineering
Columbia University
schulzrinne@cs.columbia.edu

COMET Seminar

February 13, 1997

Overview

- RTP
- RTCP scaling problems
- flood prevention
- polling
- implementation issues

RTP

- Internet standard for real-time data (audio, video, measurements, . . .)
- ITU H.323 ➠ Microsoft Netmeeting, Intel Videophone

Data:

- end-to-end transport of streams with timing information
- labeling: source, content
- encryption, sequencing, loss detection, . . .

RTCP

Control = RTCP:

- send via multicast to same group (“any-to-any”)
- member names:
  - member list
  - loop detection
  - multiple media $\rightarrow$ single conference
- sender reports ➠ rate estimation, lip-sync
- receiver reports ➠ loss, jitter, delay feedback ➠ adaptive applications

RTCP Algorithm

- RTCP packets sent periodically
- in $[1.25\text{s}, 3.75\text{s}]$ after joining
- media bandwidth b, RTCP packet size p, session size N
- period

$$T = \max(5\text{s}, \frac{pN}{0.05b \cdot 0.75})$$

- compute $\rightarrow$ set timer $\xrightarrow{T}$ send
- randomized $[0.5, 1.5] \leftrightarrow$ packet synchronization

RTCP Scaling Problems

works fine for high-speed networks, smaller groups (< 1000)

initial transient: lots join at the same time: "movie at ten", channel surfing ➠ flooding, announcements lost

state storage: each receiver stores whole receiver list

slow estimate: learn only $\leq 1/\mathrm{s}$ for 28.8 kb/s

QOS transients: can't distinguish 50% loss, 1 minute vs. 5%, 10 minutes

multicast: DVMRP prune timeout ➠ RTCP packets flooded everywhere

RTCP Transient Suppression

Conditional reconsideration: if N at send instant $> N$ at timer computation, recompute T and check interval

Unconditional reconsideration: recompute even if N hasn't increased

SendMax: accelerate convergence for new members

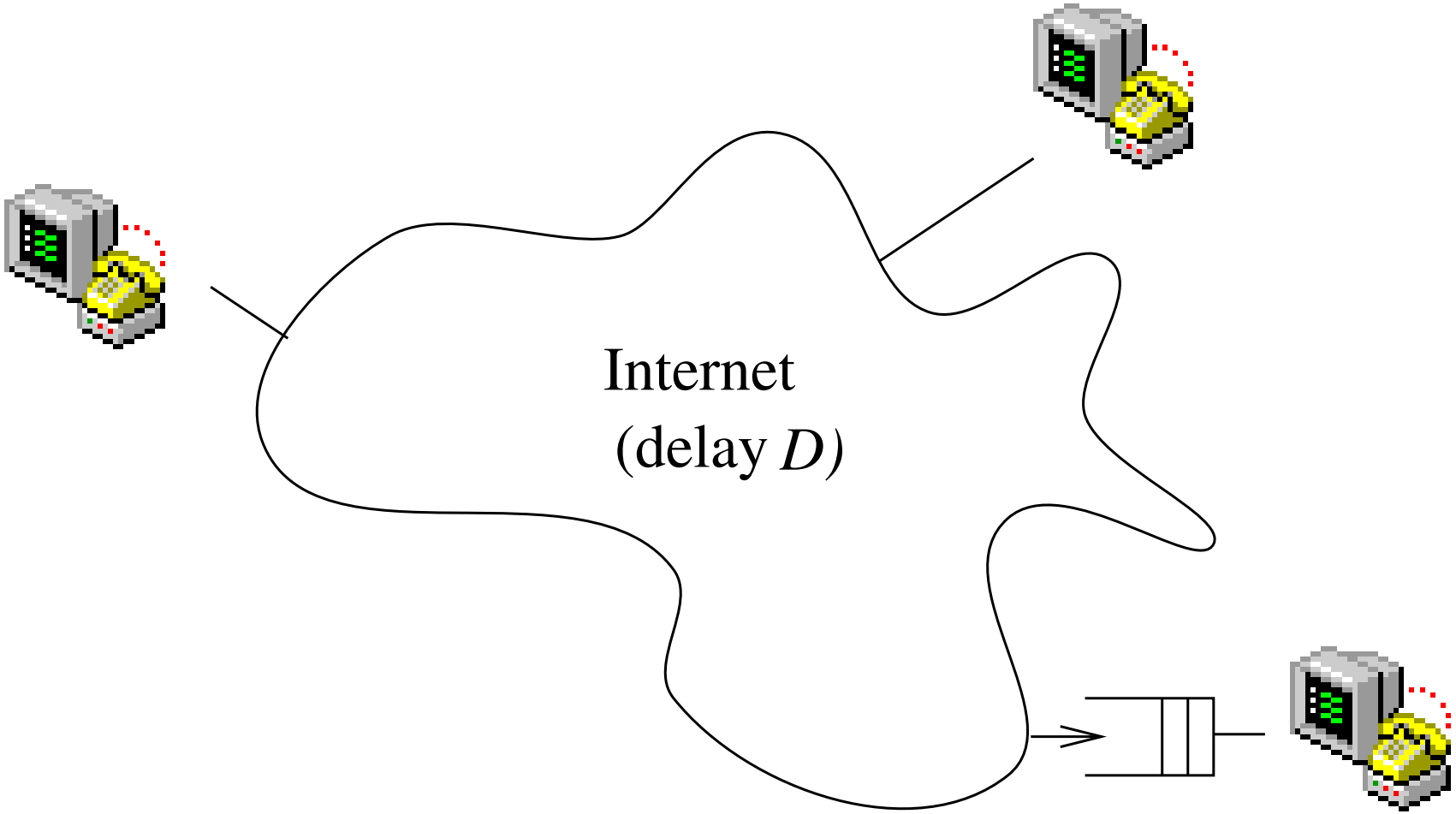
- send in RTCP packet: observed number of users
- compute interval $T = f(\max(\text{observed}, \text{allannounced}))$
- must track announcement for each member

Characteristics:

Learning curve: $L(t)$ = estimate at each member

Packets sent: ideally (no loss): = $L(t)$

Reconsideration: Network Model

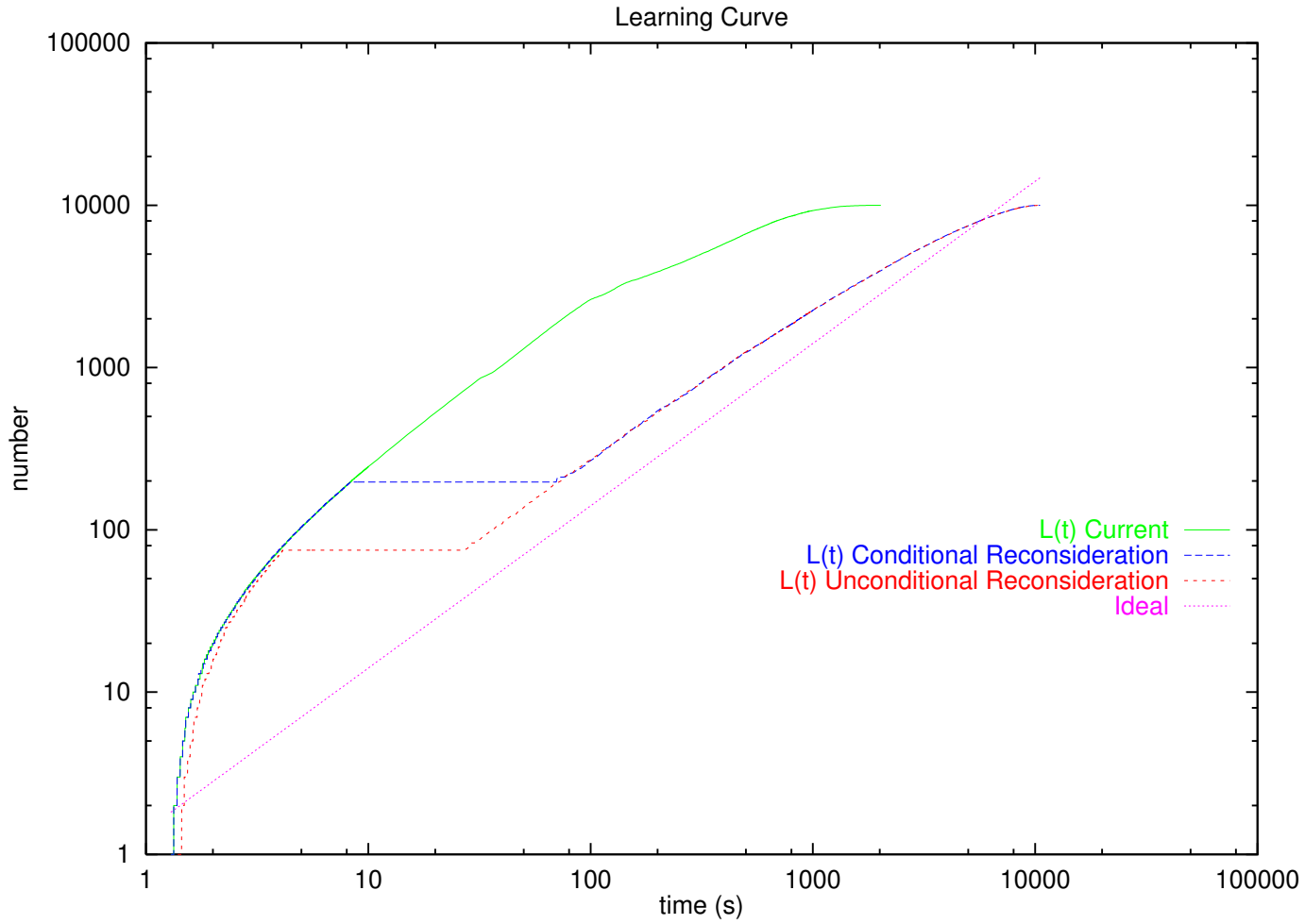


Reconsideration: Experiments

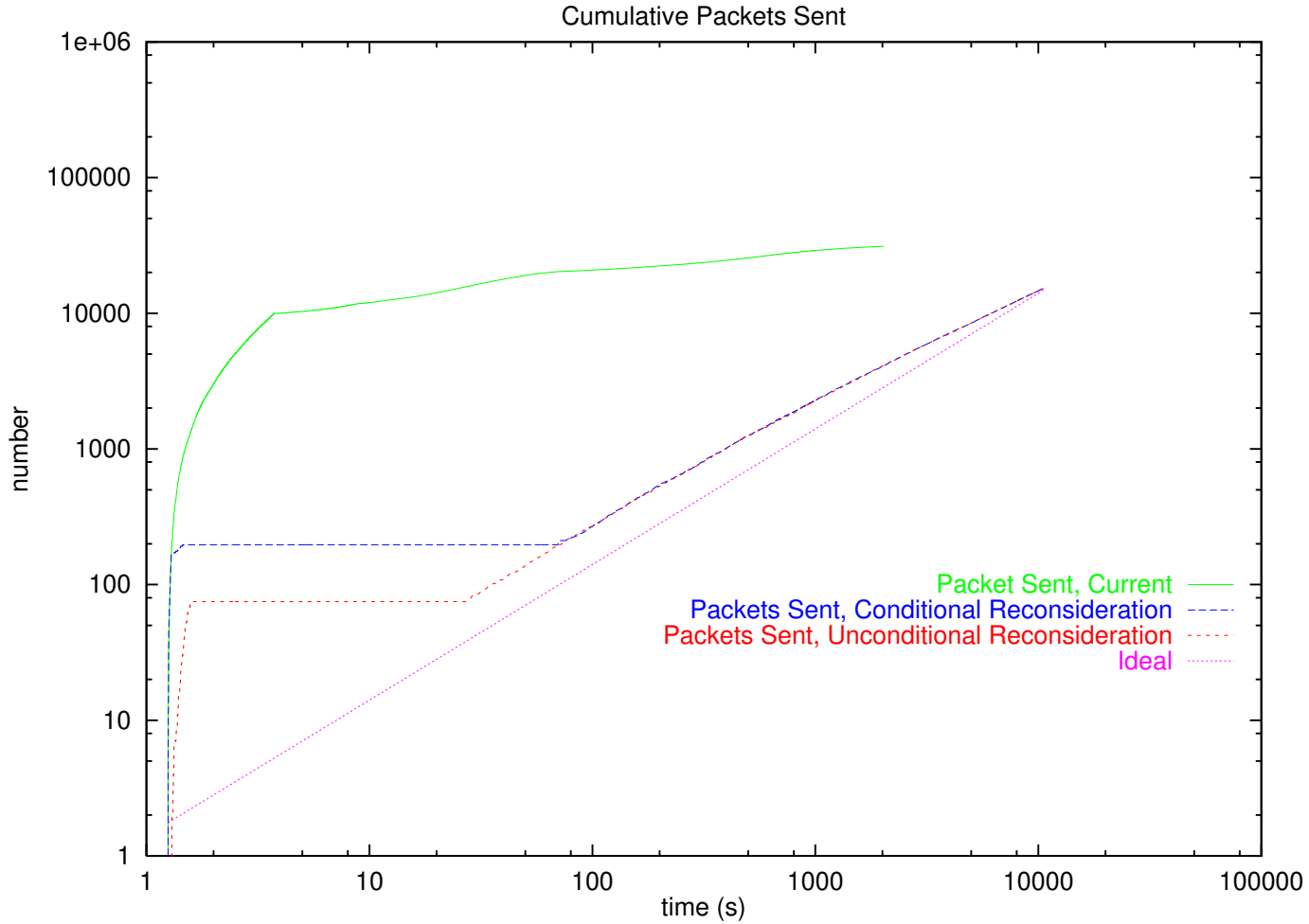
- 10,000 modem (28.8 kb/s) users in group
- join at $t = 0$ or 200 users every 200 ms
- 300 ms average delay, uniformly on [0 ms, 600 ms]
- 100 kB network buffer
- 128 byte RTCP packets

Note: backwards-compatible with existing spec

Reconsideration: Step Join



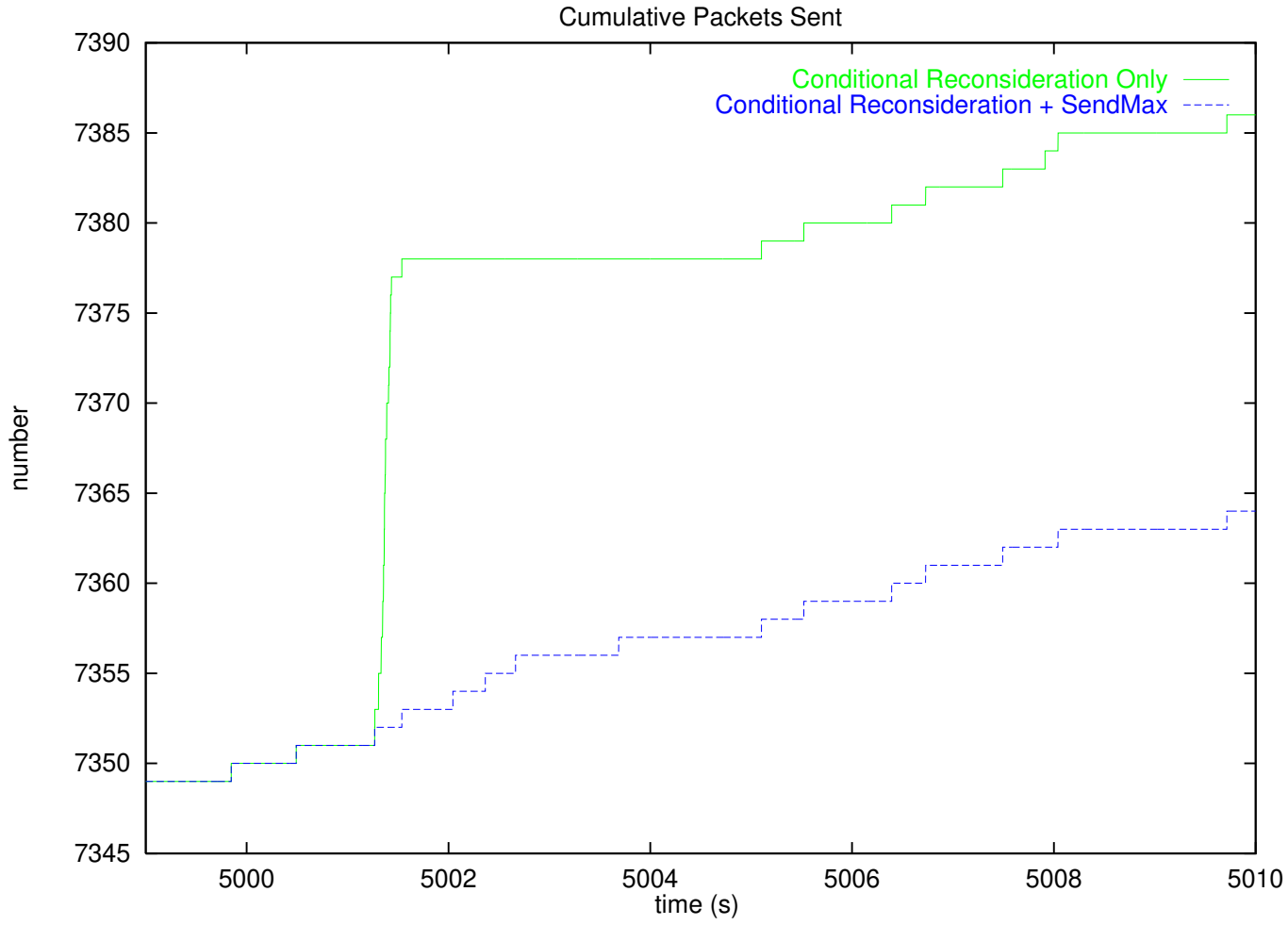
Reconsideration: Step Join



delay ➠ worse; slight dependence on distribution

Reconsideration: Linear Join

$t = 0$: +5000, $t = 5000$: +5000



Reconsideration: Analysis

- all users join at $t_0 = 0$
- initial analysis: no delay, loss
- all users are receivers
- C: nominal packet interval *for whole session*
- RTCP transmission

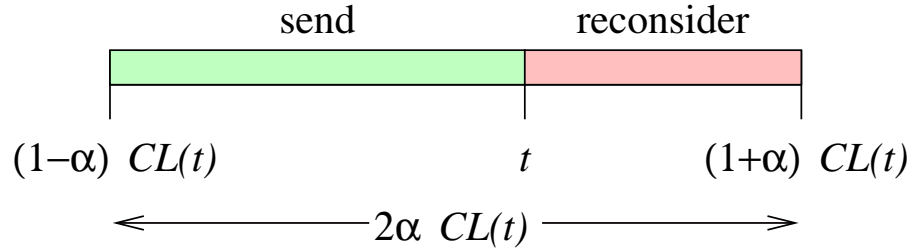
$$t_n = t_{n-1} + R(\alpha)\max(5, CL(t_{n-1}))$$

  where $R(\alpha)$ uniformly in $(1-\alpha)\ldots(1+\alpha)$

- $\alpha \in (0, 1)$

Reconsideration: Analysis

given random instant t for first transmission:



- $P_{\text{send}} = \frac{t-(1-\alpha)CL(t)}{2\alpha CL(t)}$
- don't send if $t > (1-\alpha)CL(t)$
- maximum learning curve: mt, where $m =$ modem rate
- maximum learning curve with reconsideration:

$$L_{\max}(t) = \frac{1}{(1-\alpha)C}t > L(t)$$

- steady state: $L(t) = t/C$

Reconsideration: Analysis

- packets scheduled with $d(t)$ = initial packets scheduled

$$\frac{dL}{dt} = d(t)P(t, L(t))$$

- scheduled rate $d(t)$:

$$d(t) = \frac{N}{C\frac{1-\alpha}{2-\alpha}L(t)}$$

- ➠ ODE, solvable

Reconsideration: Delay

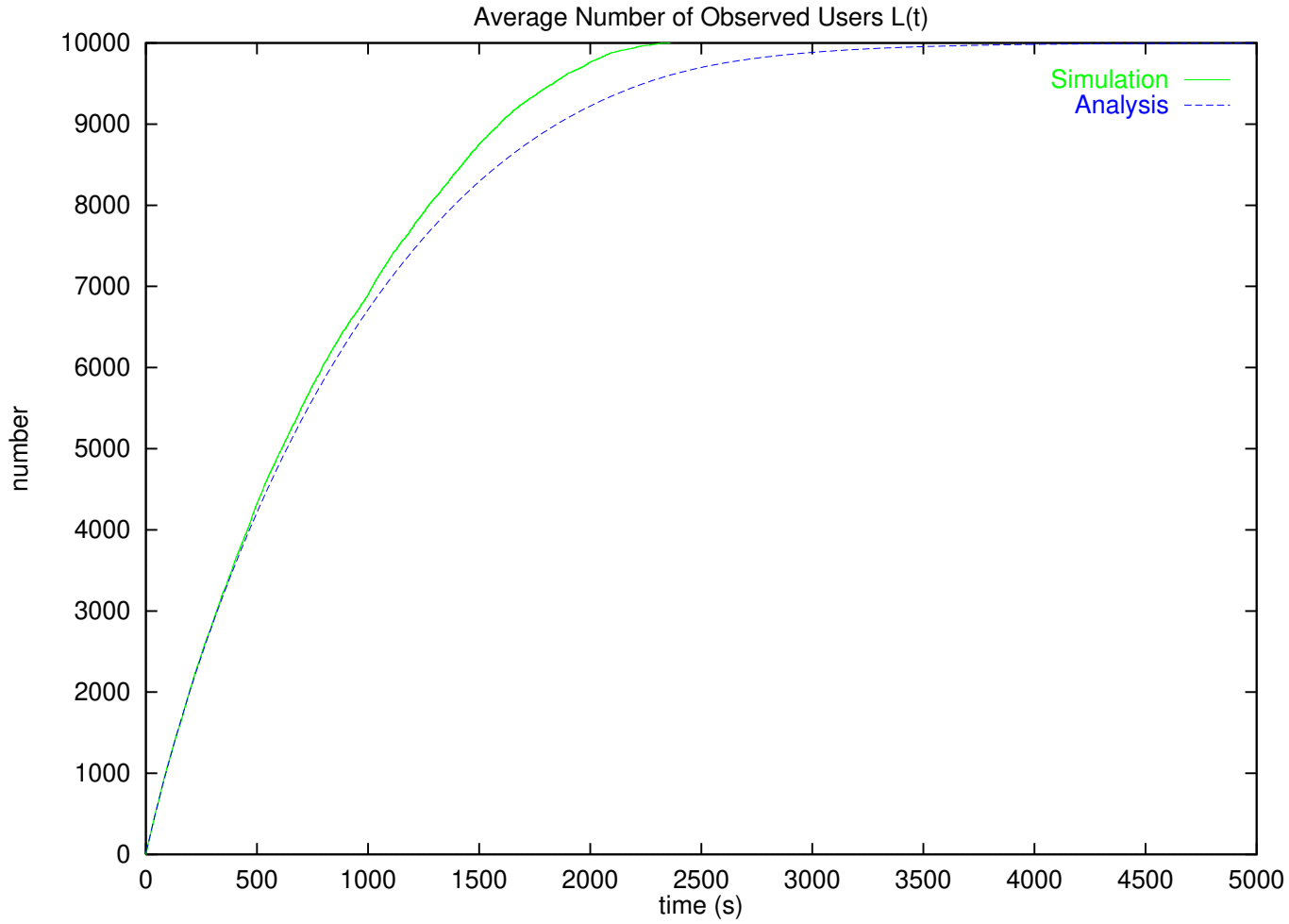
- always: transmission delay (35 ms for modem)
- propagation delays
- no reconsideration: $n = N$
- packets sent for conditional reconsideration, delay D, modem speed m:

$$n = \frac{N}{2\alpha T_{\min}} \left(D + \frac{1}{m} \right)$$

- unconditional reconsideration:

$$n \sim N \left(\frac{D + c\frac{1}{m}}{T_{\min}} \right)^2$$

Analysis vs. Simulation



RTCP State Maintenance

- need several Mbytes just to count users
- sources identified by 32-bit SSRC, random
- interarrival for single user too long, may be gone, mislead others ➠ flooding
- maintain fixed sampling buffer
- keep only fraction of announcements (bit mask, range, threshold)
- if buffer about to overflow:
  - reduce acceptance probability
  - remove non-matches from buffer
- if below threshold, increase probability, maintain count

RTCP Group Size Estimation

- previously: any-to-any feedback
- only get to know 3,600 members/hour ➠ not suitable for CATV
- often, only sender cares about group size, QOS ➠ no multicast needed
- may want to get repeat QOS from same member

➠ polling instead of announcements

- send SSRC (32 bits), mask, time interval
- respond within time interval only if SSRC match
- allows repeat sampling or new
- on average Np responses, roughly normally distributed

Summary

- RTCP scaling problems: flooding, convergence, state maintenance
- ➠ reconsideration, polling, statistical sampling
- applicable to other distributed learning problems?
- e.g., McCanne/Jacobson video adaptation