Scalable solutions for information collection

Chadi BARAKAT

INRIA Sophia Antipolis, France
Planète group

Email: Chadi.Barakat@sophia.inria.fr
WEB: http://www.inria.fr/planete/chadi

Problem statement

- Many current situations in which one is interested in collecting information from a large number of sources spread over the Internet.
  - Reports on receivers within a multicast session.
  - Measurements collected by hosts, routers, sensors or traffic capture devices.
  - Data generated by the branches of a distributed company.
  - etc.
- Challenge: This collection, if done simultaneously, could congest the network and cause implosion at the collector.
- Transport solutions are needed. SCALABALITY too !

Information filtering

- Some information don't need to be collected entirely:
  - One piece is enough in case of clients asking a multicast source to retransmit a packet.
  - A subset is enough in case of applications looking for some general function calculated over the entire information set.
    - Average temperature, std, distribution, etc.
    - Number of active clients.
    - Statistics on particular flows inside the network.
    - Statistics on Internet hosts.
- Other information needs to be entirely collected.
  - Quality of service received by the different clients for billing purposes.
  - Network monitoring. Banking operations. etc.

Framework for the study

- We look for end-to-end solutions. No intermediate nodes are deployed to aggregate the information (as in ConCast for example)
- Two case study:
  - Counting the number of clients (or sources). The information in this case is identical and filtering can be done to reduce the overload on the network and the collector.
    - Counting is done by probabilistic filtering and periodic probing.
    - Validation on real traces.
  - Information to be entirely collected.
    - We develop TICP, a TCP-friendly Information Collection Protocol.
    - TICP provides congestion and error control functionalities.
    - Validation with ns-2.

Counting the number of clients in a multicast session

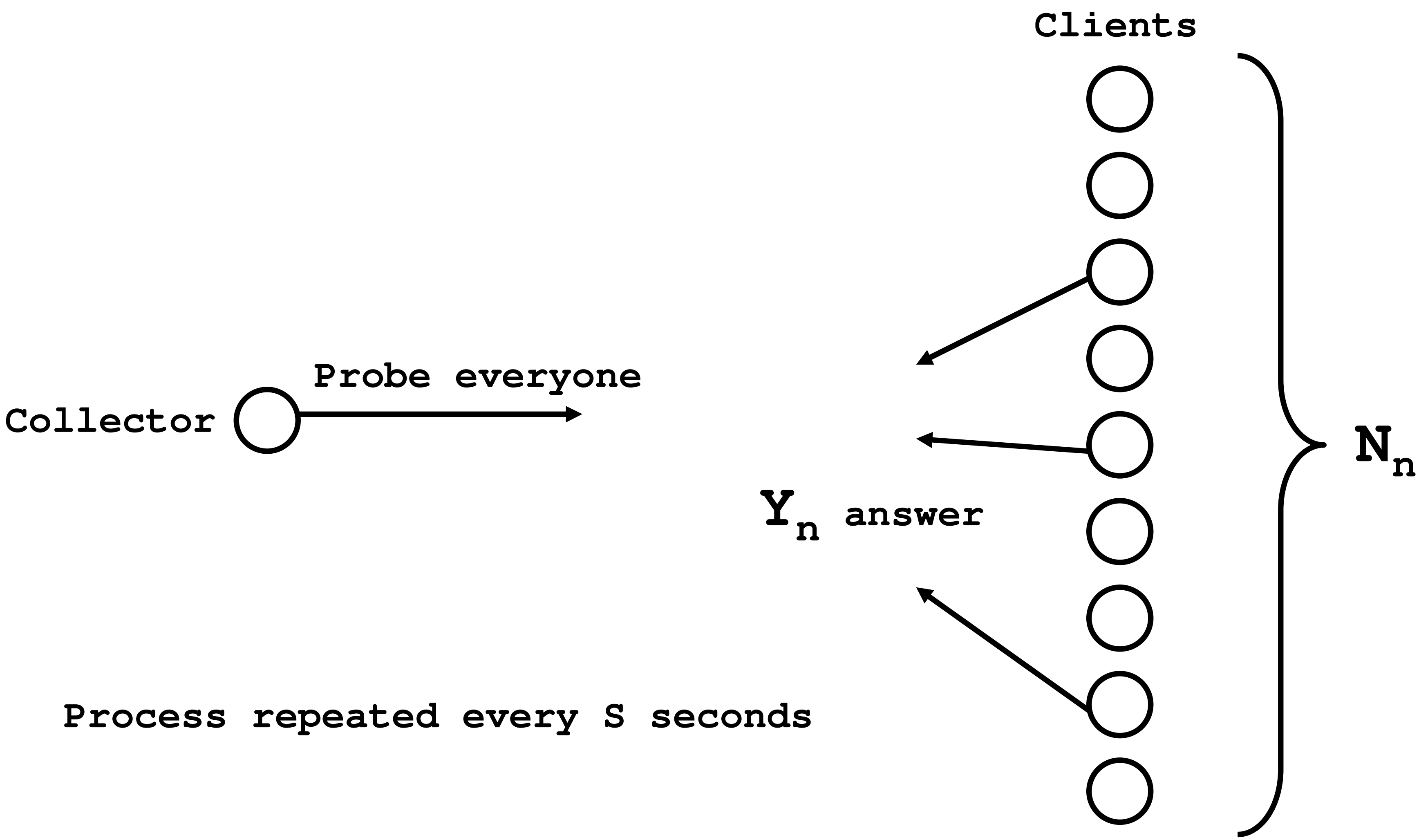
Counting the number of clients in a multicast session

- Interesting multicast applications (distance learning, video-conferences, events, radios, televisions, live sports, etc.)
- Membership is required for:
  - Feedback suppression (RTP, SRM).
  - Tuning amount of FEC packets for reliable multicast.
  - Stopping transmission when no more receivers.
  - Pricing.

  and especially for radios and future TVs, to:
  - Characterize audience preferences
  - Adapt the transmission content

Counting the number of receivers in a multicast session

- Problem of ACK implosion in case of large sessions:
  - The solution is to ask clients to send periodically ACKs to the collector with probability " p ". The collector has then to develop its own estimators to infer the number of clients.
- Methodology:
  - Collector: Periodically requests from clients to send ACKs with probability " p " every " S " seconds.
  - Clients: Every S seconds, send ACK to collector with probability p .
  - Collector: Stores Y_n number of ACKs received at time nS .
  - Objective: Use noisy observation Y_n to estimate membership $N_n = N(nS)$.


Clients
Probe everyone
Collector
N_n
Y_n answer
Process repeated every S seconds

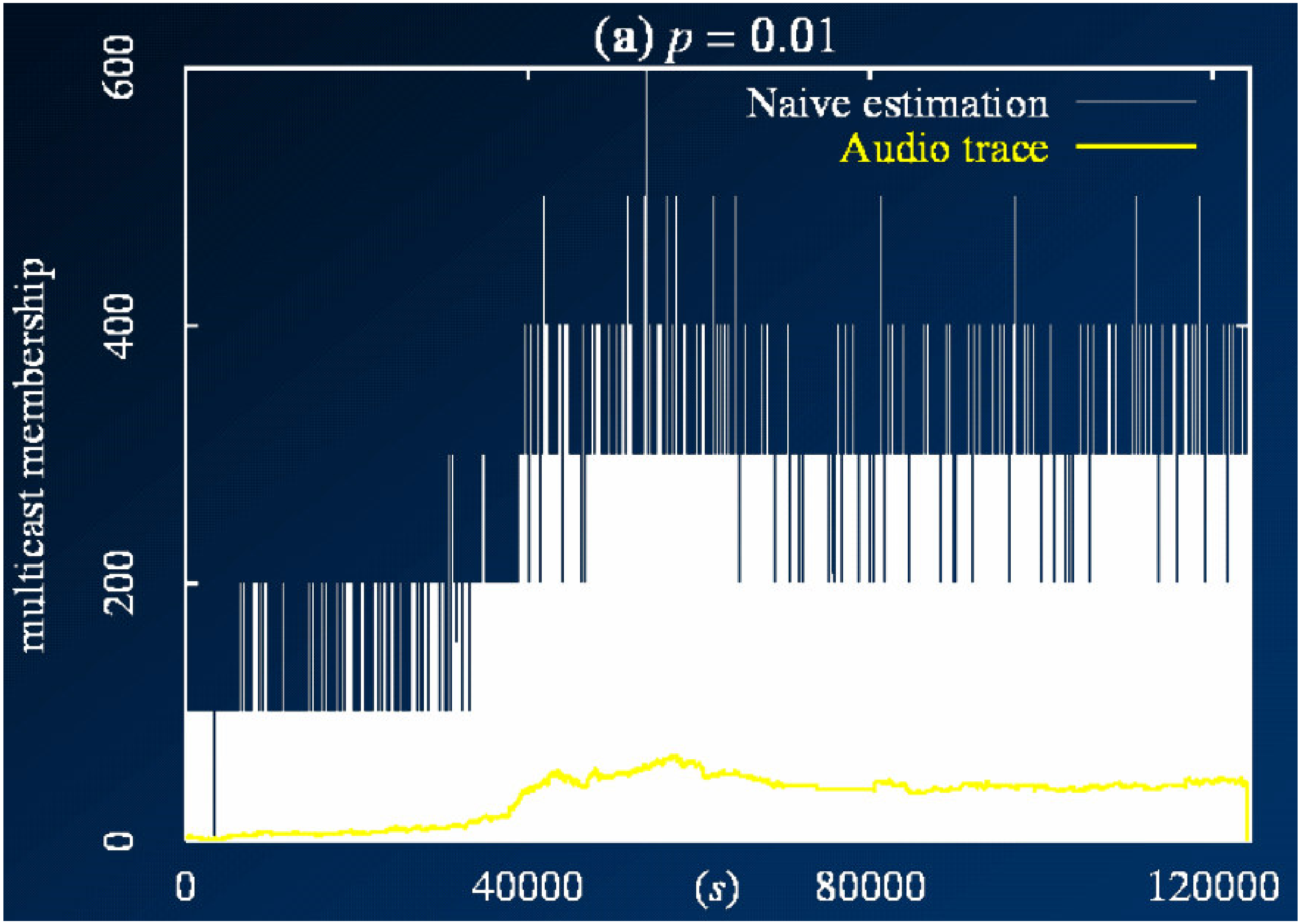
Naive estimation

$$\hat{N}_n = \frac{Y_n}{p}$$

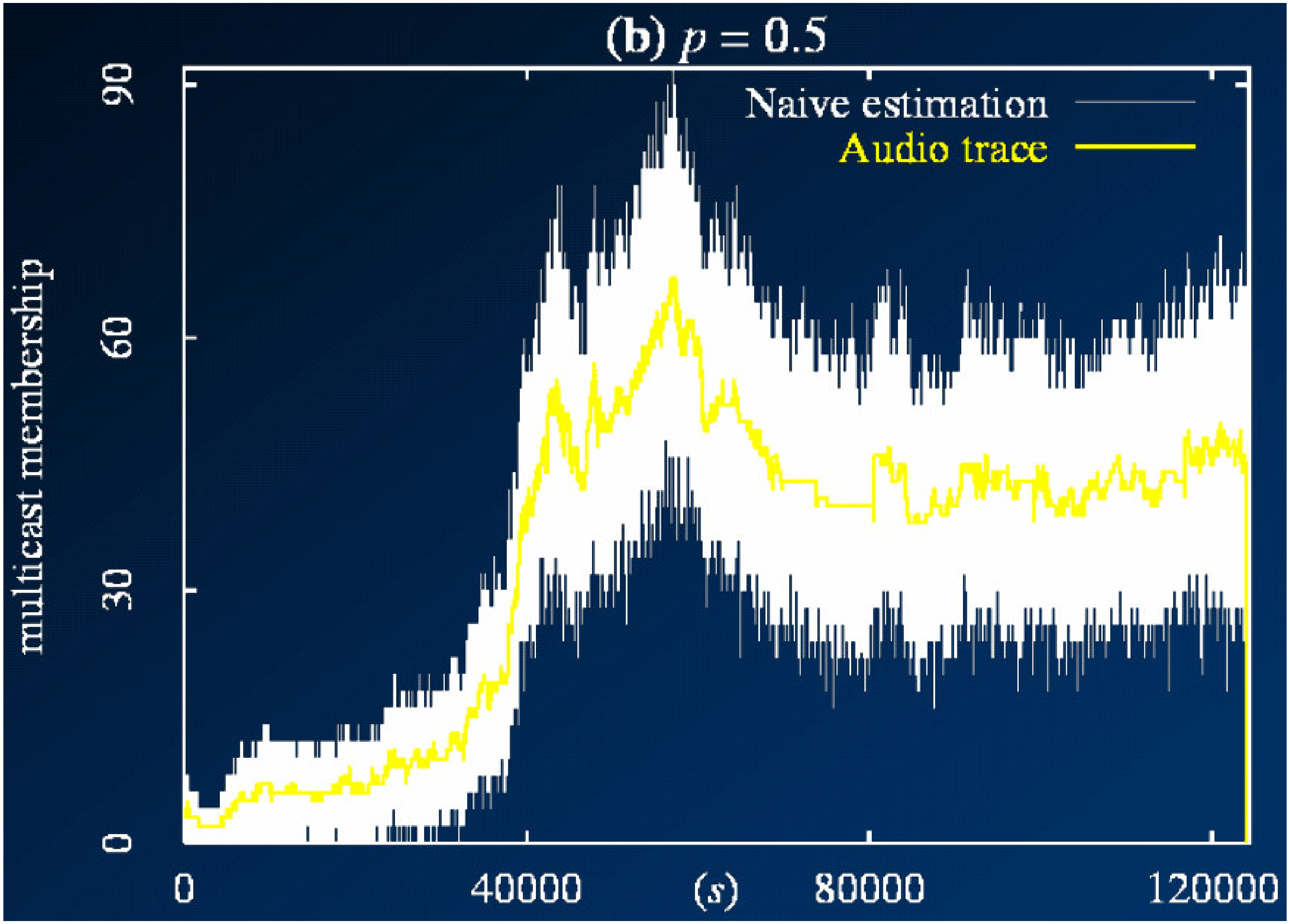
Drawbacks:

- Very noisy (s.l.l.n. $\lim_{N \to \propto} Y/N = p$).
- No profit from correlation (no use of previous estimate).

Naive estimation : p = 0.01



Naive estimation : p = 0.5



EWMA estimation

$$\hat{N}_{n,a} = \alpha \hat{N}_{n-1,a} + (1-\alpha)\frac{y_n}{p}$$

$$0 < \alpha < 1$$

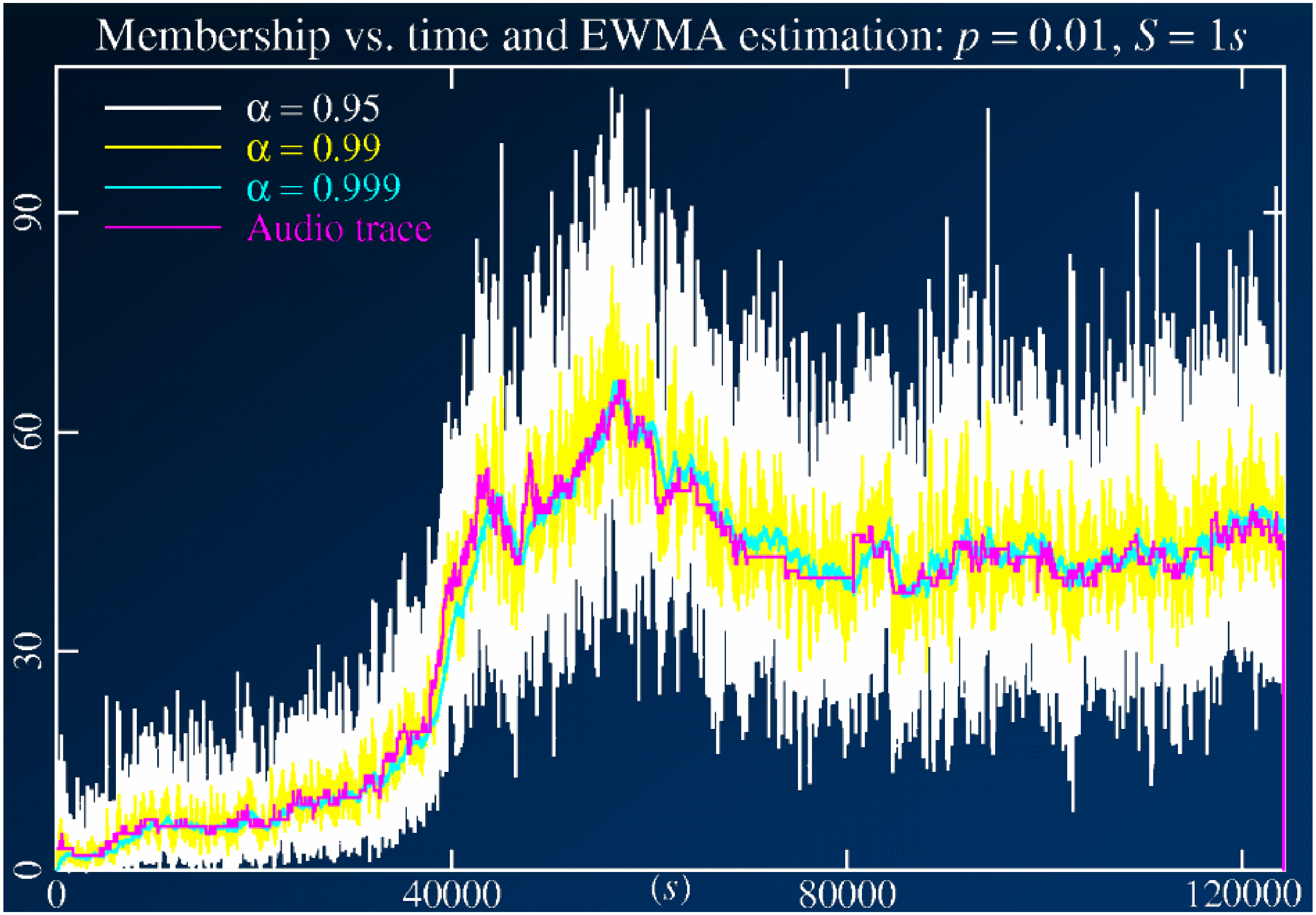
Advantages:

- Use of previous estimate.
- No a priori information needed.

Drawbacks:

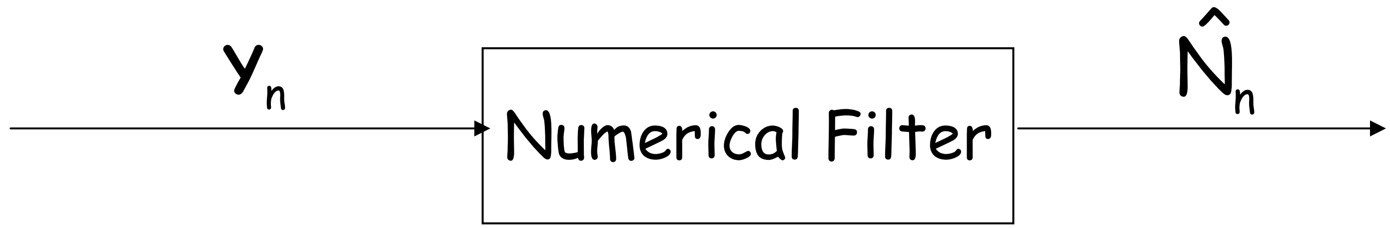
- What value for α ?
- Estimator does not depend on ACK interval S.

EWMA estimation



Estimation using filter theory

- Noisy observation Y_n:
  - Centered version $y_n = Y_n - E[Y_n]$, $E[y_n] = 0$.
- Desired signal N_n:
  - Centered version $\nu_n = N_n - E[N_n]$, $E[\nu_n] = 0$.
- Filter output $\hat{N}_n$ (resp. $\hat{\nu}_n$) estimation of N_n (resp. ν_n)



Find the optimal linear filter that minimizes the mean-square error,

i.e. $E[(\hat{N}_n - N_n)^2]$

Wiener filter = Optimal Linear Filter

Introduce:

power spectrum of $\{y_n\}_n$, $S_y(z) = \sum_{k=-\infty}^{\infty} Cov_y(k) z^{-k}$

z - transform of $Cov_{\nu y}(k)$, $S_{\nu y}(z) = \sum_{k=-\infty}^{\infty} Cov_{\nu y}(k) z^{-k}$

Canonical factorization, $S_y(z) = \sigma G(z) G\left(z^{-1}\right)$

$G(z)$: part of $S_y(z)$ having its zeros and poles inside the unit cercle

Compute $H(z) = \left[\dfrac{S_{\nu y}(z)}{G\left(z^{-1}\right)}\right]_+ \Rightarrow H_o(z) = \dfrac{H(z)}{\sigma G(z)}$

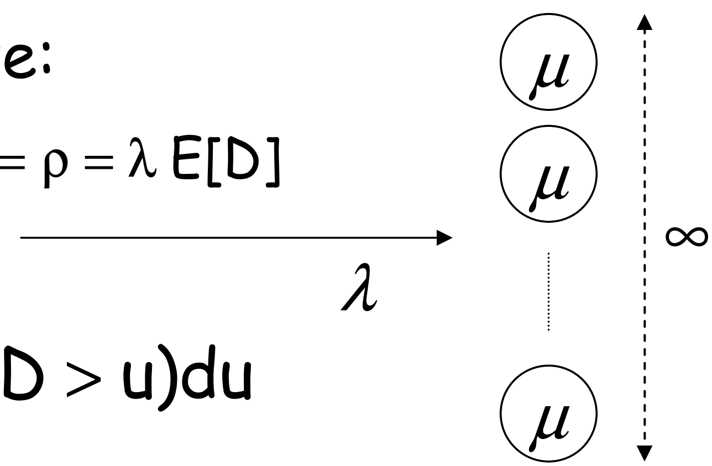
$H_o(z) = \dfrac{N(z)}{Y(z)}$ is the transfer function of the optimal filter

M/G/∞ model for the session

- One needs a model for the system in order to compute $H_o(z)$
  - Participants arrive according to a Poisson process of intensity λ
  - On-times have common probability distribution and are independent; **D** denotes a generic random variable of average $1/\mu$.

$\Rightarrow$ N(t) is then the occupation process in the M/G/∞ queue

- Characteristics of N(t) in steady-state:
  - Poisson random variable, Mean = Variance = $\rho = \lambda E[D]$
  - Autocorrelation function



$$Cov(N(t), N(t+h)) = \lambda \int_{|h|}^{\infty} P(D > u)du$$

Application to M/M/∞ model

When $D \sim Exp(\mu)$

$$Cov_v(k) = \rho\gamma^{|k|}, \quad \gamma = \exp(-\mu S)$$
$$Cov_{vy}(k) = p\, Cov_v(k)$$
$$Cov_y(k) = p^2 Cov_v(k) + \mathbf{1}(k=0)\rho p(1-p)$$

We find $H_o(z) = \dfrac{B}{1 - Az^{-1}}$

where $$A = \frac{1+\gamma^2(1-2p) - \sqrt{\left(1-\gamma^2\right)\left(1-\gamma^2(1-2p)^2\right)}}{2\gamma(1-p)}$$

$$B = \frac{-\left(1-\gamma^2\right) + \sqrt{\left(1-\gamma^2\right)\left(1-\gamma^2(1-2p)^2\right)}}{2\gamma^2 p(1-p)}$$

Application to M/M/∞ model

Transfer function $$H_o(z) = \frac{B}{1 - Az^{-1}}$$

Impulse response for the centered processes

$$\hat{v}_n = A\hat{v}_{n-1} + By_n$$

Optimal Linear Estimator for group membership

$$\Rightarrow \hat{N}_n = A\hat{N}_{n-1} + BY_n + \rho(1 - A - pB)$$

⌧ Auto-regressive process of order one.

Optimal first-order linear filter

- Find $A \in (0,1)$ and B such that
  - $\hat{v}_n = A\hat{v}_{n-1} + By_n$
  - mean-square error $\varepsilon = E\left[\left(v_n - \hat{v}_n\right)^2\right]$ minimized
- Steady-state $\hat{v}_n = B\sum_{k=0}^{\infty} A^k y_{n-k}$
- Minimize

$$\varepsilon = \rho - 2pBg(A) + \left(\frac{pB^2}{1-A^2}\right)\left(2pg(A) + \rho(1-2p)\right)$$

where $g(z) = \sum_{k=0}^{\infty} z^k Cov_v(k)$

Optimal first-order linear filter

- System to solve $\begin{cases} \dfrac{\partial \varepsilon}{\partial A} = 0 \\ \dfrac{\partial \varepsilon}{\partial B} = 0 \end{cases}$

- Solution is unique

- $D \sim Exp(\mu) \Rightarrow$ same solution as Wiener filter

- $D \sim Hyperexpnential(L, \mu_i, p_i, i = 1 \ldots L)$

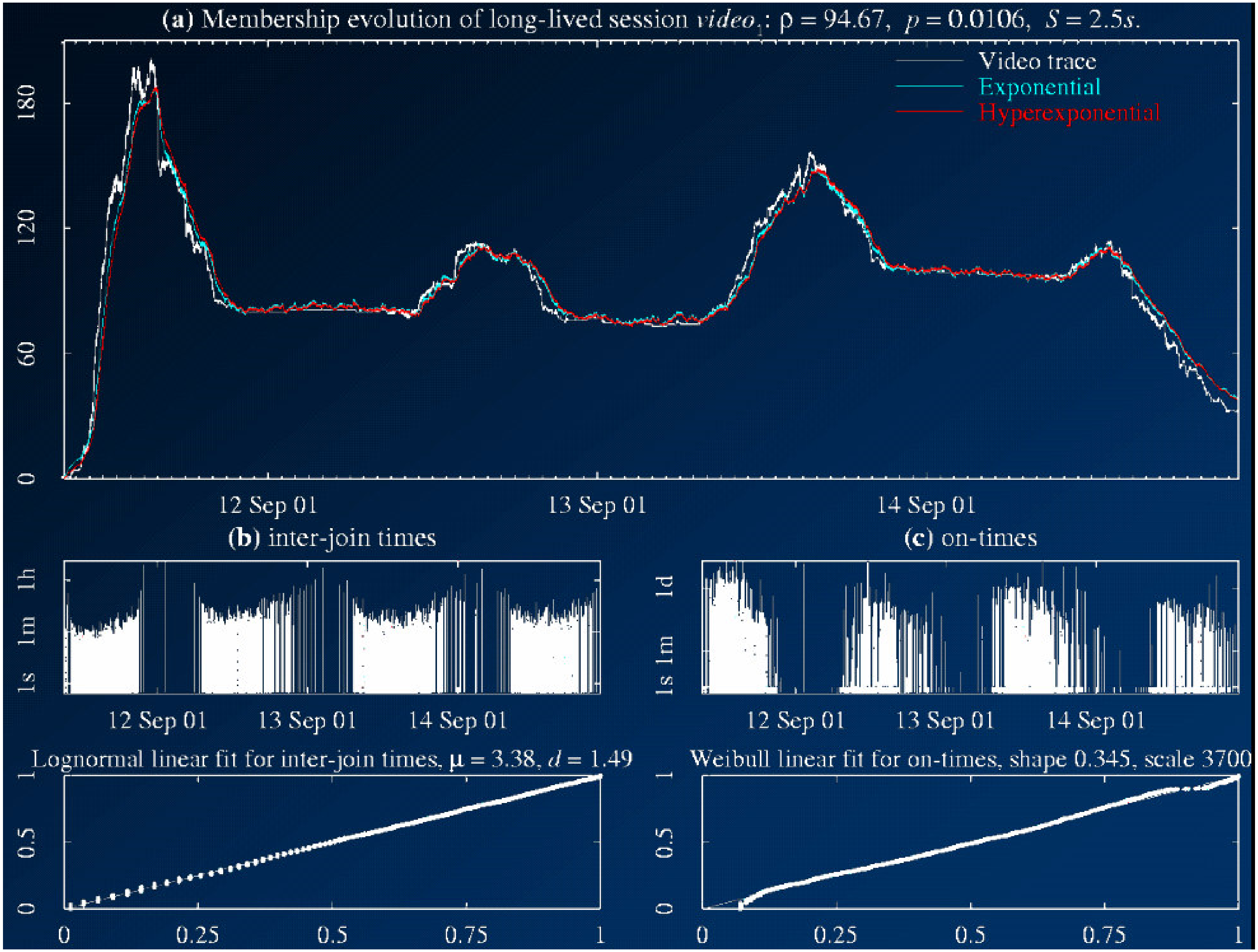
  $\Rightarrow$ Numerical solving

Validation with real traces

- Wiener filter (optimal linear filter)
  - $M/M/\infty$ model
  - Estimator $\hat{N}_n^E$
  - ρ and μ are assumed known
- Optimal first-order linear filter
  - $M/H_2/\infty$ model
  - Estimator $\hat{N}_n^{H_2}$
  - ρ, μ_1, μ_2 and p_1 are assumed known ($p_2 = 1 - p_1$)

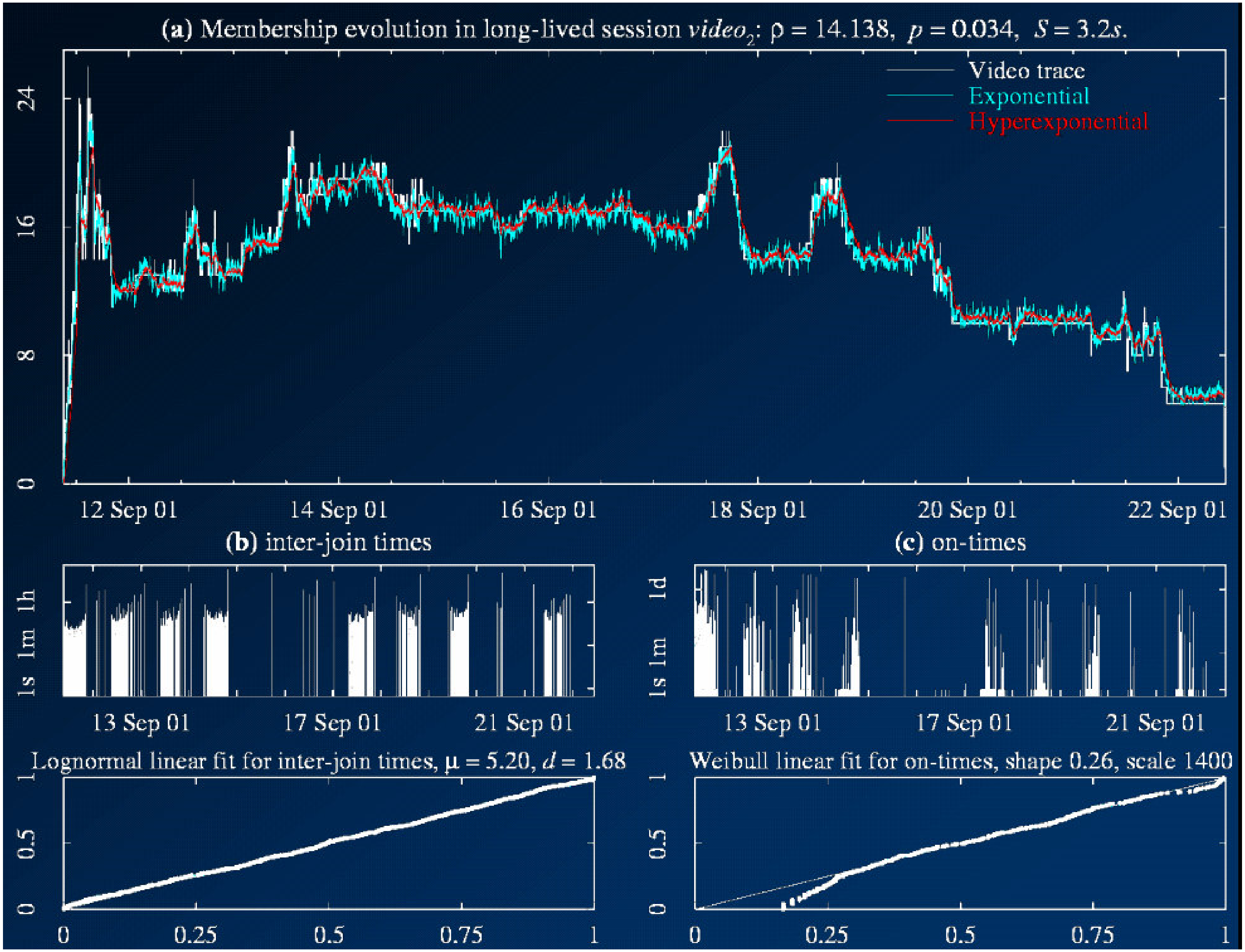
For values of p.E[N] of the order of 1, we found a relative error of few percents for both estimators.

$p \times \rho = 1$

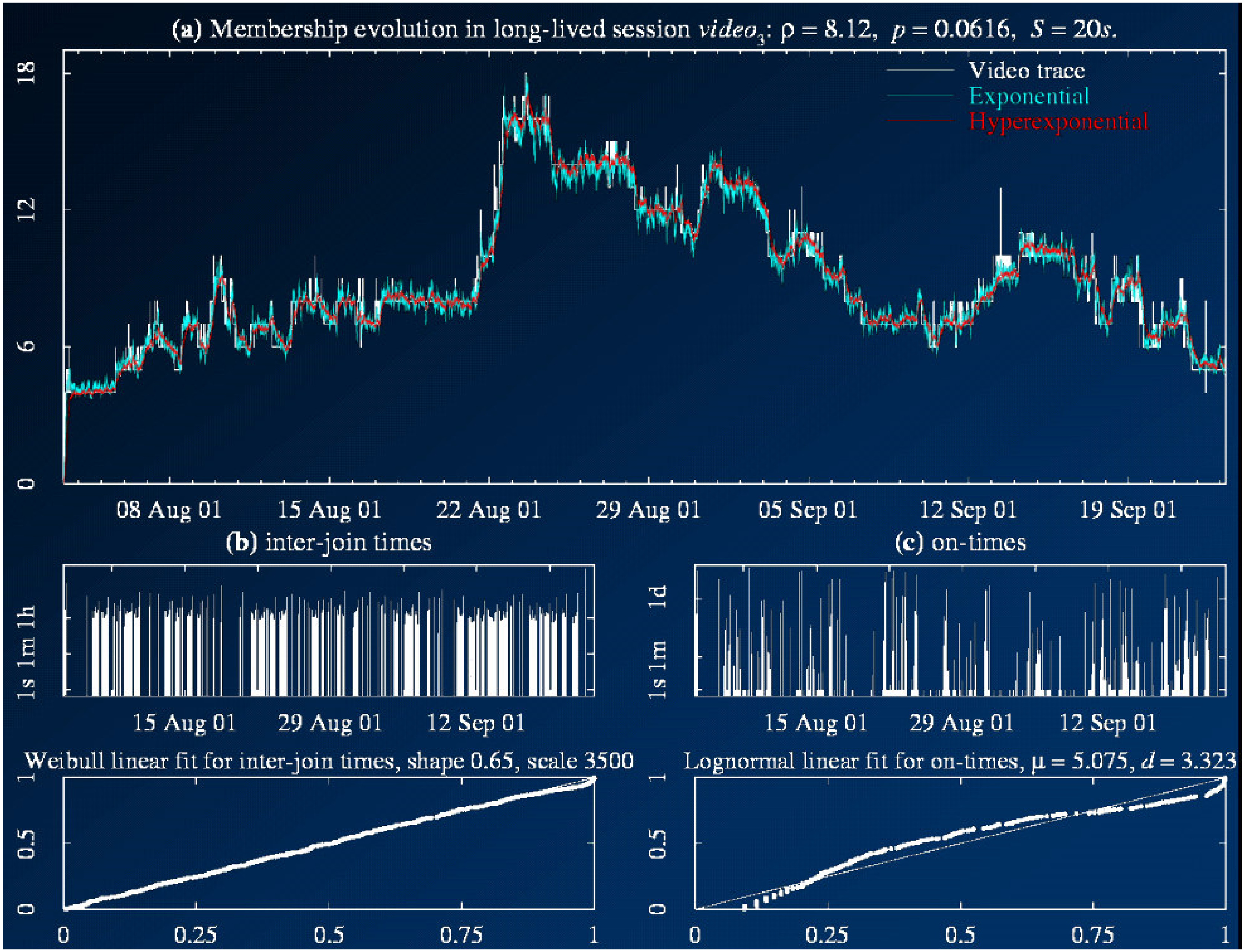


(a) Membership evolution of long-lived session video₁: ρ = 94.67, p = 0.0106, S = 2.5s.
Video trace
Exponential
Hyperexponential
0
60
120
180
12 Sep 01
13 Sep 01
14 Sep 01
(b) inter-join times
1s
1m
1h
12 Sep 01
13 Sep 01
14 Sep 01
(c) on-times
1s
1m
1d
12 Sep 01
13 Sep 01
14 Sep 01
Lognormal linear fit for inter-join times, μ = 3.38, d = 1.49
0
0.5
1
0
0.25
0.5
0.75
1
Weibull linear fit for on-times, shape 0.345, scale 3700
0
0.5
1
0
0.25
0.5
0.75
1

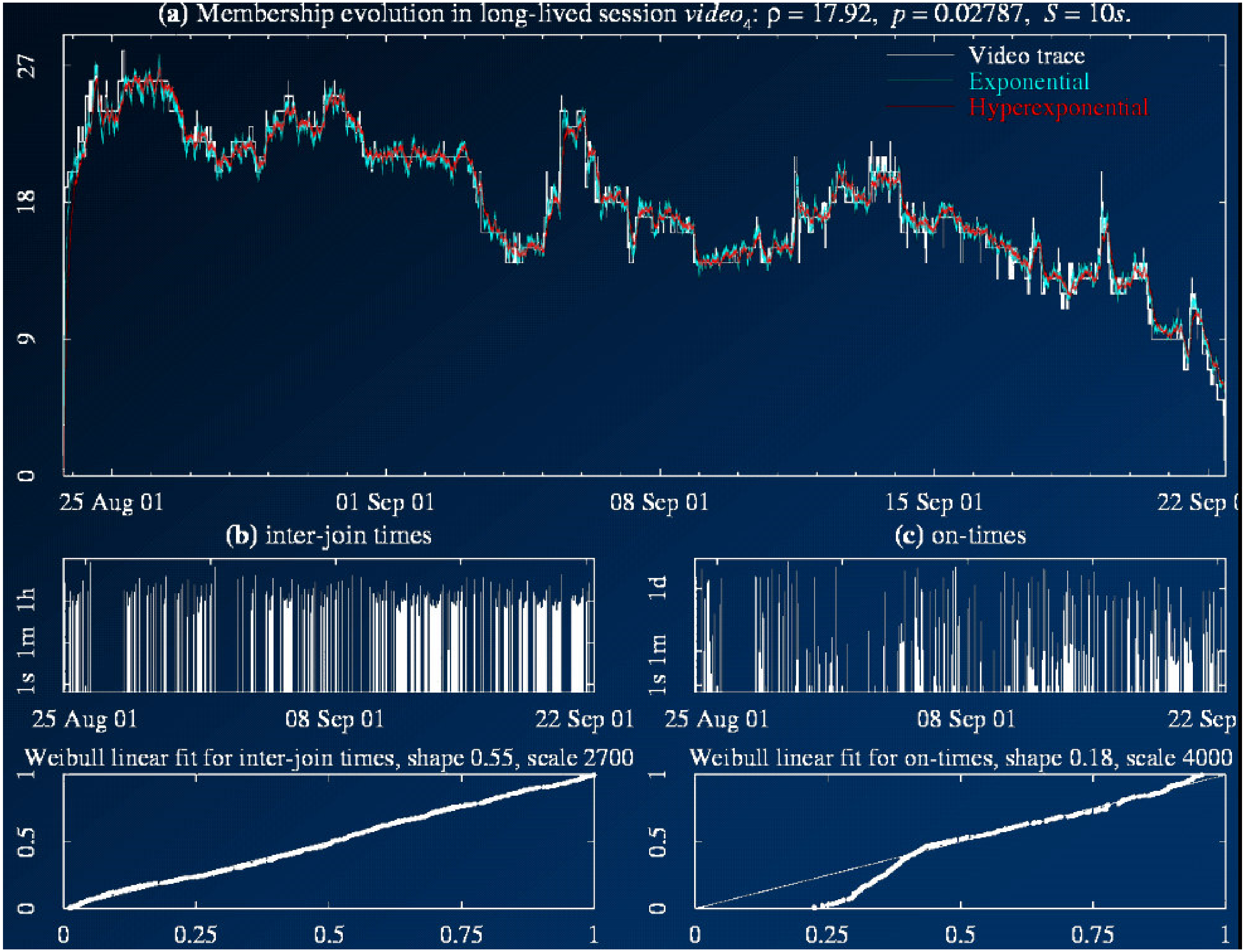
$p \times \rho = 0.5$



p × ρ = 0.5


(a) Membership evolution in long-lived session video3: ρ = 8.12, p = 0.0616, S = 20s.
Video trace
Exponential
Hyperexponential
0
6
12
18
08 Aug 01
15 Aug 01
22 Aug 01
29 Aug 01
05 Sep 01
12 Sep 01
19 Sep 01
(b) inter-join times
(c) on-times
1s 1m 1h
1s 1m 1d
15 Aug 01
29 Aug 01
12 Sep 01
Weibull linear fit for inter-join times, shape 0.65, scale 3500
Lognormal linear fit for on-times, μ = 5.075, d = 3.323
0
0.5
1
0.25
0.75

p × ρ = 0.5


(a) Membership evolution in long-lived session $video_4$: ρ = 17.92, p = 0.02787, S = 10s.
Video trace
Exponential
Hyperexponential
27
18
9
0
25 Aug 01
01 Sep 01
08 Sep 01
15 Sep 01
(b) inter-join times
(c) on-times
1s 1m 1h
1s 1m 1d
25 Aug 01
08 Sep 01
22 Sep 01
Weibull linear fit for inter-join times, shape 0.55, scale 2700
Weibull linear fit for on-times, shape 0.18, scale 4000
0
0.25
0.5
0.75
1

Entire collection of information

TICP transport protocol

Objectives

- Complete and reliable collection of information from a large number of clients.
- No constraint on the collected information:
  - Quality of reception of a TV transmission (who received what), etc.
- The information needs to arrive entirely at the collector:
  - In the literature, protocols for collecting identical information exist (e.g, collect NACKs in a reliable multicast transmission).
  - Probabilistic collection cannot be applied in our case, since the entire information needs to be received.
  - We want the solution to be end-to-end, so intermediate solutions don't work as well (concast).

Congestion control & TCP-friendliness

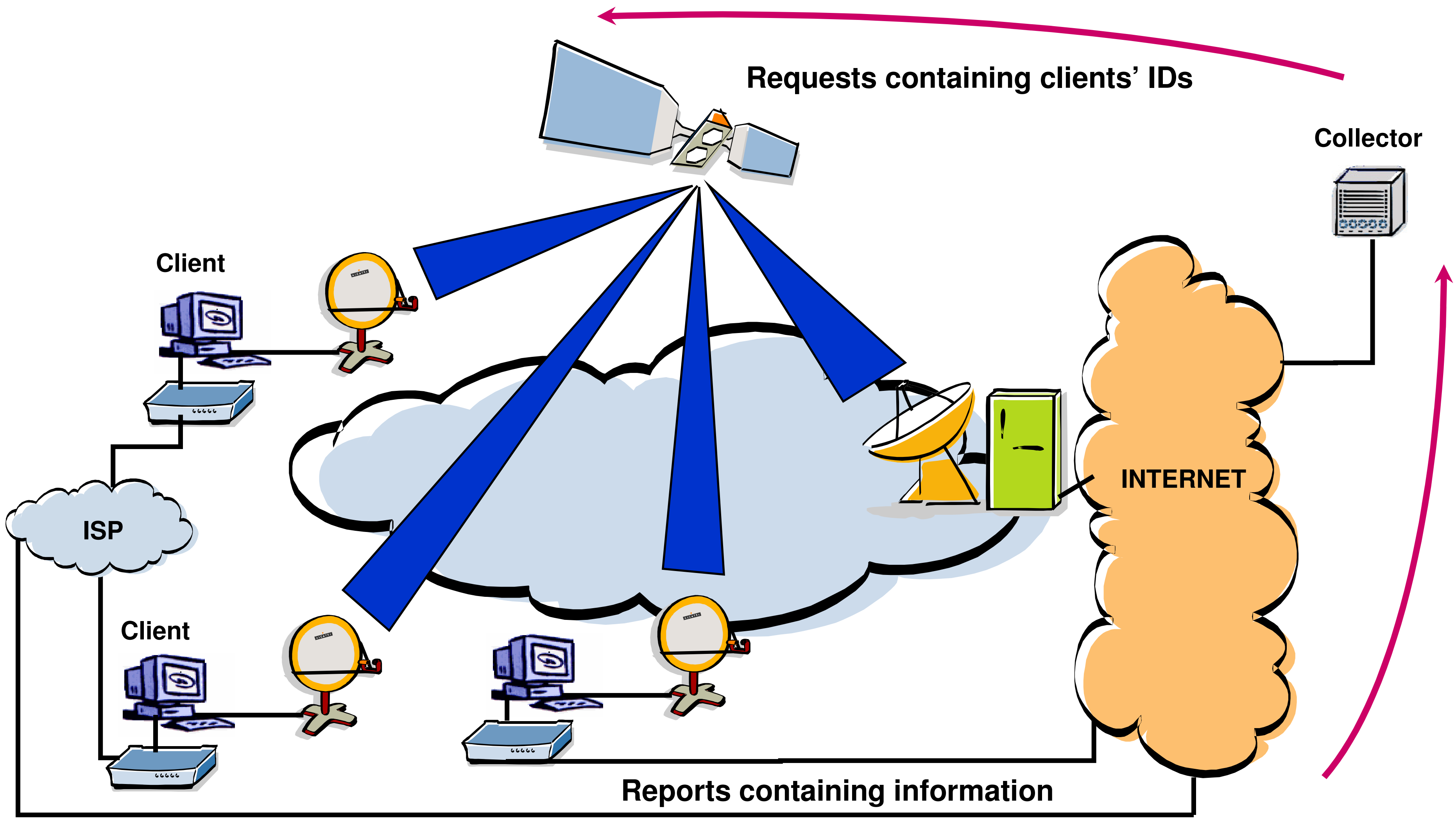
Challenges (caused by the large number of clients):

- The protocol must control the congestion of the network in the forward and in the reverse directions:
  - High throughput (Good utilization of the available bandwidth).
  - Low loss ratio (short queues in network routers).
- The protocol must be friendly with other applications, mainly with applications using TCP.
  - The protocol must be designed so as to be TCP-friendly.

Requirements

- The collector sends requests to clients via multicast.
  - The case of unicast (or P2P) is left for future research.
- The clients send their reports back via unicast.
- The collector has a list of the IDs of all clients:
  - **ID**: IP address, session ID, name of the machine, etc.
- The collector is able to probe multiple clients in one packet.
- A client sends directly a report (its information) when it receives a request packet containing its ID.

Example: Our protocol over satellite



INRIA SOPHIA ANTIPOLIS

The two extreme cases

- Stop-and-Wait collection:
  - Probe one client and wait until its information arrives.
  - When the information arrives, probe another client, and so on.

- All-at-once collection:
  - Probe all clients at the same time and wait for their reports.
  - After a certain time, consider reports that did not arrive as lost.
  - Probe clients that did not answer, wait another time, and so on.

- An optimal tuning is located somewhere between the 2 cases.

Protocol in brief: Congestion control

- A window-based flow control:
  - `cwnd`: maximum number of clients the collector can probe before receiving any report.
- The collector increases `cwnd` and monitors at the same time the loss ratio of reports (during a time window in the past).
  - The protocol has two modes: slow start and congestion avoidance.
- Congestion of the network is inferred when the loss ratio of reports exceeds some threshold.
- Upon congestion, divide `cwnd` by 2, and restart its increase.

Protocol in brief: Error Control

- The protocol is reliable in the sense that it ensures that all clients have sent their reports.
- To reduce the duration of the session:
  - In the first round, the protocol probes clients to whom a request has not been yet sent (no retransmission of requests).
  - In the second round, the protocol probes clients whose reports were lost in the first round.
  - In the third round, the protocol probes clients whose reports were lost in the first two rounds.
  - Continues in rounds until all reports are received.

Measuring the loss ratio

- The source disposes of a timer, called TO:
  - The timer is set to SRTT + 4 RTTVAR, where SRTT is the average round-trip time, and RTTVAR its mean deviation.
  - The timer is rescheduled every time it expires.
  - The value of the timer can be seen as an upper bound on RTT.
- The timer serves to measure the loss rate.
  - All reports sent during one cycle of the timer have to arrive during the next cycle at the latest, otherwise they are supposed lost.

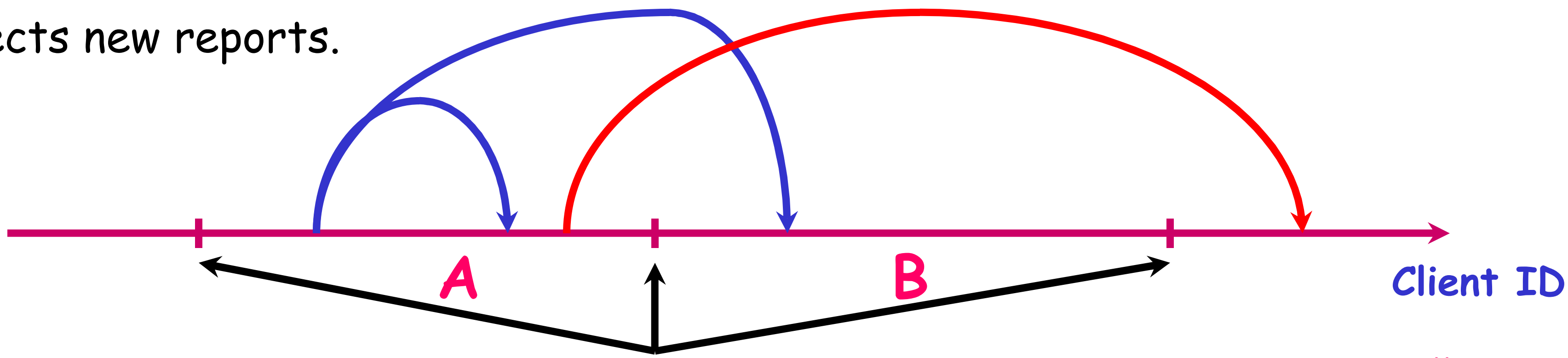
Protocol in graphics

Reports received before their deadline:

- Increase the congestion window.
- Injects new reports.

Reports received after their deadline:

- Do nothing, only take the information.



Moments of expiration of the timer:

- Measure the loss ratio.
- Given the loss ratio, conclude whether to keep the window unchanged, to divide it by 2 (--> CA), or to reset it (-->SS).
- Reschedule the timer.
- Decide that reports not received before their deadlines are lost, and inject new requests into the network (if the window allows).

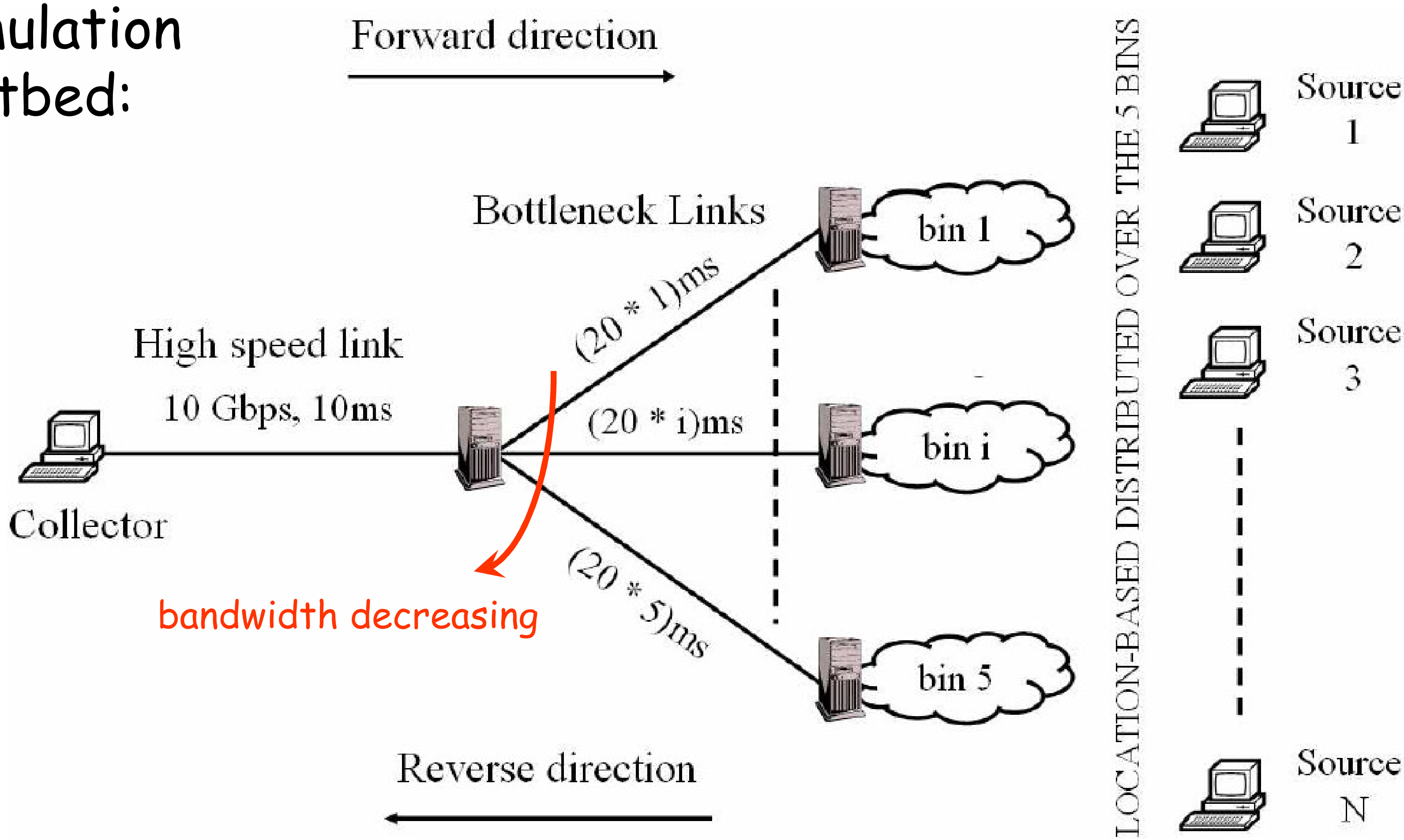
To receive in B all reports whose requests were sent in A.

Clustering of clients

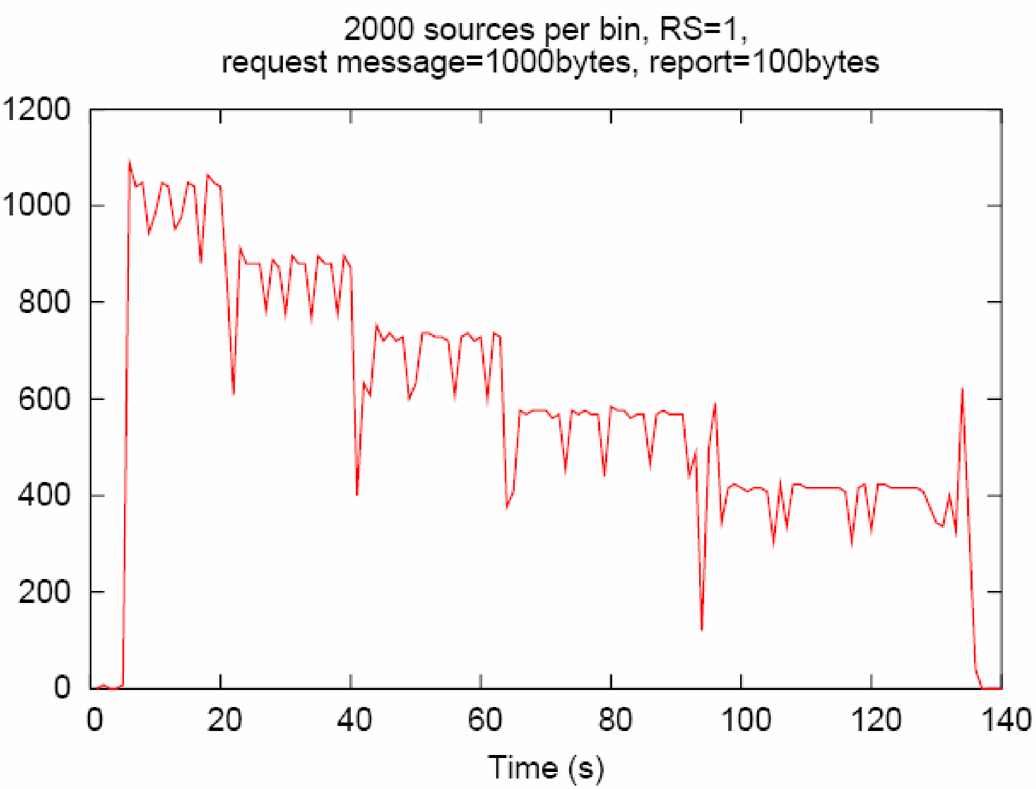
- For the congestion control to be effective, it is important to probe clients located behind the same bottleneck, before switching to clients located behind another bottleneck, and so on.
- We propose to use one of the existing methods for the clustering of hosts in the Internet:
  - Landmarks.
  - Decentralized coordinate systems.
  - Domain names.
  - Autonomous systems.
  - BGP update messages.

Validation by simulation

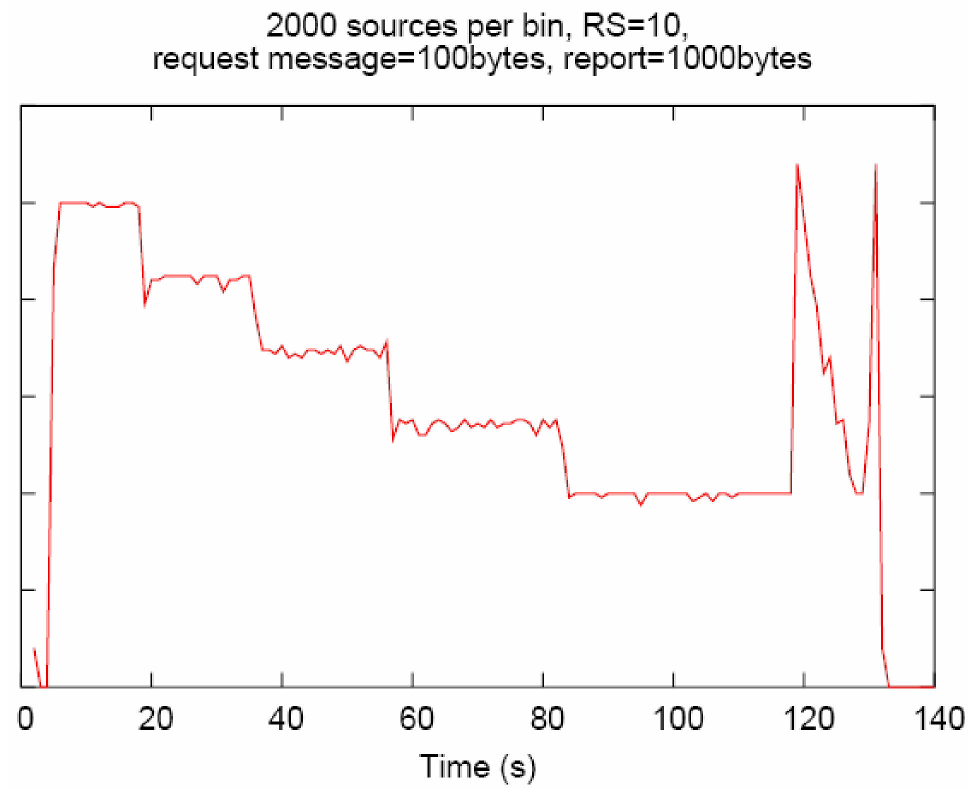
- We implement the protocol in ns-2.
- Simulation testbed:



Without competing TCP traffic

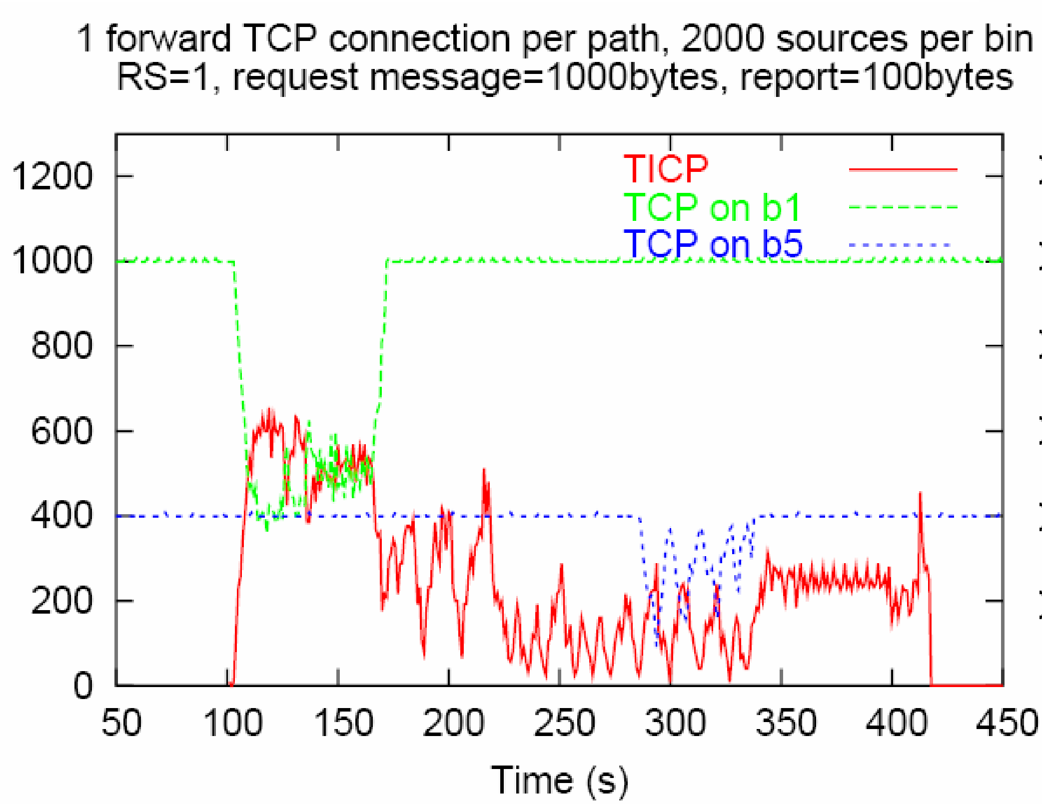


Throughput of requests (kbps)
Congestion in the forward path

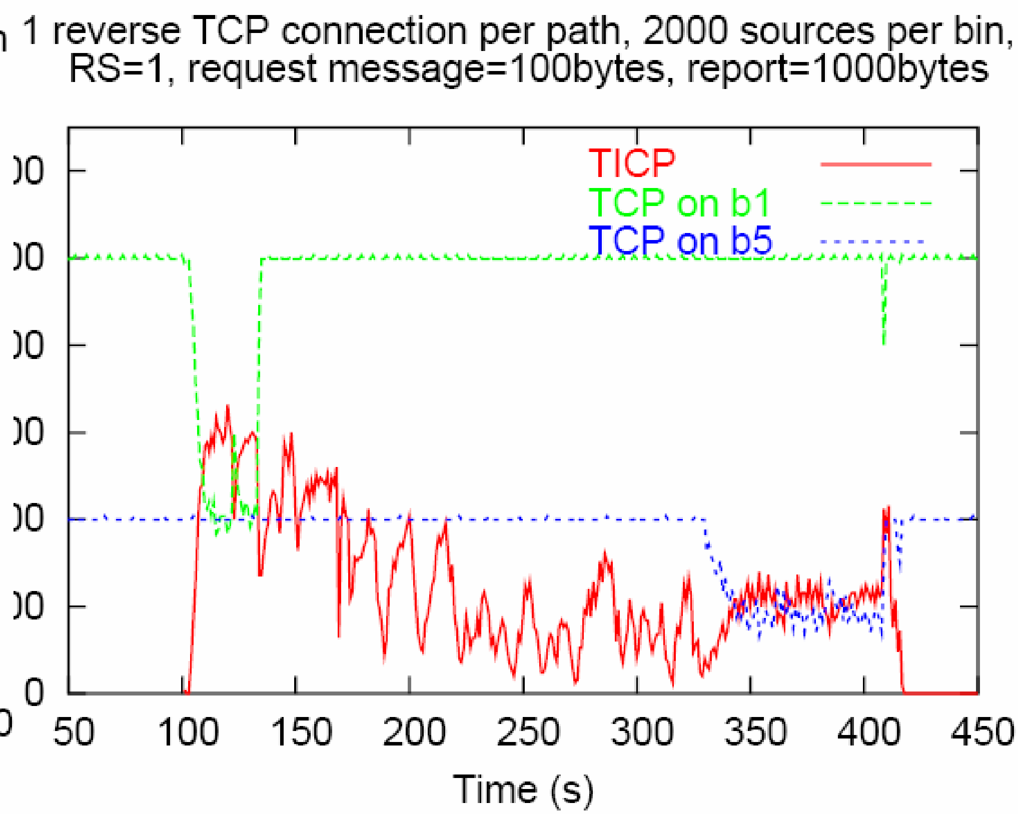


Throughput of reports (kbps)
Congestion in the reverse path

With competing TCP traffic



Throughput of requests (kbps)
Congestion in the forward path



Throughput of reports (kbps)
Congestion in the reverse path

The same throughput as TCP can be obtained if the protocol parameters are chosen equivalent to their counterparts in TCP.

Conclusions, Perspectives

- A protocol to collect identical information by probabilistic probing.
  - We focus on the sum of information i.e., Number of active clients
  - More functions can be done as well: mean, std, distribution, etc.
  - Also needed a mechanism to set the probing probability as a function of network conditions.
- A protocol for entire and reliable collection of information.
  - To be implemented and tested in reality.
- Can we relax the multicast assumption in the forward direction? P2P ?
- Can a dialogue between clients improve the collection?

Selected Publications

- Chadi Barakat, Mohammad Malli, Naomichi Nonaka, "TICP: Transport Information Collection Protocol", to appear in Annals of Telecommunications. INRIA Research Report 4807.
- Sara Alouf, Eitan Altman, Chadi Barakat, Philippe Nain, " Optimal Estimation of Multicast Membership", IEEE Transactions on Signal Processing - Special Issue on Signal Processing in Networking, vol. 51, no. 8, pp. 2165-2176, August 2003.
- Sara Alouf, Eitan Altman, Chadi Barakat, Philippe Nain, "Estimating Membership in a Multicast Session", in proceedings of ACM SIGMETRICS, San Diego, CA, June 2003.