FlightPath: Obedience vs. Choice in Cooperative Services

Authors: Harry C. Li, Allen Clement, Mirco Marchetti, Manos Kapritsos, Luke Robison, Lorenzo Alvisi, and Mike Dahlin

Presentor: Yinzhi Cao

Problems in P2P System

- Byzantine Peers

  They want to disrupt the service.

- Selfish Peers

  They may use the service without contributing their fair share.

Existing Work

- Works that use incentives information to argue that rational peers will obey a protocol, like KaZaA and BitTorrent.
  - Drawbacks: Some users may gain better service quality when cheating
- Works emphasizes rigor by using game theory (like Nash equilibrium) to design a protocol's incentives and punishments.
  - Drawbacks: Do not allow dynamic membership, waste network bandwidth to send garbage data

This Work: FlightPath

- High Quality Streaming
  - Good Service to Every Peer
- Broad Deplorability
  - Peak Upload Bandwidth is limited to ADSL Bandwidth
- Rational-tolerant
  - 1/10-Nash Equilibrium
- Byzantine-tolerant
  - Works well when 10% of peers act maliciously
- Churn-resilient
  - Has good performance when 30% peers churn every min

Nash Equilibrium

$$\forall x = (x_1, x_2, \dots, x_n) \in S = S_1 \times S_2 \times \dots \times S_n (S_i \text{ is a strategy set})$$

$$\exists f = \big(f_1(x), f_2(x), \dots, f_n(x)\big) \text{ is a payoff function of } x.$$

$$\text{We can define a Nash Equilibrium Point } x \text{ as}$$

$$\exists\ x = (x_1, x_2, \dots, x_n)\ s.t.\ \forall i\ \forall x_i^* \neq x_i\ f_i(x_i^*) \geq f_i(x_i)$$

$$\text{Also, we can define a } \varepsilon - \text{Nash Equilibrium Point } x \text{ as}$$

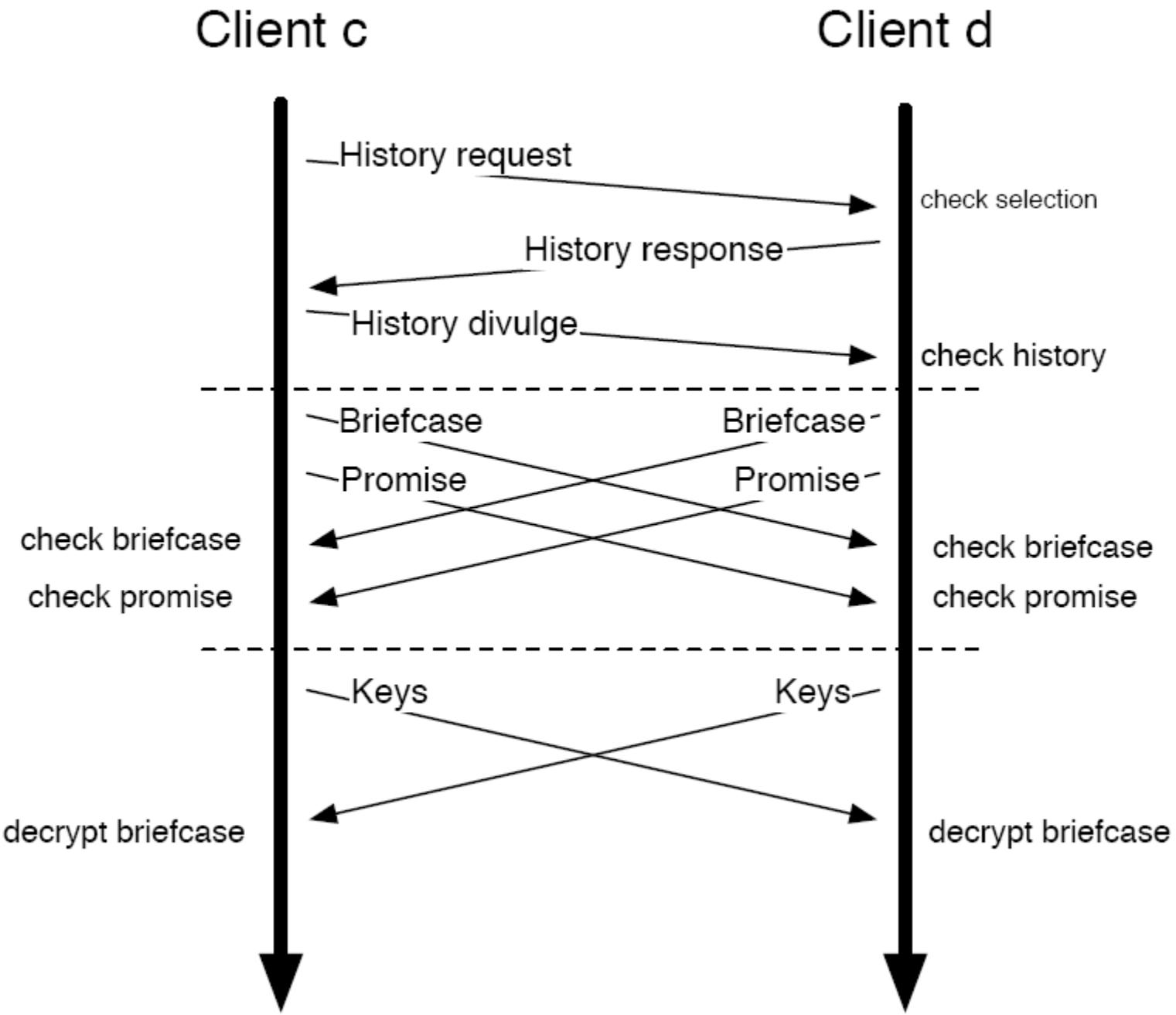
$$\exists\ x = (x_1, x_2, \dots, x_n)\ s.t.\ \forall i\ \forall x_i^* \neq x_i\ f_i(x_i^*) \geq (1-\varepsilon) f_i(x_i)$$

Model: BAR Model

- Time is divided into rounds that are r_{len} seconds long.
- In each round, the source generates num_ups unique stream packets that expire after deadline round.
- All peers work together to distribute packages before deadline.

How can One Node Work?

- We call one node's work during a round a trade.
- A trade has four phases.
  - Partner Selection
  - History Exchange
  - Update Exchange
  - Key Exchange


Client c
Client d
History request
check selection
History response
History divulge
check history
Briefcase
Briefcase
Promise
Promise
check briefcase
check briefcase
check promise
check promise
Keys
Keys
decrypt briefcase
decrypt briefcase

Taming Gossip(How to improve previous approach)

- Reservations
- Splitting Need
- Erasure Codes
- Tail Inversion
- Imbalance Ratio
- Trouble Detector

Reservations

- We partition n peers into $\lfloor \log n \rfloor$ bins and require a peer to choose a partner from a verifiable pseudo randomly chosen bin.
- Within a bin, we restrict nodes that a peer can communicate based on its id. A peer can only communicate nodes that the hash of its id and the other's id is less than some p.

$$\left[1-\left[1-p(1-F_{byz})\right]^{\frac{n}{\lceil \log n \rceil}}\right]^{\lceil \log n \rceil} \geq 1-\frac{1}{n}$$

Reservation Cont’d

- A node should make a reservation before it establish the connection.
- Peer d accepts a reservation only if it has not already accepted another reservation for the same round. Otherwise it rejects it.
- A node can indicate it has few options left in order to let others to accept it.

Splitting Need

- A peer splits its need into several parts and sends its needs to different nodes.
- This approach can reduce the possibility that a node receive duplicate package.

Erasure Codes

- n **erasure code** transforms a message of *n* blocks into a message with more than *n* blocks, such that the original message can be recovered from a subset of those blocks.
- We use erasure code here in order to evade Byzantine participants which may receive tracker's packages but not distribute them.

Tail Inversion

- A older package has a higher priority than new one.
- The reason is older one may be near the deadline.

Imbalance Ratio

- Imbalance Ratio a means a node can download N traffic but only upload aN traffic.
- According to the authors, a = 10% is a good tradeoff

Trouble Detector

- A node which observe itself has a bad performance may initiate more than one trade during a round.

Flexibility for Churn

- Epochs
  - An epoch is defined as e_{len} rounds. At the boundary between epochs e and e+1, the tracker shuffles membership list for epoch e+2 so that new members can join in this P2P system.
- Tub Algorithm
  - We classify peers into tubs based on their come-in time.

Tub Algorithm

- A node in a tub should obey the following three constraints.
  - Peer *d is in c's view only if d precedes c in the list.*
  - If *d is in tub t or t − 1, then d is in c's view if* the hash of concatenating *c's member id with d's* member id is less than *p.*
  - If *d is in a tub* $t' < t - 1$*, then d is in c's view if* the hash of concatenating *c's member id and d's* member id is less than a parameter p'.

Equilibrium Analysis(1)

- We define *u = (1 − j)β − wκ as utility function.*

  j is the average number of jitter events per minute

  β is the benefit from watching a jitter-free stream

  w is the average upload bandwidth used in Kbps

  κ is the cost per Kbps.

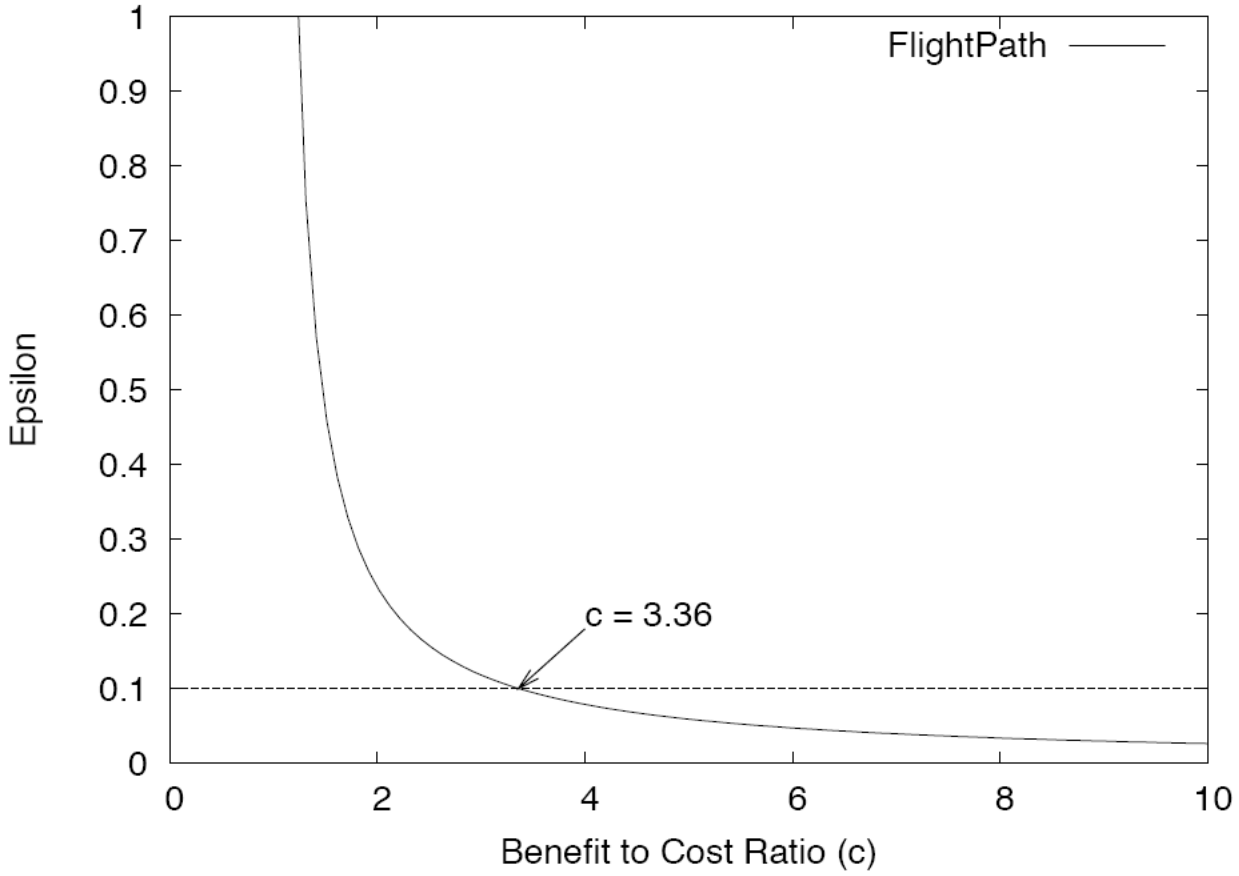
Equilibrium Analysis(2)

$$\varepsilon = \frac{u_o - u_e}{u_e} = \frac{(j_e - j_o)\beta - (w_o - w_e)\kappa}{(1 - j_e)\beta - w_e\kappa}$$

$$let\ \frac{w_0}{w_e} = b, \frac{\beta(1 - j_e)}{w_e k} = c, j_0 = 0$$

$$\varepsilon = \frac{\frac{c j_e}{1 - j_e} + (1 - b)}{c - 1}$$

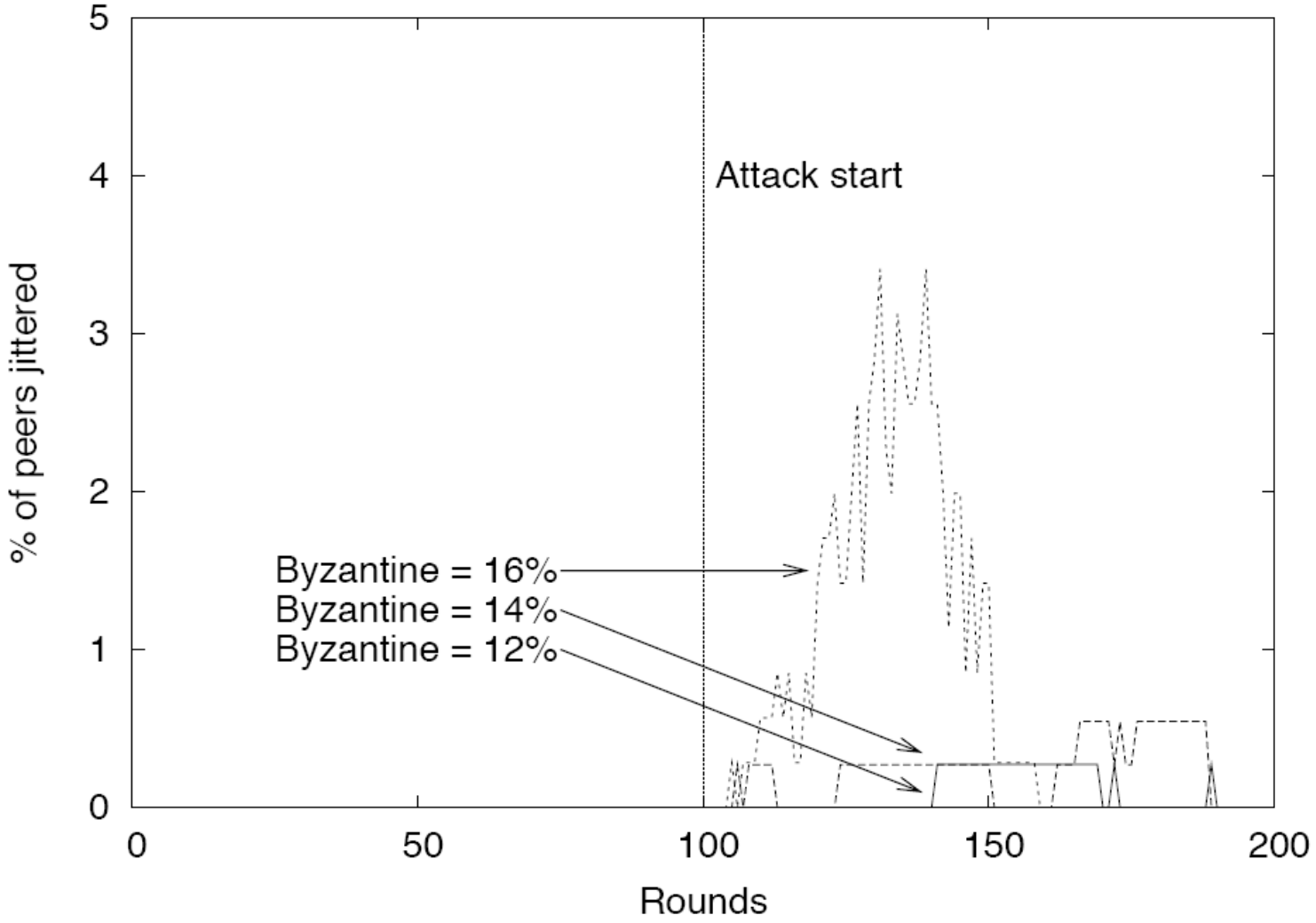
Equilibrium Analysis(3)



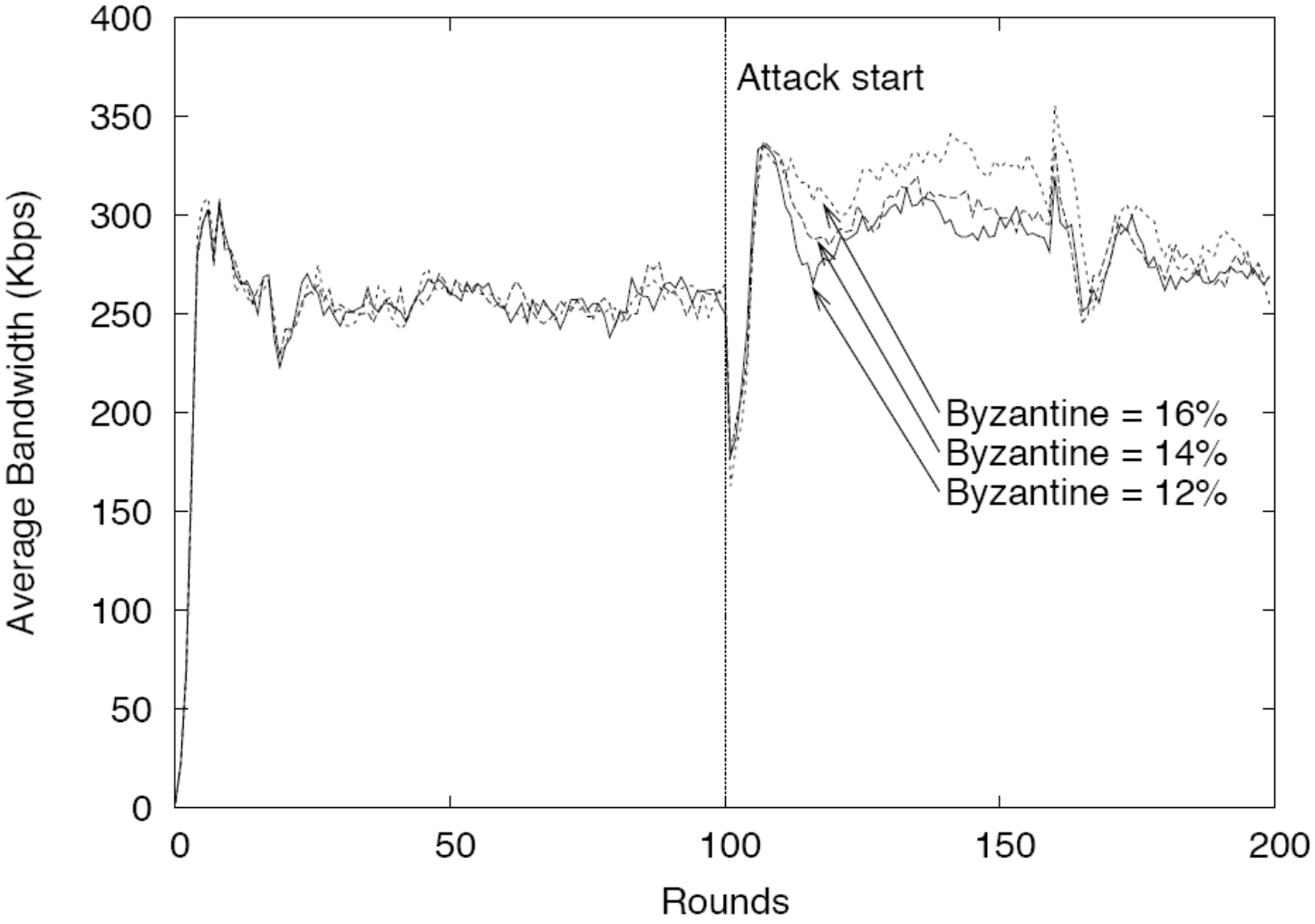
Evaluation

- *Reduces jitter by several orders of magnitude compared* to BAR Gossip
- *Caps peak bandwidth usage to within the constraints* of a cable or ADSL connection
- *Maintains low jitter and efficiently uses and width* despite flash crowds
- *Recovers quickly from sudden peer departures*
- *Continues to deliver a steady stream despite churn*
- *Tolerates up to 10% of peers acting maliciously(**Example**)*

Tolerates up to 10% of peers acting maliciously



Tolerates up to 10% of peers acting maliciously



Summary

- Merits
  - Improvement on Previous Work
- Drawbacks
  - Its 1/10 Nash Equilibrium is based on some parameter's value. But it fails to prove that value is achievable. Also it fails to prove its utility function can represent users' motivation.