Random Peer Sampling

Christian Grothoff

Technische Universität München

March 30, 2014

Why Random Peer Sampling?

Why Random Peer Sampling?

- Needed for gossip protocols, e.g. for multicast
- Needed for anonymity protocols (Tor)
- Needed for construction of unstructured overlays:
  random links provide robustness, expansion
- Useful for statistics (get information from random peer)

Environment for Random Peer Selection

- Structure of the topology?
- Support for churn?
- Working set size (for large network)?
- Adversarial behavior?

Adversarial Goals

- Isolate nodes
- DoS
- Promote themselves
- Bias statistics

Adversarial Capabilities

- Controls fraction f of the nodes
- Byzantine failure

System Model

- Each peer has a view V with a set of IDs it knows
- V needs to be "small"
- V evolves in (unsynchronized) rounds

PUSH Gossip

- Peer sends his ID to random peer in V
- Receiving peer adds new ID to V
- Discard (random or oldest or ...) ID if $|V|$ gets too large

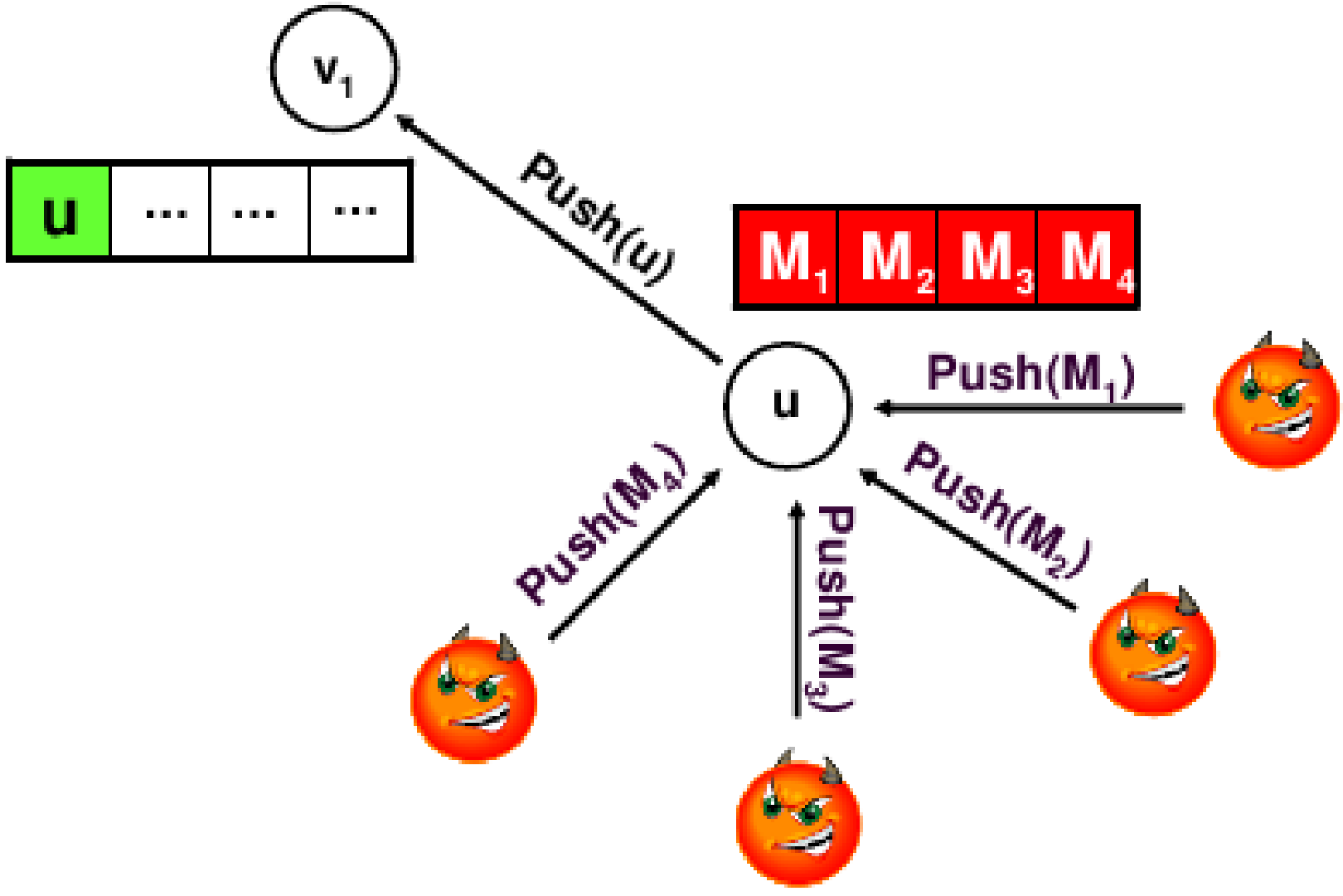
PULL Gossip

- Peer selects random peer p from his V, asks for V_p.
- Upon receipt, merge V_p into V

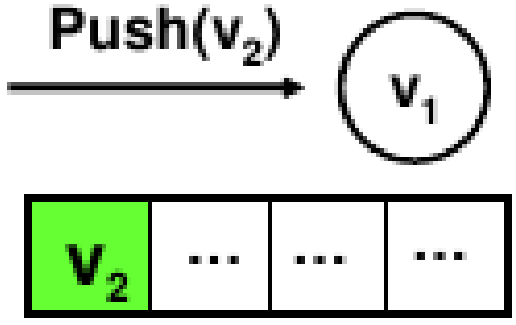
PUSH and PULL Gossip

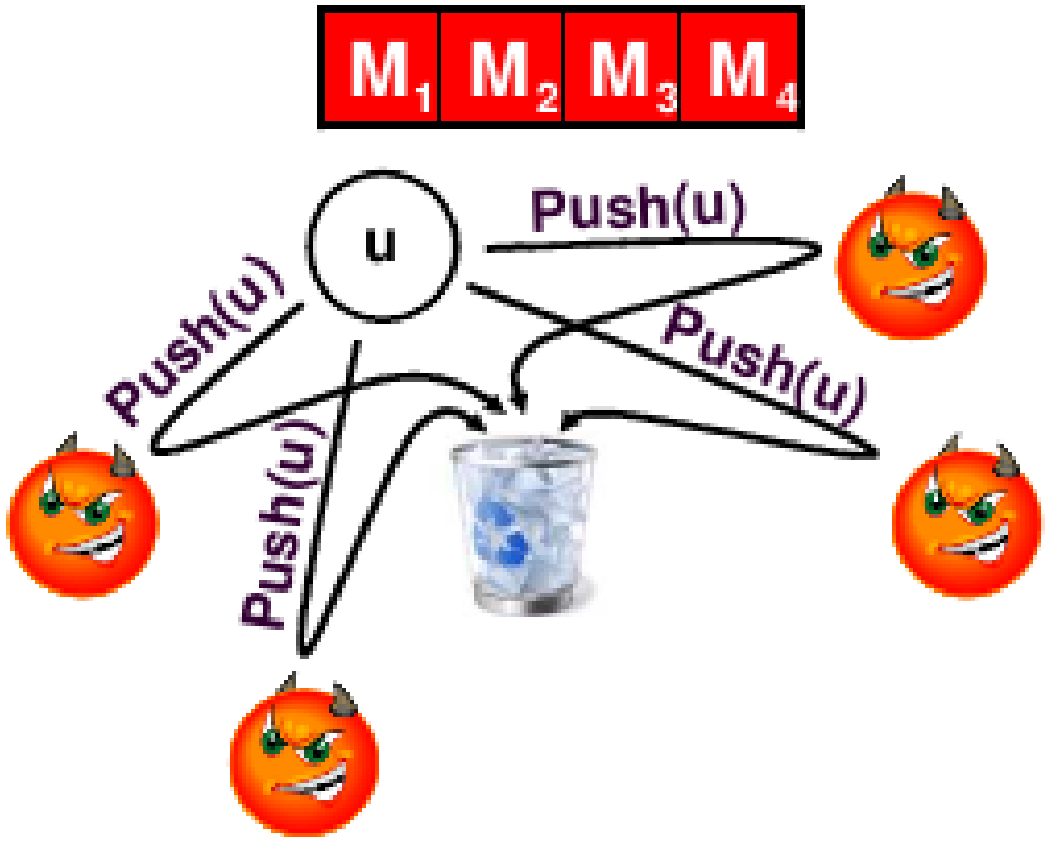
- Allavena [1] showed both must be combined
  otherwise: isolation, star topologies happen too frequently
- But: still not robust against Byzantine failure!
- Brahms [2] provides Byzantine fault-tolerant peer sampling!

PUSH Attack (1/2)



PUSH Attack (2/2)





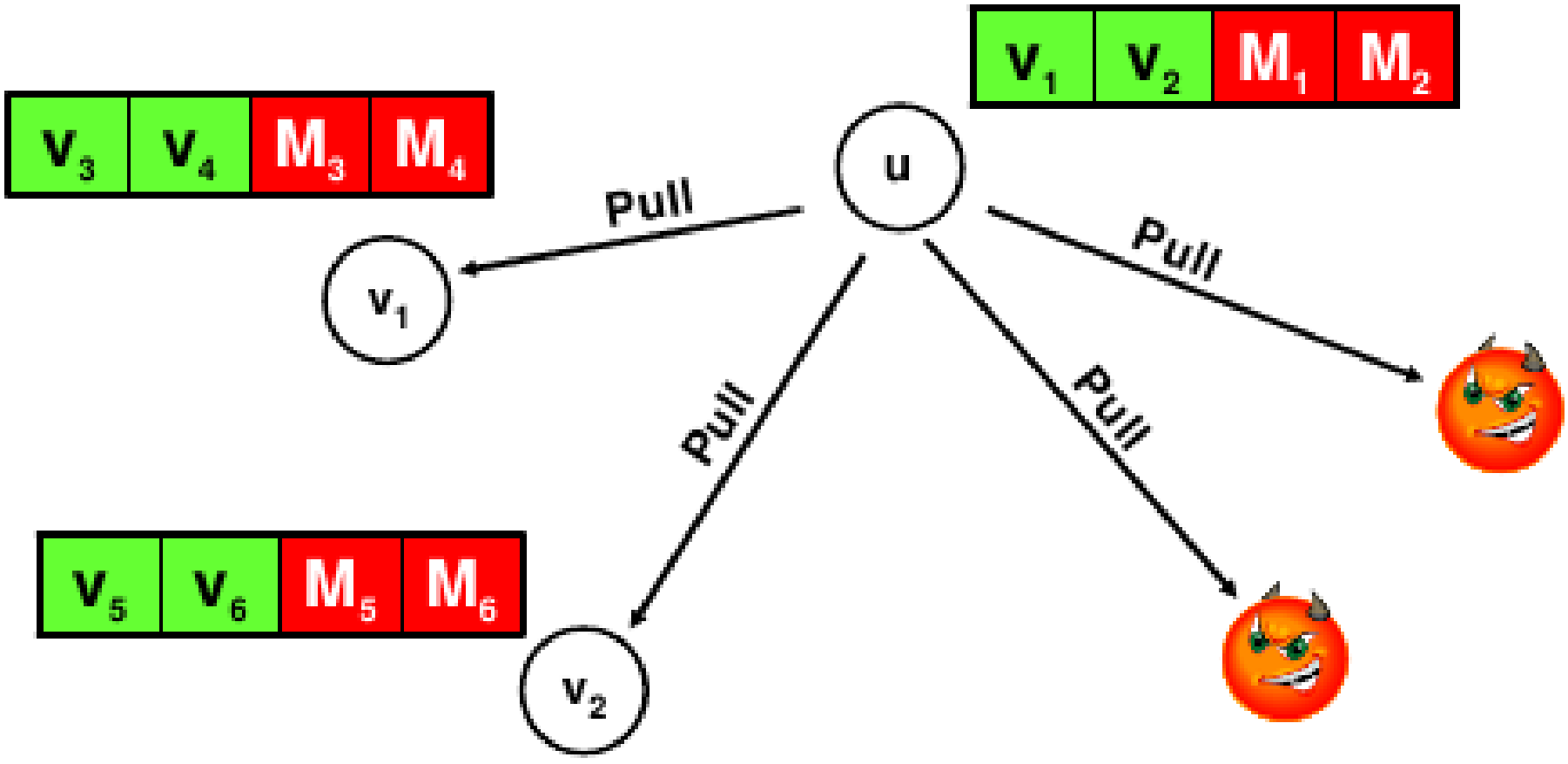
Brahm's PUSH Defense

- Limit rate at which nodes can PUSH
  e.g. using computational puzzles
- If more PUSHes are received than expected in a given time interval, ignore all of them

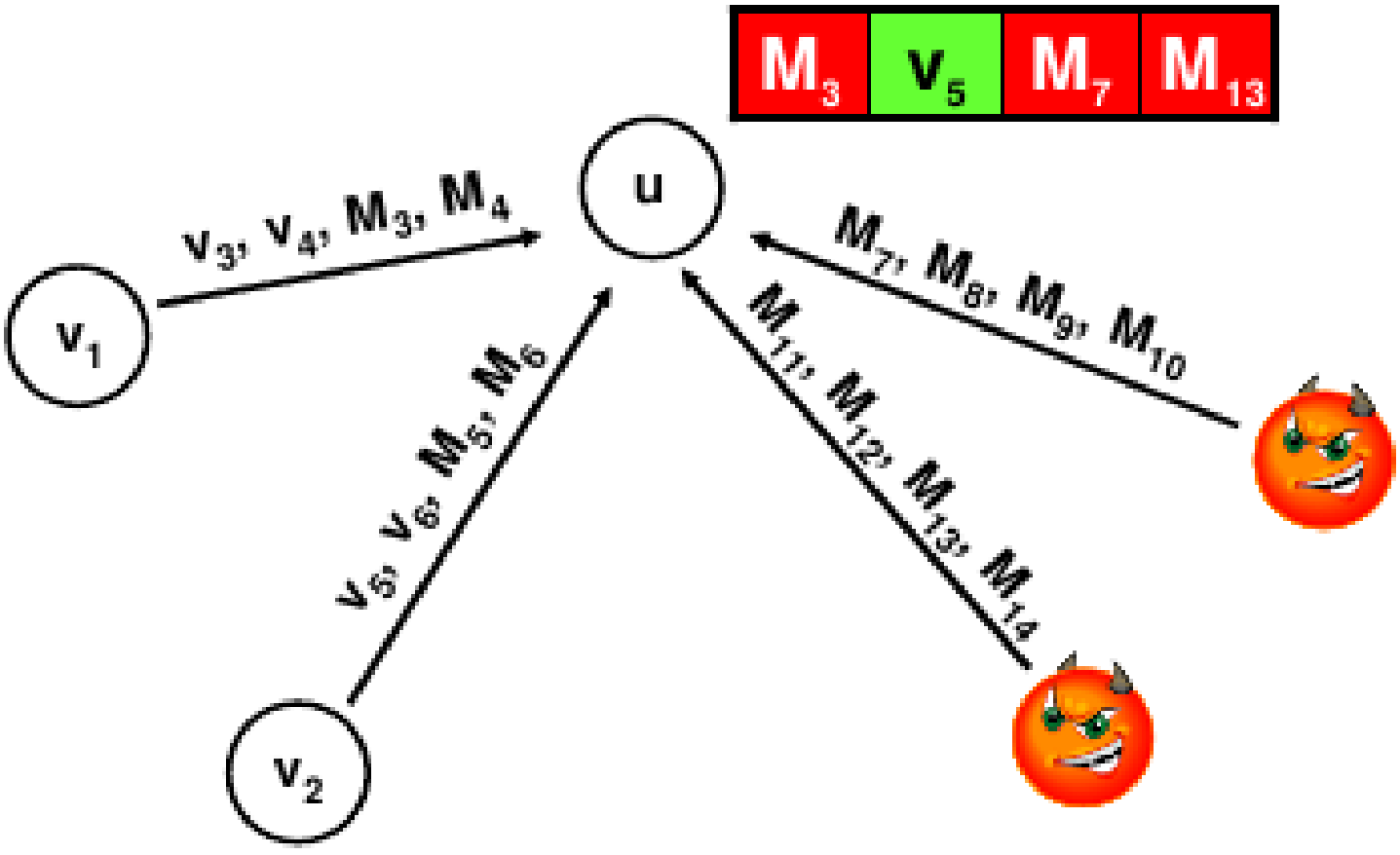
Brahm's PUSH Defense

- Limit rate at which nodes can PUSH
  e.g. using computational puzzles
- If more PUSHes are received than expected in a given time interval, ignore all of them
- This slows down the attack

PULL Attack (1/2)



PULL Attack (2/2)



Brahm's PULL Defense

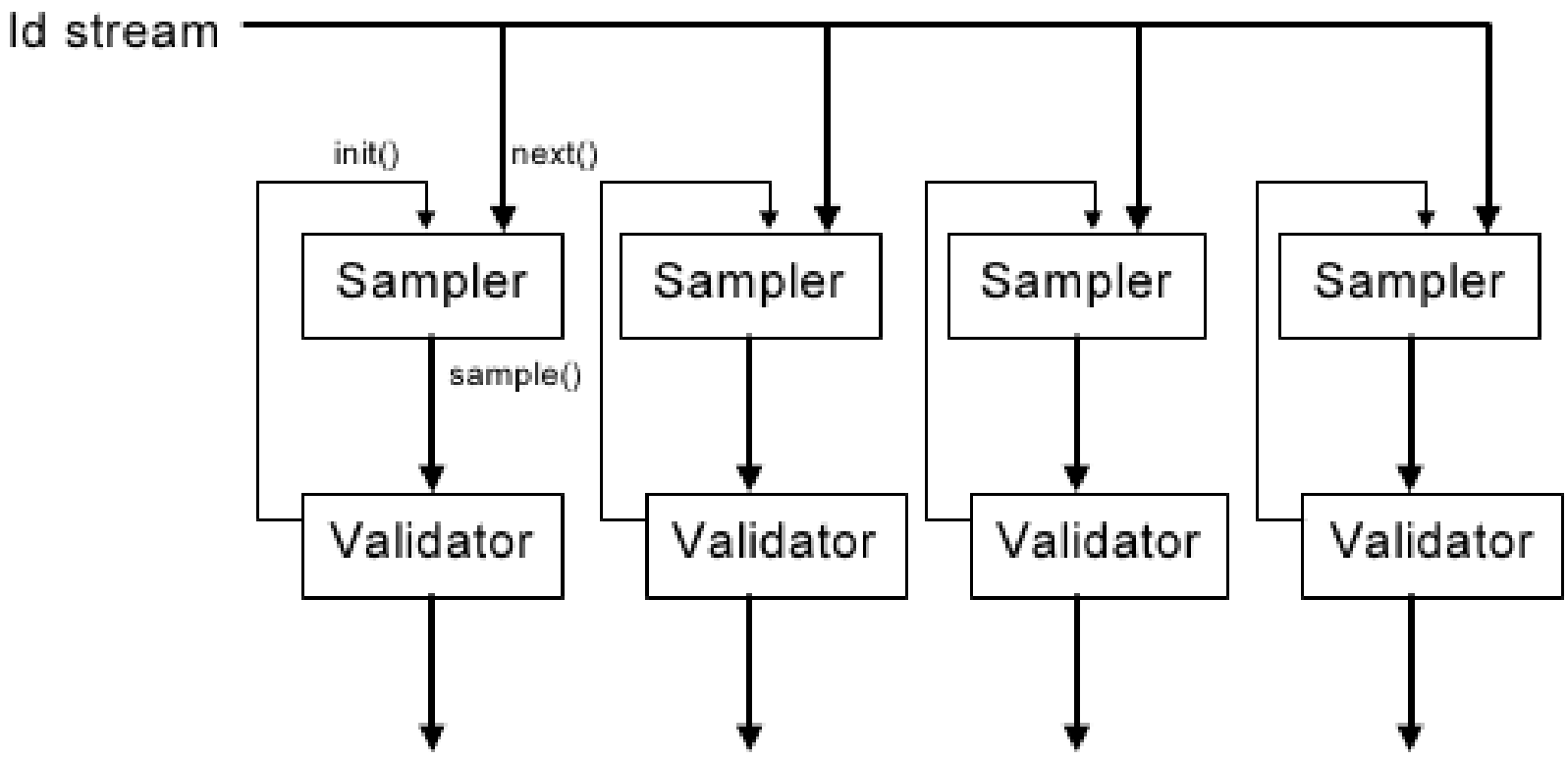
- Control contribution of PUSH IDs ($\alpha|V|$)
- Control contribution of PULL IDs ($\beta|V|$)
- Use history samples ($\gamma|V|$)
- $\alpha + \beta + \gamma = 1$.

$\Rightarrow$ If history contains non-faulty nodes, we win!

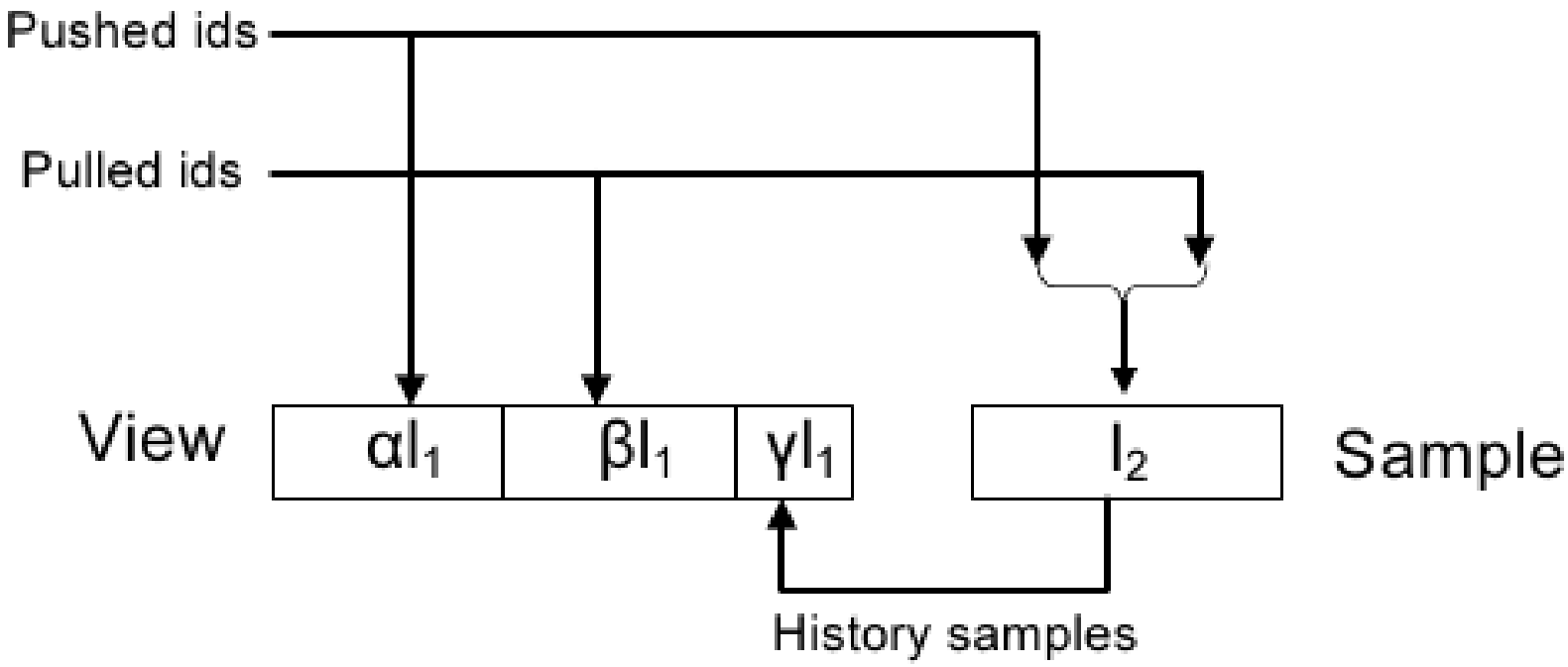
Sampler

```
1: function Sampler.init()
2:     h ← randomPRF(); q ← ⊥
3: function Sampler.next(elem)
4:     if q = ⊥ ∨ h(elem) < h(q) then
5:         q ← elem
6: function Sampler.sample()
7:     return q
```

Sampler and Validator



Brahms


Pushed ids
Pulled ids
View
αI1
βI1
γI1
I2
Sample
History samples

Parameter Choice

- $|V| = |S| = \Theta(\sqrt[3]{n})$
- $\alpha = \beta = 0.45$, $\gamma = 0.1$

$\Rightarrow$ time to partition $>$ time to convergence of 1st good sample

Conclusion and Future Work

- Byzantine fault tolerant random peer sampling is possible
- We need an upper bound on n to make it work (!)
- Choice of parameters is critical
- AFAIK Brahms has been simulated, but not implemented by P2P network

Questions?

References

Andre Allavena, Alan Demers, and John E. Hopcroft.

Correctness of a gossip based membership protocol.

In *PDOC'05*, pages 292–301. ACM, 2005.

Edward Bortnikov, Maxim Gurevich, Idit Keidar, Gabriel Kliot, and Alexander Shraer.

Brahms: Byzantine resilient random membership sampling.

In *PDOC'08*. ACM, August 2008.