TECHNISCHE UNIVERSITÄT DRESDEN

A Semantic Overlay for Self-* Peer-to-Peer Publish/Subscribe

Liu Tianhai

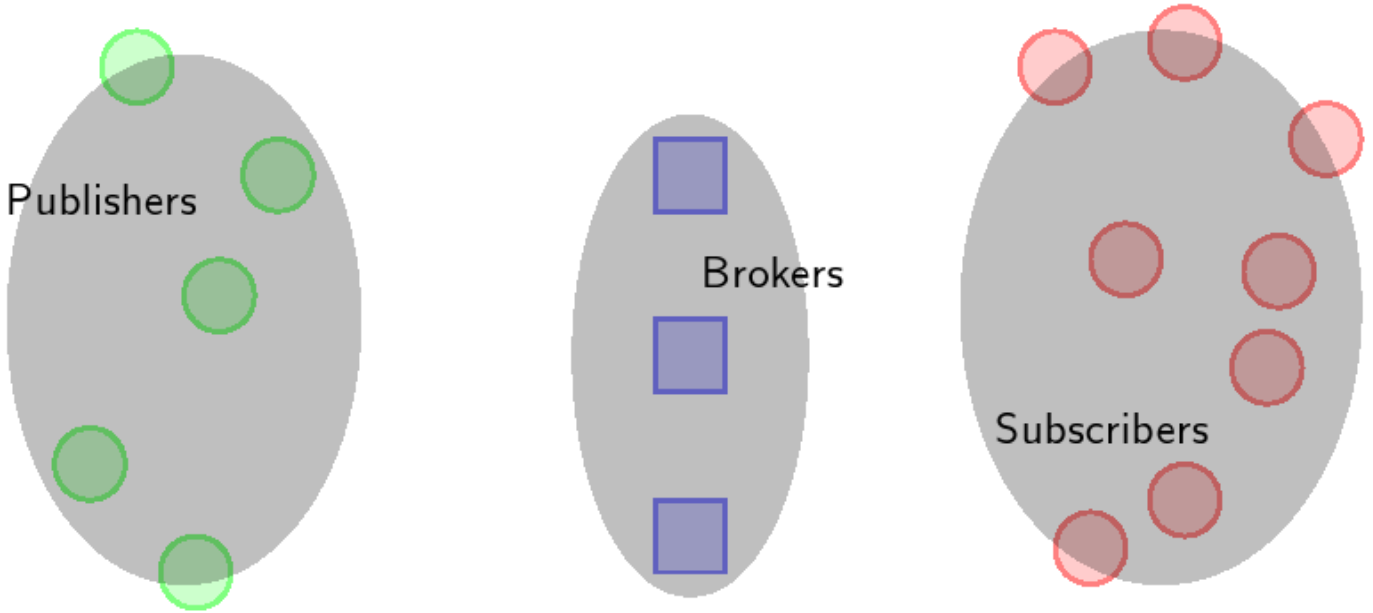
TECHNISCHE UNIVERSITÄT DRESDEN

Outline

- Introduction
- DPS Framework
- DPS overlay
- DPS Algorithms
- Evaluation of DPS
- Simulation of DPS
- Conclusion
- Question

Introduction

- Broker-based Pub/Sub (e.g. Siena)
  - rely on a network of dedicated servers (Brokers)
  - controlled by administrators
  - not good for dynamic decentralized scenarios (P2P)



Figue 1. Broker-based Pub/Sub. [2]

Introduction (cont.)

- Topic-based P2P Pub/Sub systems (e.g. Scribe)
  - self-* capabilities
  - using DHT (hard for content-based)
  - Rendezvous (single root)

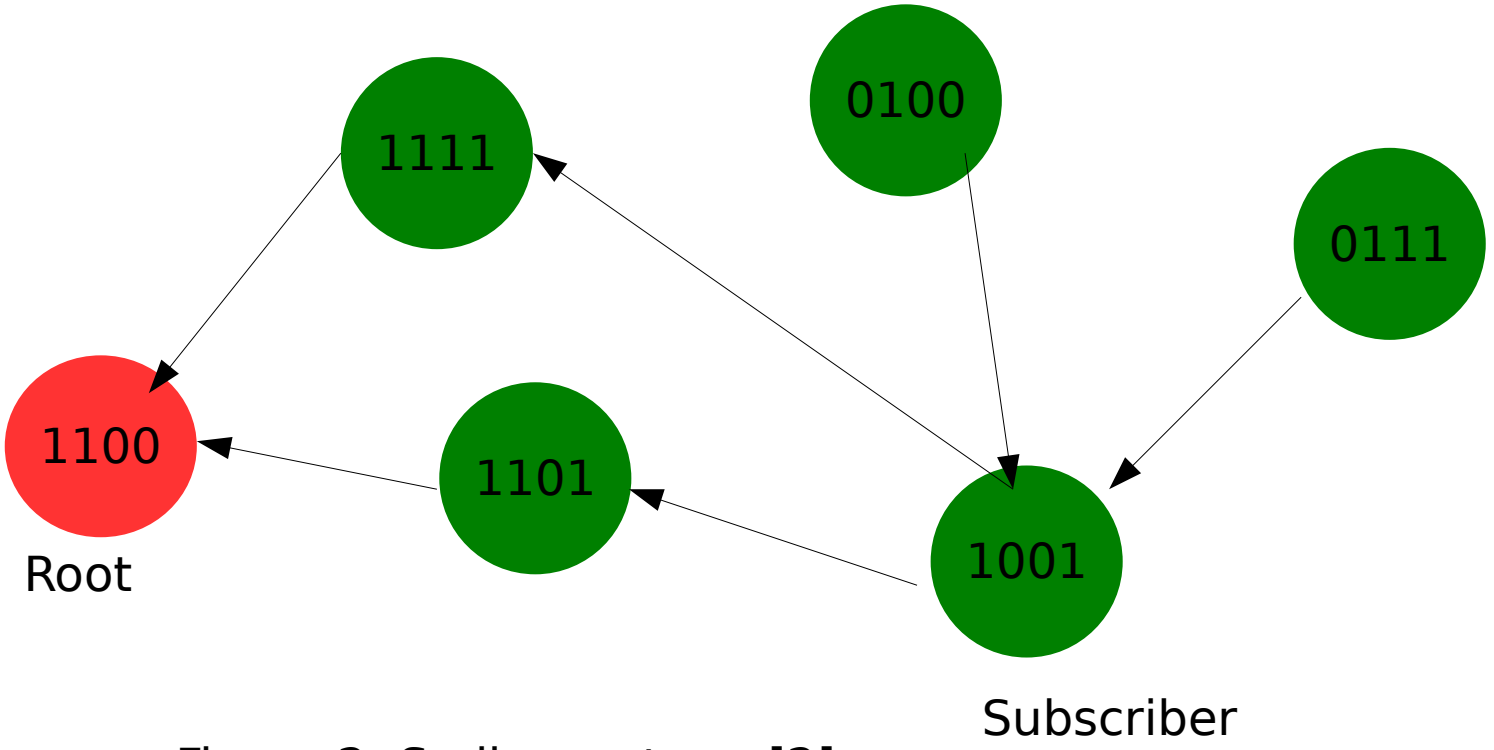


Figure 2. Scribe system. [2]

DPS

- Content-based P2P Pub/Sub systems (e.g. DPS)
- DPS (Dynamic Publish/Subscribe)
  - no brokers; peer-to-peer.
  - subscription-driven semantic overlay – self-organization
  - one tree structures one attribute – self-configuration
  - self-healing
  - subscription not replicated
  - just know neighboring groups

Framework

- Date Model
  - subscription (filter)
    - $F = AF_1 \wedge ... \wedge AF_j$
      - $AF_i = (name_i\ Op_i\ c_i)$ (e.g. $a < 5$)
  - Events
    - $E = AV_1 \wedge ... \wedge AV_k$
      - $AV_i = (name_i = v_i)$

DPS Overlay

- Terminology
  - similar nodes
    - share at least one common predicate
    - e.g. $p \bowtie_{AF} s$,
  - semantic group (simply group)
    - members of a group are similar nodes for a common AF
    - e.g. $\forall p, s \in G_{AF} : p \bowtie_{AF} s$
  - predicate inclusion relation + group predicates => group predecessor relation (e.g. $G_{a>2}$ is predecessor of $G_{a>3}$)

DPS Overlay (cont.)

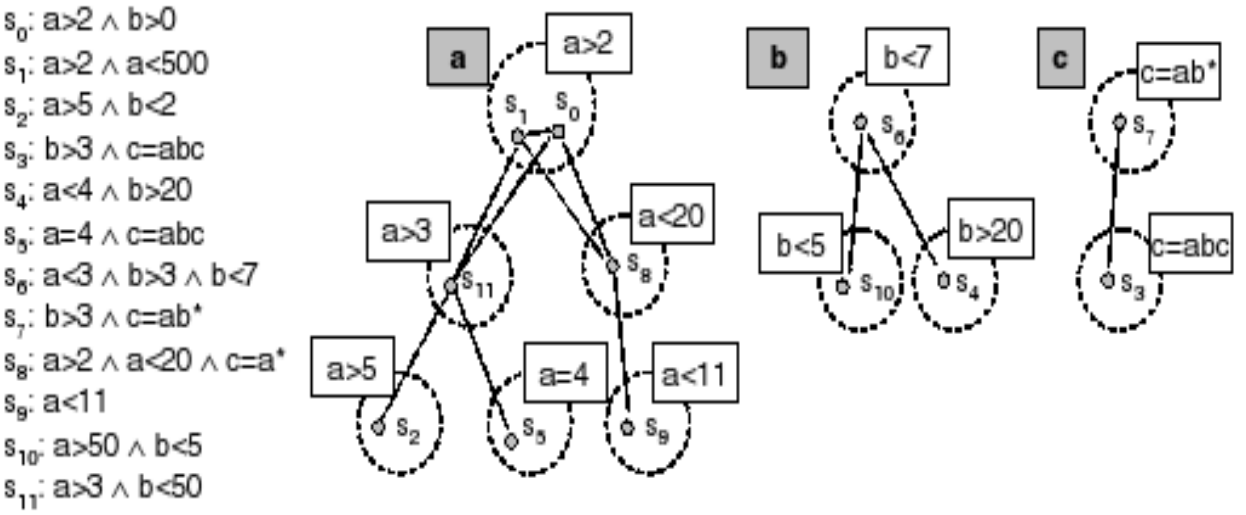


Figure 3. Logical Trees [1]

- each attr. owned by one subscriber
- trees connected to a logical forest
- new subscriber joins
- decompose subscription to attr.s
  - false positives
- ambiguity avoid
  - e.g. a = 4, when locate a group upon subscription
  - consistent convention: odd->greater-than branch; even...

Tree Traversal Alg.

- **Tree Traversal Alg.**
  - root-based approach
    - contact point: root
    - start from the root, traverse downwards
    - lower latency but hotspot
  - generic approach
    - contact point: any point
    - traverse both directions
    - load balanced but more messages

Tree Traversal Alg. (cont.)

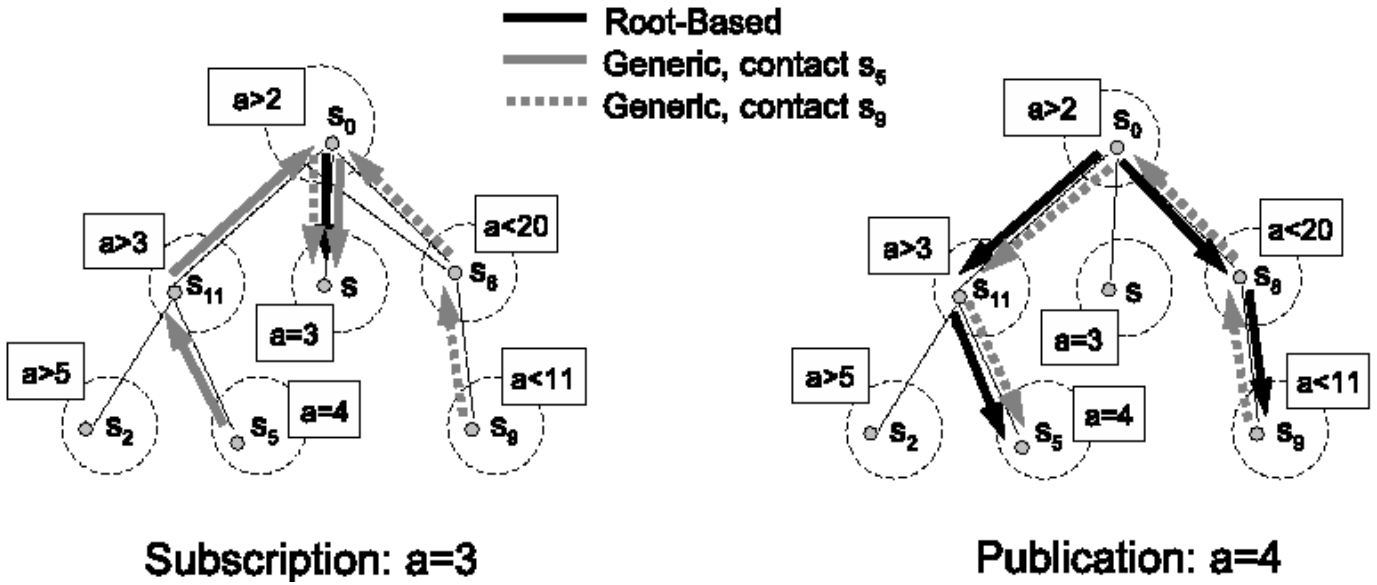


Figure 4. Tree Traversal Example. [1]

- subscription scheme
  - primitives: FIND_GROUP, SUBSCRIBE_TO and CREATE_GROUP

Tree Traversal Alg. (cont.)

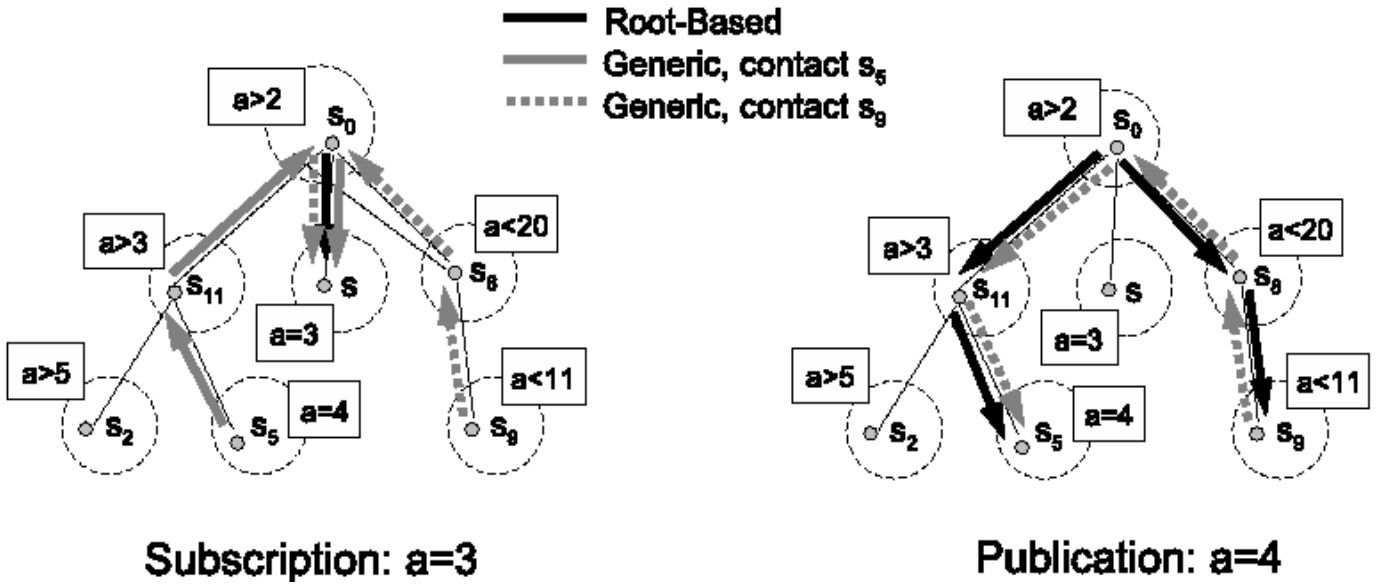


Figure 5. Tree Traversal Example. [1]

- publication scheme
  - primitives: PUBLISH_GROUP

Communication Alg.

- leader-based comm.
  - leader
  - co-leaders, for leader failure
  - subscription
    - primitives: CREATE_GROUP, SUBSCRIBE_TO
    - invoked by leader of predecessor group
  - publishing
    - primitive: PUBLISH_GROUP

Communication Alg. (cont.)

- epidemic comm.
  - each node communicates with a subset of other groups
  - message copied => fault-tolerance
  - periodic gossiping to maintain the data structures
  - subscription
    - primitives: CREATE_GROUP, SUBSCRIBE_TO
    - GOSSIP_SUB:
      - update and propagate views in the group
      - send to F_s other nodes with probability *p*
      - concurrent similar subscriptions avoid => *view_update* message
  - publishing
    - each node gossiping to k neighbors
    - forwarding stopped for *p*

TECHNISCHE UNIVERSITÄT DRESDEN

Evaluation of DPS

- complexity analysis
  - efficiency (probability of a event matching the filter)
    - root-based, generic
    - the key: position of contact point
  - message complexity (number of messages sent)
    - leader-based comm.
    - epidemic-based comm.
    - both the same: root-based less complexity

Simulation of DPS

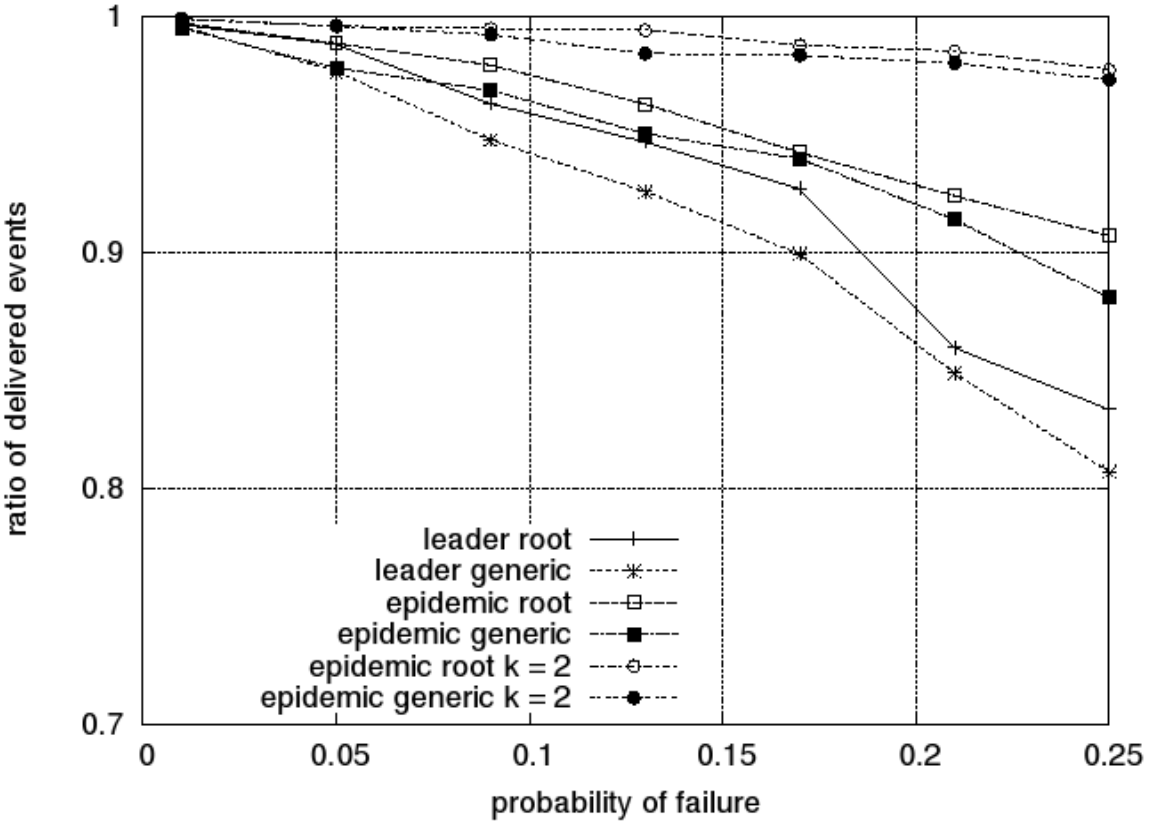


Figure 6. Dependability. [1]

- epidemic survives better

Simulation of DPS (cont.)

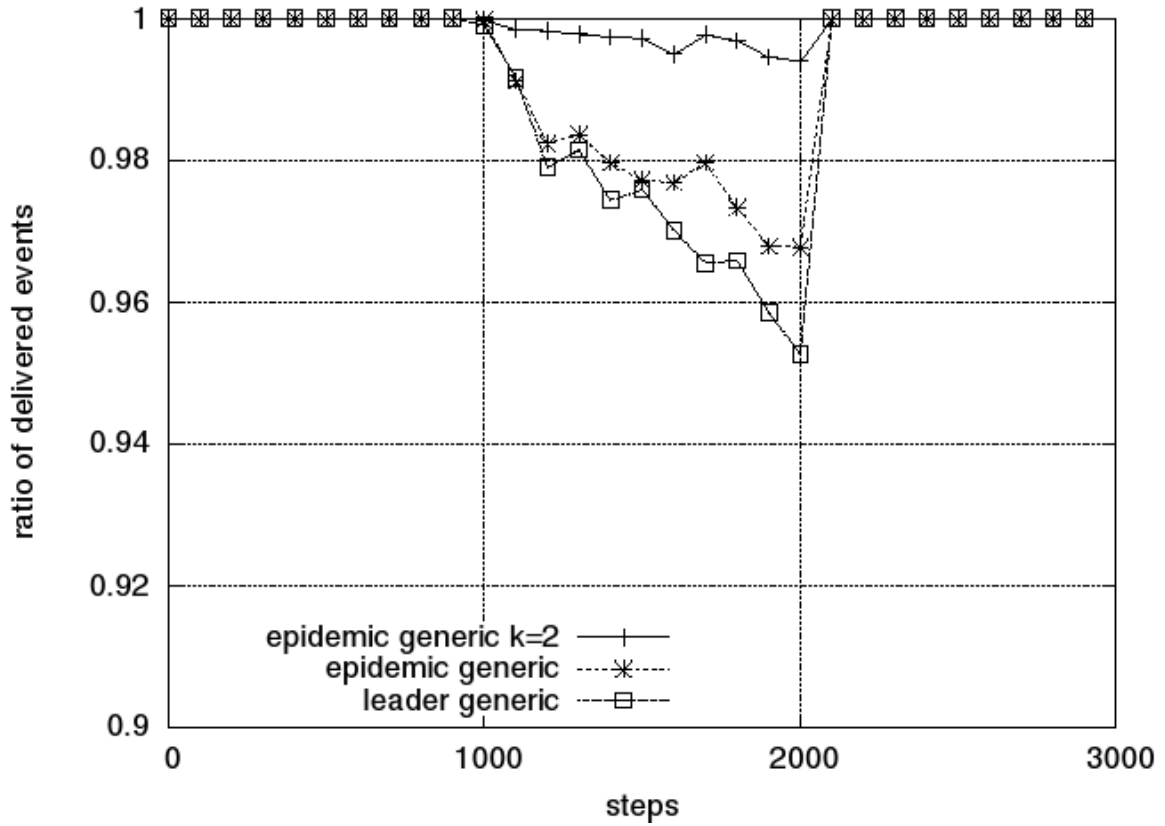


Figure 7. Recovering from failures (generic). [1]

- self-recovery of DPS

Simulation of DPS (cont.)

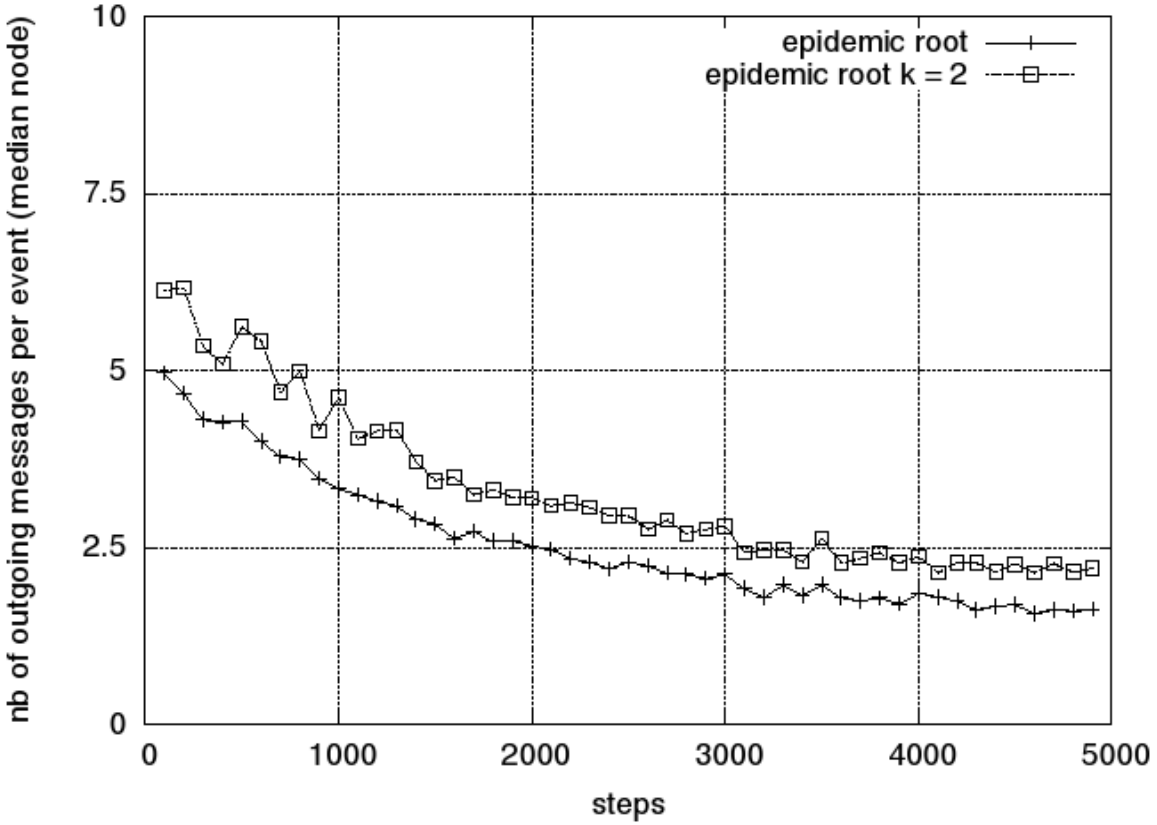


Figure 8. Scalability: outgoing messages per event (median). [1]

Simulation of DPS (cont.)

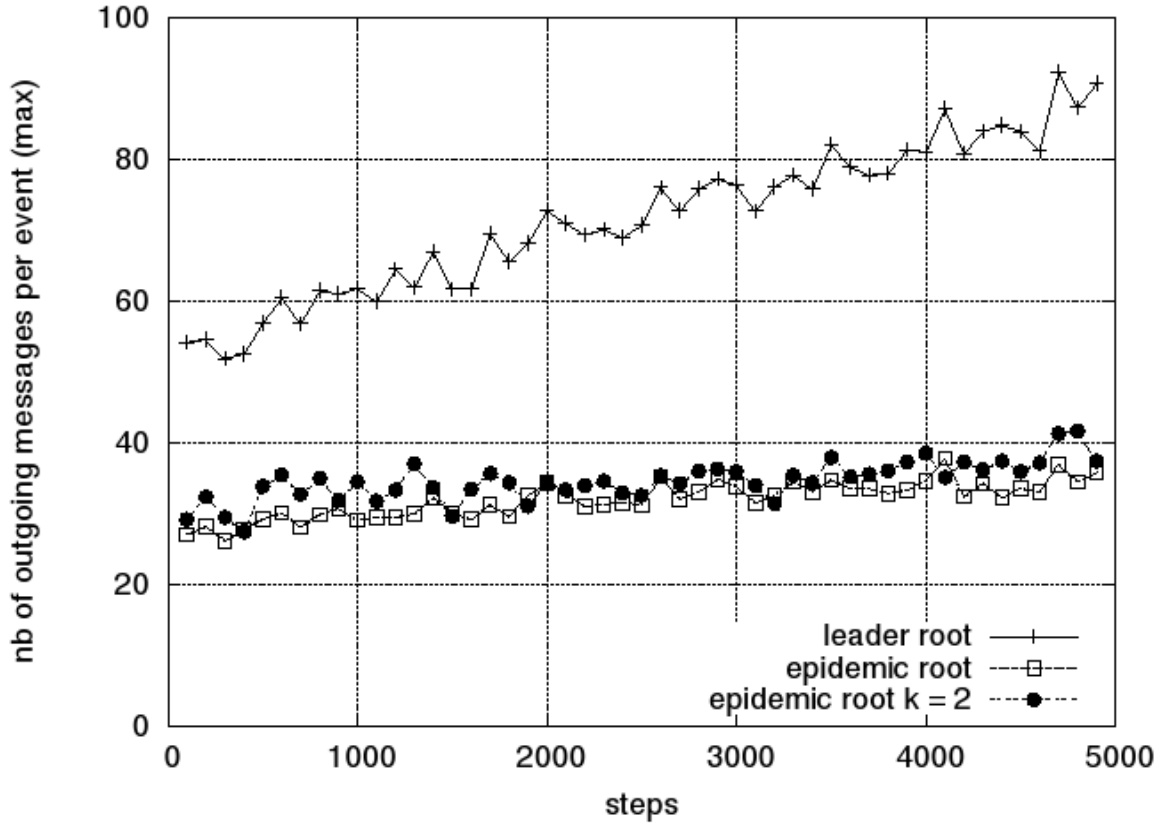


Figure 9. Scalability: outgoing messages per event (max). [1]

- epidemic more scalability

Simulation of DPS (cont.)

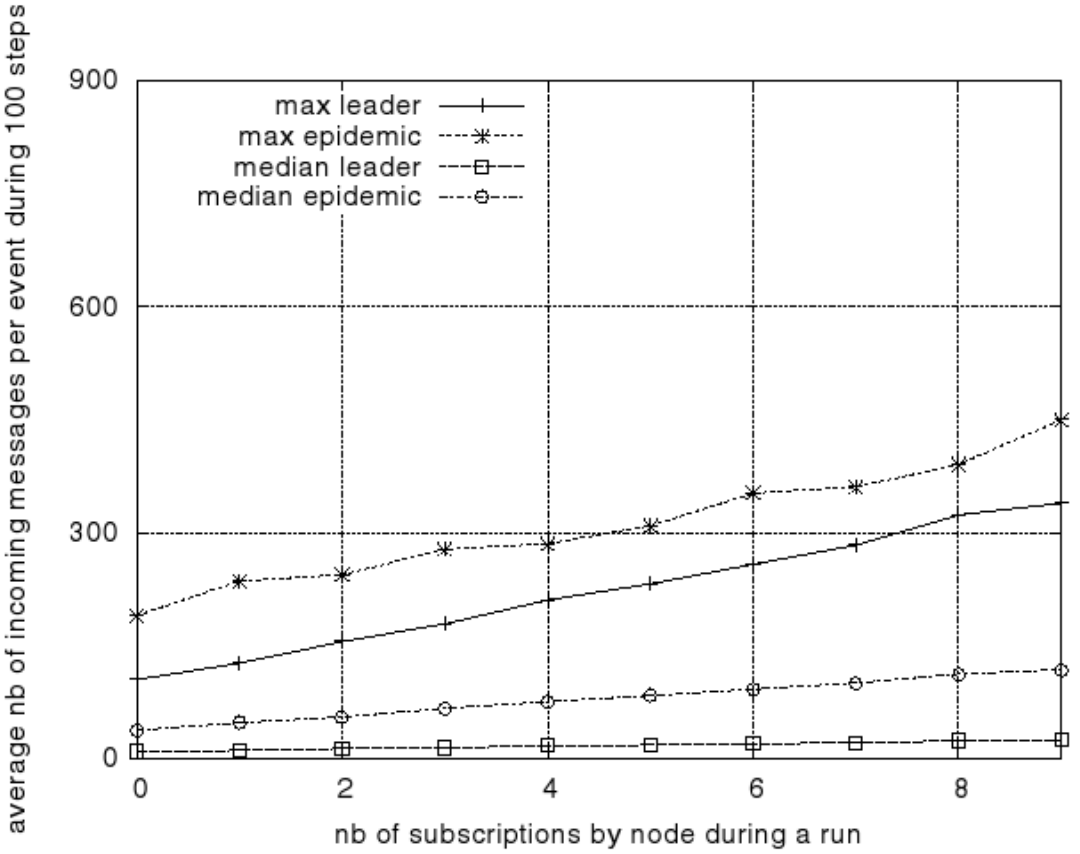


Figure . Leader vs. Epidemic Approaches: Received Messages. [1]

- epidemic receives more messages

Simulation of DPS (cont.)

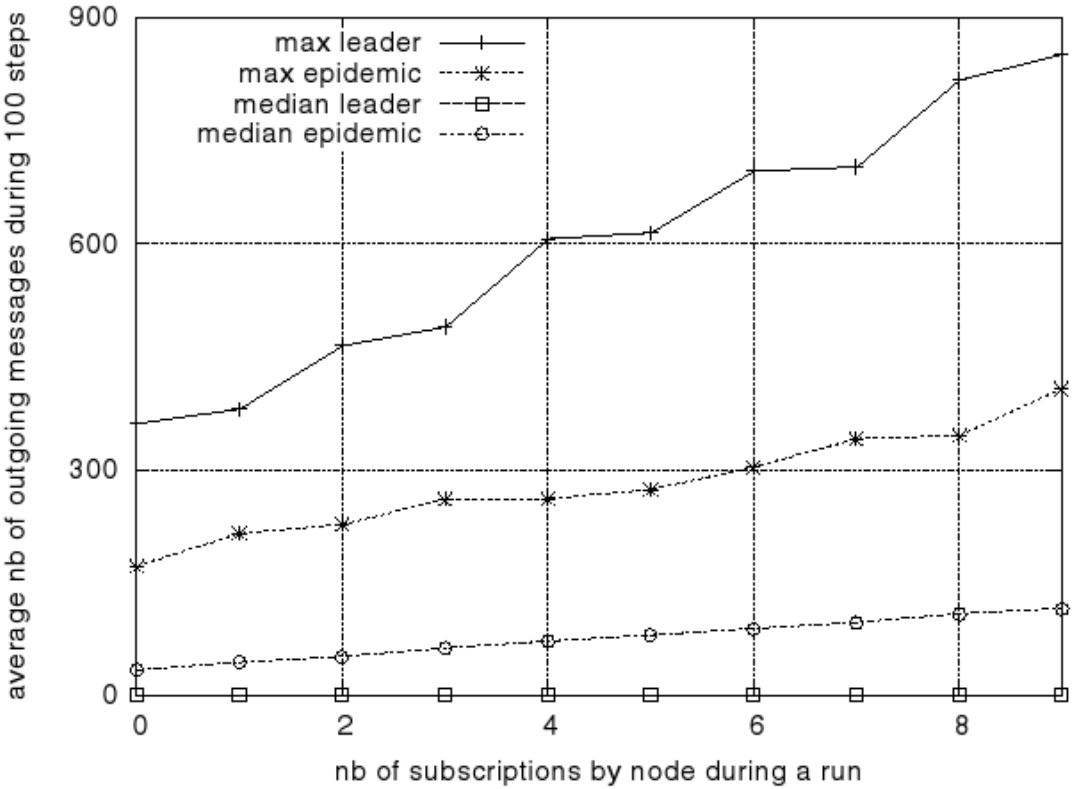


Figure 10. Leader vs. Epidemic Approaches: Sent Messages. [1]

- epidemic better load balance
- leader-based increases highly

Simulation of DPS (cont.)

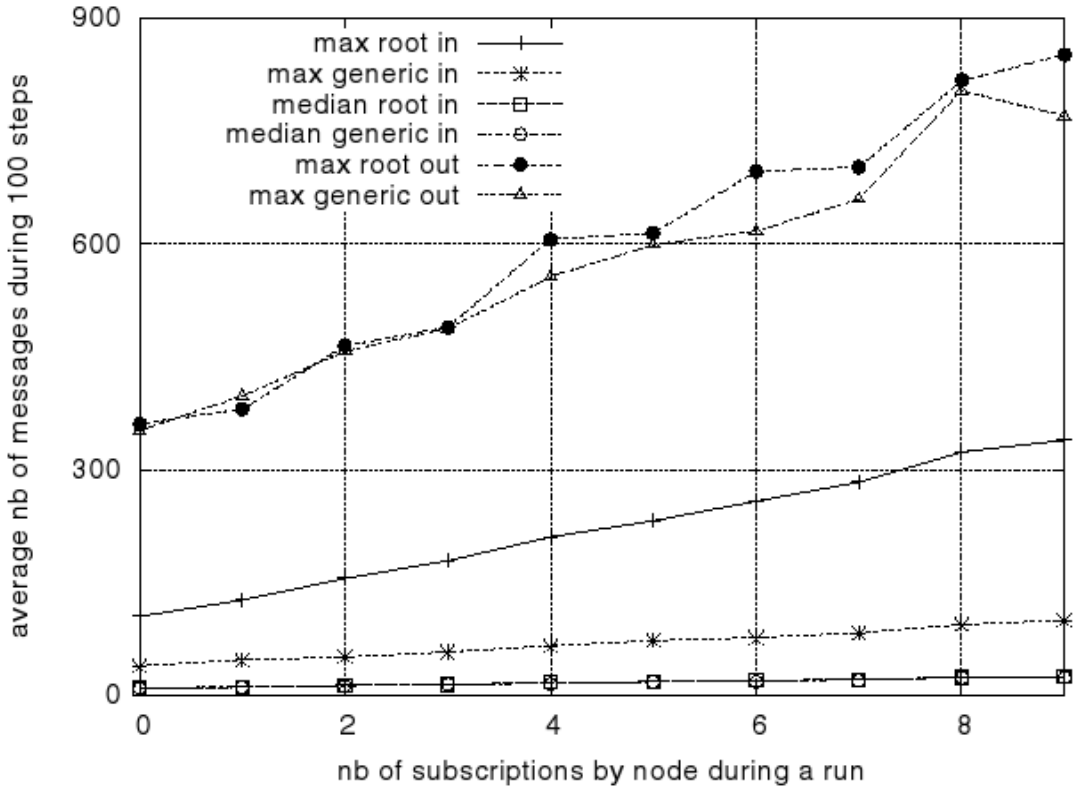


Figure 11. Root-based vs. Generic Approach. [1]

- generic distributes the incoming load among nodes

TECHNISCHE UNIVERSITÄT DRESDEN

Conclusion

- tree traversal alg.
  - root-based approach
    - better for publication
    - lower latency
  - generic approach
    - better for subscription
    - better distribute the load
- communication alg.
  - leader-based approach
    - suitable for small system
    - less failures occur in this case
  - epidemic approach
    - higher dependability, better scalability, load balancing
    - higher message complexity

TECHNISCHE UNIVERSITÄT DRESDEN

Reference

[1] E. Anceaume, A. K. Datta, M. Gradinariu, G. Simon, and A. Virgillito
A Semantic Overlay for Self- Peer-to-Peer Publish Subscribe
in Int. Conference on Distributed Computing Systems (ICDCS), 2006

[2] Gwendal Simon. Publish Subscribe Looking for a Peer-to-Peer Approach
Department of Computer Sciences GET - ENST-Bretagne mars 2007

[3] E. Anceaume, A. K. Datta, M. Gradinariu, G. Simon, and A. Virgillito.
DPS: Self* dynamic reliable content-based publish/subscribe system.
Technical Report 1665, IRISA, 2004.

- Question