Advanced Networking

Project Demo

Mehdi Sadri
Jamshid Esmaelnezhad

Spring 2012

Introduction

Problem: Routing in Pub/Sub systems

Publishers who submit information and
Subscribers who express their interest to receive messages in specific types of information.

Two common forms of filtering:

- **Topic-Based**
- **Content-Based**

- Decoupling in **Space**, **Time** and **Flow**
- Fixed or Dynamic set of **Brokers**
- **Broker overlay** is important, no flooding
  - Change overlay in order to make content related nodes closer.

SIENA

Related Works

Scalable Internet Event Notification Architecture

- Generic **Scalable** Pub/Sub Event Notification Service
- Maximizing both expressiveness and scalability
- A network of Servers.
- Tree based routing scheme.

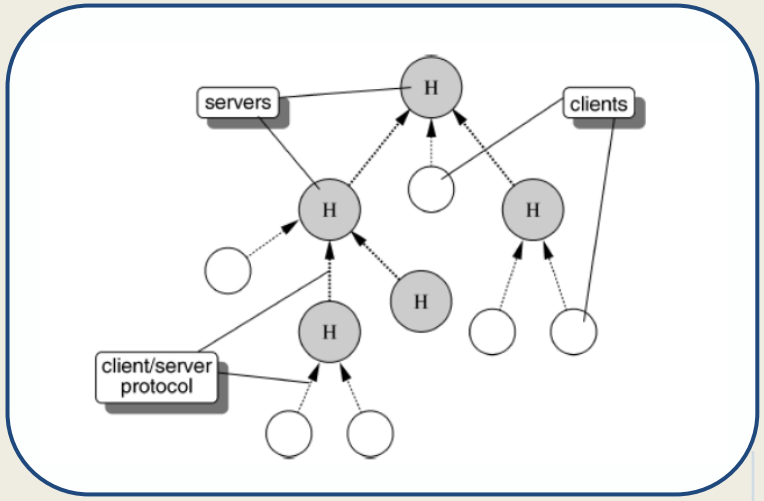
- API:
  - **Publish**: a publisher publish a notification.
    Subscribe: a subscriber describe types of notifications it is interested in.
  - **Unsubscribe**: a subscriber describe types of notifications it is no longer interested in.
  - **Advertise**: a publisher describe types of notifications it publishes.

Unadvertise: a publisher describe types of notifications it no longer publishes.

SIENA (cont.)

Routing

- Simple Solution: broadcast (flooding)
  - has a lot of overhead
- Some principles to improve it:
  - **Downstream replication**
  - **Upstream filtering and pattern assembly**
  - **Common principles**: loop-free paths, efficient forwarding, reduced state, etc.
  - **Implemented by two classes of algorithms**: subscription forwarding and advertisement forwarding.



Reorganizing topology

Contributions and Algorithms

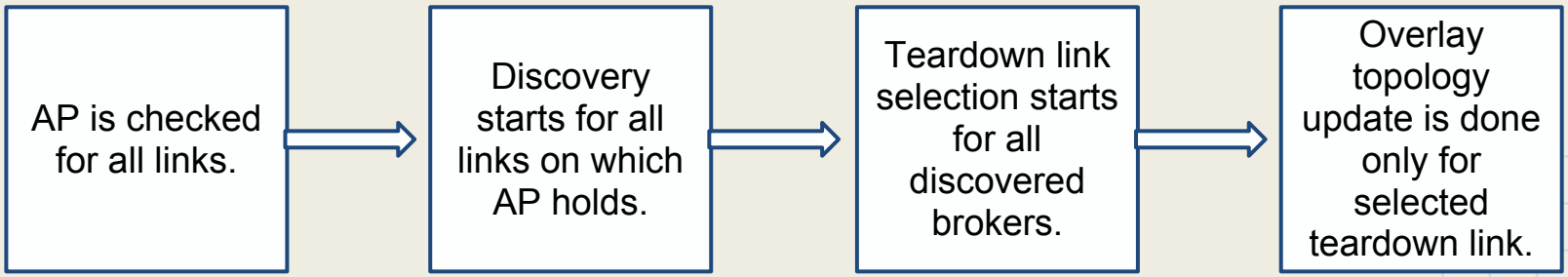
- **Main goal**: brokers with more similar interests become closer to each other
- **Main idea:** Bi and Bl are not directly connected, their similarity > the similarity of a pair of directly connected brokers in their path => 1.they should be connected, 2.those two brokers should be disconnected.
- **Similarity Function:**

$$a_{i,j} = \frac{|\{e \in m_i : \exists s \in S_{B_j}, e \; matches \; s\}|}{Q_i}.$$

Reorganizing topology (cont.)

- **Four phases:**

a. Triggering of a similar broker discovery.
b. The actual broker discovery phase.
c. Tear down link selection.
d. Overlay topology update.



Reorganizing topology (cont.)

Phase 1: Triggering of a similar broker discovery

- Periodically
- Backoff policy
  - delta
- checks this predicate for all links

- intuition: probably there should be some broker behind Bj
- too little information? mi has to have at least Q/2 events.

$$\mathbf{AP}: \alpha_{i,j}(m_i) > a_{i,j}.$$

Reorganizing topology (cont.)

Phase 2: The actual broker discovery phase

- done for any link for which AP holds
- DREQ message:
  - HS is built up through the forwarding process
  - for each broker HS has
    i. broker similarity to the initiator
    ii. link weight of broker and the previous broker
  - checks FP and forwards DREQ on each link for which FP holds:

$$\textbf{FP} : \exists l_{j,h} \neq l_{j,k} : \alpha_{j,h}(m_i) > a_{i,j}.$$

  - of no link holds FP, DREP is created and sent back.
- DREP message: A tree recursive approach is used to return the most similar broker to the initiator.

Reorganizing topology (cont.)

Phase 3: Tear down link selection

- Bi and Bl are the two ends of the resulting HS
- The new link to be added is then selected:
  - $l(Bi) = 0$ & $l(Bl) = 0$
  - $l(Bi) > 0$ & $l(Bl) > 0$
  - $l(Bi) > 0$ & $l(Bl) = 0$ **or** $l(Bi) = 0$ & $l(Bl) > 0$
- the next phase is done if lnew is not NULL

Phase 4: Overlay topology update

- before adding the teardown link some link should be removed to keep the acyclic property
- LOCK command sent from the initiator
- path is locked and ACK is sent back to the initiator
- CLOSE message is sent from both sides (Bi and Bl)
- UNLOCK is sent back to both of them

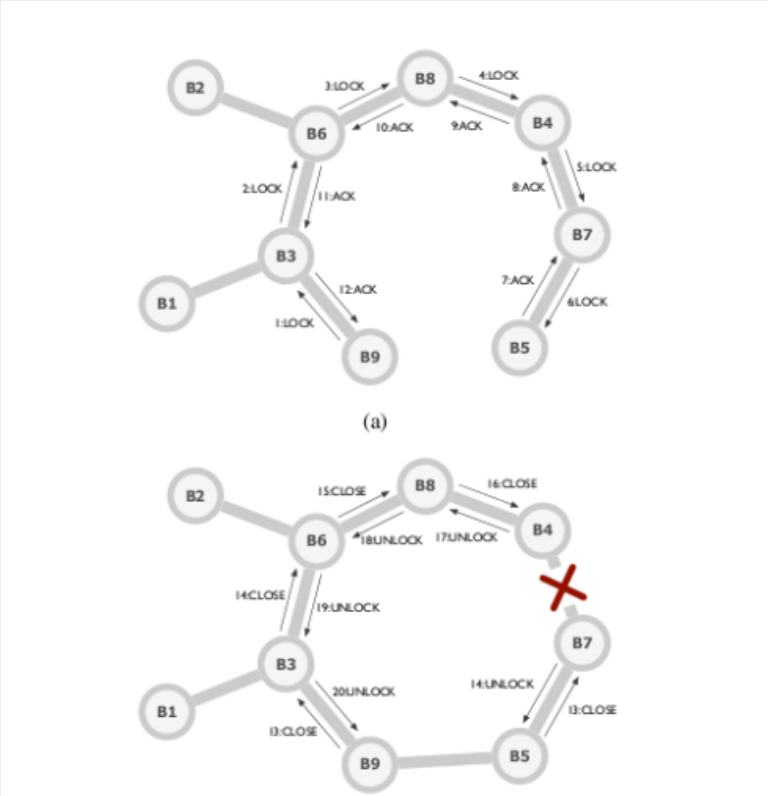
Reorganizing topology (cont.)

Phase 3: Tear down link selection

- Bi and Bl are the two ends of the resulting HS
- The new link to be added is then selected:
  - l(Bi) = 0 & l(Bl) = 0
  - l(Bi) > 0 & l(Bl) > 0
  - l(Bi) > 0 & l(Bl) = 0 **or** l(Bi) = 0 & l (Bl) > 0
- the next phase is done if lnew is not NULL

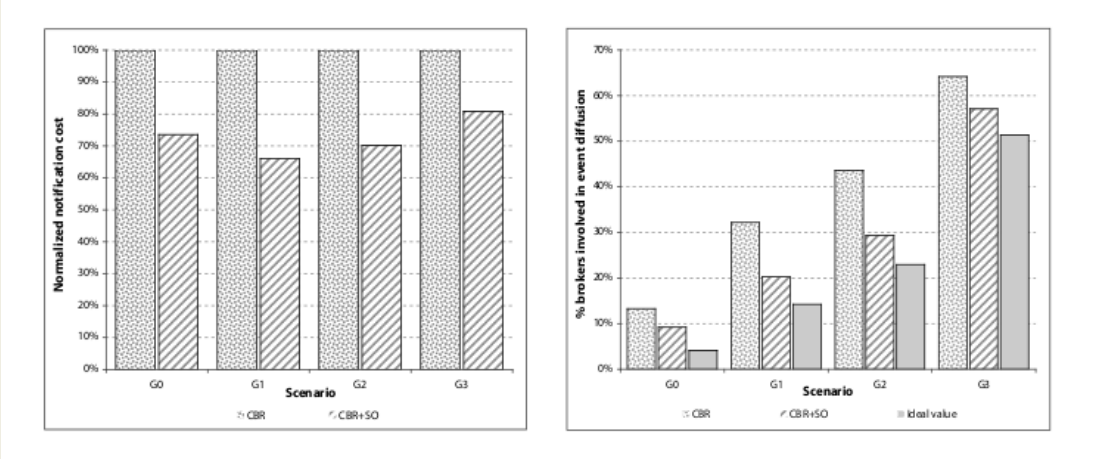
Phase 4: Overlay topology update

- before adding the teardown link some link should be removed to keep the acyclic property
- LOCK command sent from the initiator
- path is locked and ACK is sent back to the initiator
- CLOSE message is sent from both sides (Bi and Bl)
- UNLOCK is sent back to both of them



Experimental Results

- **Data model**: The experiments were run over a 100 nodes TCP/IP network
- **Performance metrics**:
  - Number of reorganizations
  - Notification cost
  - Percentage of brokers involved
- **Two scenarios**: U (Uniform) and G (Gaussian)



Left: average notification cost for event dissemination (values are normalized with CBR set to 100%), Right: average percentage of brokers involved in each event dissemination

References

[1] *R. Baldoni, R. Beraldi, L. Querzoni, and A. Virgillito,Efficient publish/subscribe through a self-organizing broker overlay and its application to SIENA, in Comput. J., vol. 50, no. 4, 2007.*

[2] *Carzaniga, A., Rosenblum, D., and Wolf, A. (2001) Design and evaluation of a wide-area event notification service. ACM Transactions on Computer Systems, 3, 332–383.*

[3] *Cugola, G., Nitto, E. D., and Fuggetta, A. (2001) The jedi event-based infrastructure and its application to the development of the opss wfms. IEEE Transactions on Software Engineering, 27, 827–850.*

[4] *M¨hl, G. (2002) Large-Scale Content-Based Publish/Subscribe Systems. PhD thesis, Technical University of Darmstadt.*

[5] *Cao, F. and Singh, J. P. (2004) Efficient event routing in content-based publish-subscribe service networks. Proceedings IEEE INFOCOM, Hong Kong, China, 7-11 March, pp. 929 – 940. IEEE, Washington.*

[6] *The Gryphon Team (2004) Achieving Scalability and throughput in a Publish/Subscribe System. Technical report.IBM Research Report RC23103.*

[7] *Pietzuch, P. and Bacon, J. (2002) Hermes: A distributed event-based middleware architecture. Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (DEBS), Vienna, Austria, 2-5 July, pp. 611 – 618. IEEE Computer Society, Washington.*

[8] *Oki, B., Pfluegel, M., Siegel, A., and Skeen, D. (1993) The information bus - an architecture for extensive distributed systems. Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles, Asheville, North Carolina, USA, 5-8 December, pp. 58–68. ACM Press, New York.*