SemCast: Semantic Multicast for Content-based Data Dissemination

Olga Papaemmanouil
Brown University

Uğur Çetintemel
Brown University
Wide Area Stream Dissemination

- Applications
  - Network monitoring
  - Real-time financial services/enterprise
  - News services (RSS feeds)

- Characteristics
  - High data volume
  - Highly dispersed sources & destinations
  - Low latency delivery
Content-based Dissemination

- High level of expressiveness
  - Profiles are query predicates
- Sources/destinations decoupling
  - Destinations depend on message content
Content-based Pub/Sub

- Content-based routing
  - Predefined acyclic overlay network
    - SIENA [Carzaniga et al., 2001], Gryphon [Banavar et al., 1999]
  - Predicate-based filtering network
    - Upstream profile aggregation

- Limitations
  - Processing cost at each hop
  - Bandwidth consumption
Processing cost overhead

At each level:
- Content-based filtering at each level
  - Maintain data structures at every broker
- Compression/decompression
  - Usually with XML streams
- Encryption/decryption
  - Integrity-critical applications (e.g., financial feeds, distributed games)
- Forwarding costs could dominate dissemination costs
Bandwidth overhead

- Missed tree optimization opportunities
- Clients’ profiles could overlap
- One spanning tree is suboptimal
  - Intermediate nodes may receive irrelevant data

Cost = 13 msg/time unit
Max Latency = 3 hops

Cost = 10 msg/time unit
Max Latency = 3 hops
Our approach: Semantic Multicast

- Constructs multiple content-based (a.k.a. semantic) dissemination channels
  - Semantic split of incoming streams
  - Channels characterized by their content
  - Independent overlay dissemination trees
SemCast’s advantages

- Low processing cost
  - Eliminates local filtering at interior brokers
  - Low processing requirements for intermediate nodes

- Low overall bandwidth requirements
  - Low-cost dissemination trees
  - Enables latency-cost trade off
    - Delay-bounded trees for latency-sensitive applications
Content-based Channelization

- SemCast decides:
  - Number of channels
  - Content of channels
  - Clients subscriptions to channels
  - Channel implementation

- Operational goals
  - No false exclusion
  - Low run-time cost: Overall bandwidth consumption
    - Minimize redundancy among channels’ contents
    - Create efficient dissemination trees
    - Optimal solution for channelization: NP-complete
      [Adler et al., 2001]
System Model

- Source brokers (S)
  - Receive streams from publishers
- Gateway brokers (GB)
  - Receive profiles from subscribers
- Rendezvous points (RP)
  - Roots of channels
- Interior brokers (I)
  - Forward incoming messages
- Coordinators
  - Identify content of channels
SemCast overview

- Subscription management
  - Join existing channels or create new channels

- Periodically reorganizes channels
  - Exploits overlap among profiles
    - Content of channel defined by profiles assigned to it
  - Assign similar profiles to same channel
  - Identify overlap between profiles
    - Statistical & syntactical information
    - Discover containment relations
    - Merge channels with high partial overlap
    - Use cost-based model to assign profiles to channels
Containment relations

- SemCast discovers *containment hierarchies*
  - \( P_j \) contains \( P_i \)
    - Messages matching \( P_i \) are subset of those matching \( P_j \)
  - Profile syntax reveals containment relations
  - Statistics approximate containment relations
- Hierarchies are possible semantic channels
Cost-based Channelization

- Containment hierarchies might cause high data redundancy
  - High message rate for non-overlapping part
  - Small message rate for overlapping part
- Partial overlapping profiles may increase cost
  - Assign similar profiles to different channels
    - Forward matching messages to more than one channel
- Use cost-based model for channelization
Cost-based containment relations

- Place profiles in the same hierarchy (channel) *if* it improves bandwidth consumption
- If $P_j$ covers $P_i$, compare cost
  - $P_i$ and $P_j$ in same channel
  - $P_i$ on different channel
  - Apply lowest-cost scenario
- Cost estimation
  - Statistics on message rate
  - Approximate number of edges
    - Calculate edges for the Steiner tree connecting brokers
Hierarchy Merging

- Merging hierarchies with *partial overlap* reduces cost
  - Use a cost-based model
  - Send non overlapping part through “noise” channels

\[ E_1 : x:\{25,..,55\} \]
\[ E_2 : x:\{15,..,50\} \]

Destinations D₁

Destinations D₂
Hierarchy Merging

- Merging hierarchies with *partial overlap* reduces cost
  - Use a cost-based model
  - Send non overlapping part through “noise” channels

```latex
\begin{align*}
E_1 \land \neg E_2 & : x: [50, \ldots, 55] \\
E_1 \land E_2 & : x: [25, \ldots, 50] \\
\neg E_1 \land E_2 & : x: [15, \ldots, 25]
\end{align*}
```
Incremental tree construction

- Base low cost heuristic
  - Request channel’s gateway broker from RP
Incremental tree construction

- Base low cost heuristic
  - Request channel’s gateway broker from RP
  - Find min-cost path to all destinations in the channel
  - Connect to closest one
  - Incremental Steiner tree

- Delay-bounded trees
  - Find min-cost path to one broker in the tree that covers delay bound.
Experimental study

- Metrics
  - Processing cost
  - Bandwidth efficiency

- Approaches
  - Unicast approach
  - SPT: Shortest Path Tree approach
    - Distributed pub-sub system [Carzaniga et al., 2001]
  - SemCast: Distributed content-based channelization
    - SemCast-O: Centralized Steiner tree construction

- Network environment
  - Graphs generated by GT-ITM
Bandwidth efficiency-
Disjoint profiles

higher bandwidth consumption

Network size = 600 nodes
Profiles= 7000

SemCast’s trees are close to Steiner trees
SemCast performs better than SPT, Unicast with no overlapping profiles
Bandwidth efficiency - Overlapping profiles

SemCast performs better than SPT with overlapping profiles
Scalability

higher bandwidth improvement

% cost improvement over SPT

<table>
<thead>
<tr>
<th>aggregated profile overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
</tr>
</tbody>
</table>

- SemCast (N=300)
- SemCast (N=400)
- SemCast (N=600)

N = Network size
Profiles = 5 × N
Selectivity factor = 2

SemCast’s improvement increases with the network size
Related Work

- Publish-Subscribe Systems
  - Distributed approaches: Content-based Routing
    - Gryphon [Opyrchal et al., 2000]
    - SIENA [Carzaniga et al., 2001]
    - ONYX [Diao et al., 2004]
  - Centralized approaches: XML Filtering
    - XFilter [Altinel et al., 2000], YFilter [Diao et al., 2002]
    - XTrie [Chan et al., 2002]
- Application-Level Multicast
  - SplitStream [Castro et al., 2003]
  - Scribe [Castro et al., 2002]
  - CAN-based Multicast [Ratnasamy et al., 2001]
Conclusions & Ongoing Work

- SemCast
  - Performs semantic split of incoming streams
  - Eliminates local filtering in interior brokers
  - Improves bandwidth consumption

- Ongoing work
  - SemCast prototype
  - More expressive profiles
    - Statefull subscriptions
  - Wireless environment