Semantic Multicast for Content-based Stream Dissemination

Olga Papaemmanouil
Brown University

Uğur Çetintemel
Brown University
Stream Dissemination Applications

- Push-based applications
  - Environmental monitoring
  - Real-time financial services

- Characteristics
  - High data volume
  - Fast stream rates
  - Highly dispersed sources & destinations
Content-based Dissemination

- Centralized filtering
  - Single node gathers profiles
  - Unicast to destinations
  - e.g., XFilter [Altinel et al., 2000]

- Distributed content-based routing
  - Interest-based routing
  - Predefined acyclic overlay network
    - e.g. SIENA [Carzaniga et al., 2001]
  - Predicate-based filtering network
    - Upstream profile aggregation
  - Content filtering of each message at each hop
Content-based routing overhead

- Processing cost
  - Local filtering of each message
  - Compression/decompression for XML streams

- Bandwidth consumption
  - Missed tree optimization opportunities
    - Different client sets request different sub-streams
    - One spanning tree of the whole network is not the best solution for low cost
Our approach: Semantic Multicast

- Constructs content-based (a.k.a. semantic) multicast channels
  - Independent overlay dissemination trees
  - Channels characterized by their content

- Advantages
  - Decreases processing cost
    - Eliminates local filtering at interior brokers
  - Decreases overall bandwidth requirements
  - Allows QoS-aware multicast trees
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’ content
    - Create efficient multicast trees
System Model

- Source brokers (S)
  - Receive XML streams from publishers
- Gateway brokers (GB)
  - Receive XPath profiles from subscribers
- Rendezvous points (RP)
  - Roots of channels
- Interior brokers (I)
  - Forward incoming messages
- Coordinator
  - Identifies content of channels
SemCast Overview

- Membership Management
  - Syntactical analysis of profiles
- Dynamic Channelization
  - Exploit statistical information
  - Reorganize channels
Membership Management

- Adding subscriptions
  - Gateway broker forwards unsatisfied subscriptions to coordinator
  - Coordinator creates a new channel for any subscription not covered by existing channels

- Removing subscriptions
  - Gateway broker forwards request to upstream brokers
  - Interior brokers remove channel entries from routing tables
  - Coordinator removes channel if there are no more remaining clients
Profile Containment Hierarchies

- Identify channels covering a profile based on profile syntax
  - Containment algorithms for XPath expressions. e.g. [Wood, 2003]
- Maintain *syntax-based containment hierarchies*
- Parent profile covers children
  - Root is the channel’s content expression

![Diagram of containment hierarchies]

WEBDB 2004, June 18, Paris
Dynamic Channelization

- Syntactic analysis might cause high data redundancy
  - Undiscovered full/partial overlapping profiles
  - Assign similar profiles to different channels
    - Forward matching messages to more than one channel
- Use statistics to re-evaluate channelization
  - Profile overlap
  - Run-time stream rate
Profile Overlapping Relations

- SemCast exploits also partial overlap among profiles
- Partial overlap
  - $P_i \ k$-overlaps with $P_j$: $P_i \subseteq^k P_j$, $k = \frac{\text{match}(P_i P_j)}{\text{match}(P_i)}$
- Containment
  - Special case of partial overlap, $k=1$
  - $P_j$ covers $P_i$: messages matching $P_i$ are subset of those matching $P_j$
Rate-based Hierarchies

- \( P_j \) is parent of \( P_i \) if

\[
P_i \subseteq^1 P_j \text{ and } k = \max_{P_j, i \neq j} \left\{ \frac{k}{r_{j-i}} \mid P_j \subseteq^k P_i \right\}
\]

\( r_{j-i} \) = rate of non-overlapping part between \( P_j \) and \( P_i \)

- \( P_j \) is more general than \( P_i \)
- If multiple candidate parents exist
  - Maximum overlap
  - Low stream rate of redundant messages
Hierarchy Merging

- Highly diverse profiles may increase cost
  - High message replication
  - Large number of channels
  - Large routing tables at interior brokers
- Merging hierarchies with partial overlap reduces cost
  - Use a cost-based model
  - Merge pairs with maximum benefit in bandwidth
    - High profile overlap
    - Efficient single multicast tree
Multicast Tree Construction

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

WEBDB 2004, June 18, Paris
Simulation

- Metrics
  - Processing cost
    - Eliminate need for local filtering (not in this paper)
  - Bandwidth efficiency

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

- Simulation environment
  - Random graph generated by GT-ITM
  - Up to 700 nodes and 7000 profiles
Bandwidth efficiency

% cost degradation over Optimal

profile selectivity percentage

Network size = 700
#profiles = 7000
Scalability

% cost degradation over Optimal

- Unicast
- SPT
- SemCast Heuristic

Network size (profile selectivity factor = 0.7%)
Related Work

- Publish-Subscribe Systems
  - Centralized approaches: XML Filtering
    - XFilter [Altinel et al., 2000], YFilter [Diao et al., 2002]
    - XTrie [Chan et al., 2002]
  - Distributed approaches: Content-based Routing
    - Gryphon [Opyrchal et al., 2000]
    - SIENA [Carzaniga et al., 2001]

- Application-Level Multicast
  - 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
  - Demonstrate processing cost benefit
  - Investigate affects of network topology
  - Storage-oriented model: caching XML nodes