Routing algorithms for content-based networking

Optimal and approximate solutions

José Legatheaux Martins

(Based on a joint work with Sérgio Marco Duarte)

Departamento de Informática da Faculdade de Ciências e Tecnologia da UNL

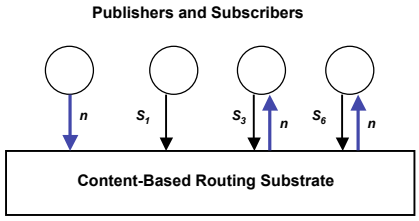
Content-based networking

- **Traditional networking systems address messages to names (network addresses, application level names, ...)**
- **Content-based networking systems address messages to a set of destinations implicitly determined by the message contents**
- **Content-based networking decouple participants in space, time and do not require any previous binding since participants are not required to previously know each other**

2

Main abstractions and terminology

- Participants must previously agree on an *Information Space* syntax and semantics of common interest
- *Receivers* publish their interests in the form of *subscriptions*
- *Publishers* publish *notifications*
- The content-based networking system delivers published notifications to the subscribers with *matching subscriptions*



3

Examples (legacy and emergent)

- Publish / Subscribe (News / RSS feeds, Alert systems, ...)
  - { Symbol = "EDP", Price < 4, Volume > 100000 }
  - { Authors ∋ "Duarte", Keywords ∋ "Event Bases Systems" }
- Distributed Data Sharing, E-Auction Systems, Games, ...
  - { Manufacturer = "VW", Type = "SUV", Kilometers ≤ 20000, Registered ≥ 2001 }
- Monitoring and Control Systems
  - { Speed ≥ 130 Km/h, Within-distance < 4 Km }
- Sensor Network Systems
  - { Temperature > 30, Ground-Humidity < 30%, Report-Period = 300 s }
- There are many possible *information spaces schemas and languages* (open / closed, SQL, XML / Xpath, ...)
- *Quality of Service or Location-awareness* directives can be conveyed in subscriptions / notifications

4

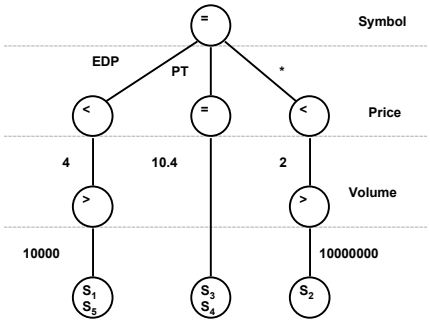
Definitions

- A notification is a tuple of equality attribute value pairs, e.g. $\{A_1=v_1, \ldots, A_n=v_n\}$
- A specific notification denotes a point in a N-dimensional notification space
- A subscription is a predicate made of a conjunction of constraints over attributes, e.g. $\{ A_i=v, A_j \in [v_{min}, v_{max}], A_k \ni v, \ldots\}$
- Constraints of the form "A_i = Any" may be omitted
- A subscription denotes a subspace of the notification space
- A notification *n* matches a subscription *S* if each constraint in *S* holds with the corresponding value in *n*. If it is the case, $n \in$ to the space denoted by *S*
- Subscription S_1 is covered by subscription S_2 iff $S_1 \subseteq S_2$

5

Matching Algorithms — Graph-Based

- Example: Parallel Search Trees
  - Non leaf nodes and edges represent constraints, leafs represent subscriptions
  - Notification *n* will match all reachable leafs

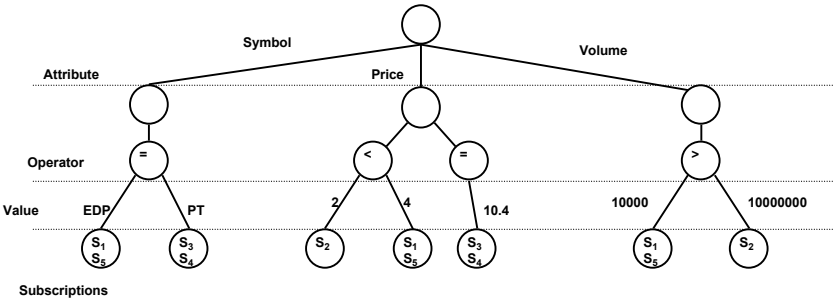


6

Matching Algorithms — Counting-Based

- **Counting Algorithm**
  - All the subscriptions constraints are grouped per attribute and value; some constraints will hold with the individual values of notification *n*
  - Notification *n* will match subscriptions S_i such that

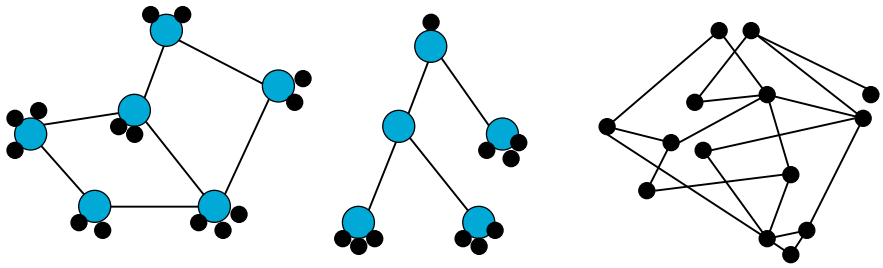
    # of holding constraints of S_i = total # of constraints of S_i



7

System organization

- **Broker-to-Broker routing**
- **End-system-to-End-system routing**



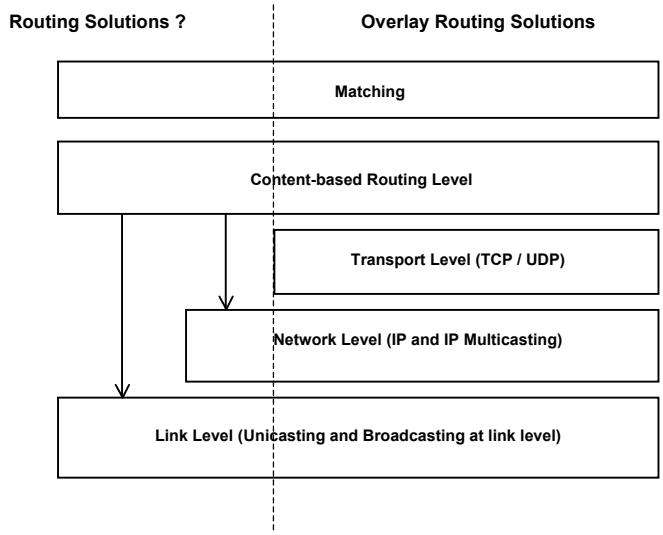
Publisher / Subscriber

Broker

Node-to-Node routing

8

Layer Organization



9

Agenda

- Motivation and definitions
- Optimal routing solution
- Approximate solutions based on multicasting
- Approximate solutions based on the "Learning by the reverse path" principle
- Approximate solutions based on key spaces
- Approximate solutions based on semantic networks
- What needs to be done next

10

Problem presentation

- Routing in a topic-based publish/subscribe problem is similar to the multicast routing problem. However, routing in a content-based system is of a different nature since the dissemination tree must be dynamically pruned to only span the matching subscribers
- There are other problem scenarios where content-based addressing is used: given a set of tuples how to determine the queries they match — e.g. given a set of database updates how to determine the set of triggers they fire ?
- The search problem is dual of the content-based diffusion one

11

Optimal Routing Solution

- A notification *n* must be routed to the group of nodes with matching subscriptions, by an optimal path
- A correct solution must exhibit no "false negatives"
- An optimal solution must deliver no "false positives" or "spam"
- However, an optimal solution must also minimize:
  - The memory utilization
  - The control plane complexity (subscription and network management)
  - The complexity of the matching algorithm (number of times it is executed and size of subscription tables)
- Flooding, is a non optimal solution that is, however optimal in what concerns these last criteria

12

Extra problems posed by mobility and wireless systems

- Mobility introduces an ever changing network were dynamicity is the rule, thus optimal multicasting may be problematic
- In addition, in mobile settings scope and context is of paramount importance
- Wireless systems have limited resources that must be optimized
- Wireless sensor networks are often not mobile. However, due to the problem of battery exhaustion, a content-based solution is:
  - quality of service driven
  - and must be integrated with a query system.

13

An optimal solution based on ideal multicasting

- Let $S = \cup S_i$ be the set of all subscriptions
- Let $\Theta = \cup n_j$ be the set of all notifications
- There is a mapping $\Psi: \Theta \rightarrow S^*$ such that
  $S_i \in \Psi (n) \Leftrightarrow Match (n, S_i)$
- Ψ allows the determination of $C = \{C_1, \ldots, C_M\}$ of M overlapping clusters, subsets of S, such that each notification n matches exactly all the subscriptions of just one of these clusters
- An (optimal) multicasting group G_i=GroupOfCluster(C_i) is associated with each cluster and each node joins all the groups to which its subscriptions belong

- A notification n is multicasted to the group of the corresponding cluster of matching subscriptions, i.e.

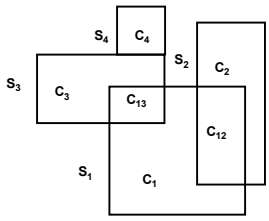
  Send (n, GroupOfCluster(Ψ(n)))

14

Optimal subscription clustering and notification subspaces

- As a subscription denotes a notification subspace, each cluster denotes a subspace entirely contained in the interception of its subscriptions; the union of these subspaces constitutes a partitioning of the notification space
- The optimal subscription clustering corresponds to M notification subspaces SC_K such that

  $\forall\, 1 \le k \le M,\ \forall\, S_i \in S : [\, SC_k \cap S_i \neq \varnothing \,] \Rightarrow [\, SC_k \subseteq S_i \,]$



$C_1=\{S_1\},\ C_2=\{S_2\},\ C_3=\{S_3\},\ C_4=\{S_4\},\ C_{12}=\{S_1,S_2\},\ C_{13}=\{S_1,S_3\}$

15

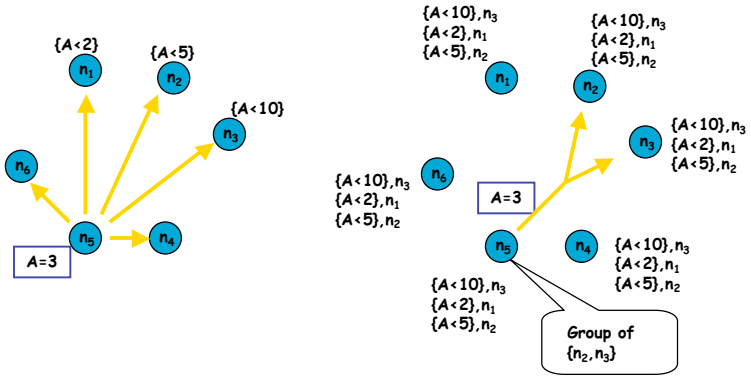
Characteristics and costs of ideal multicast

- Correct, since there are no false negatives
- No spam, since there are no false positives
- Routing cost is optimal (with optimal multicasting)

- Space requirements: O(# subscriptions) in each node
- Only the publisher executes the matching algorithm over all the subscriptions
- Control plane costs: replication of subscriptions and the M multicasting groups maintenance costs

16

Flooding versus ideal multicasting



17

Ideal multicast is in fact ideal

- **The algorithm requires that:**
  - The publisher knows all subscriptions,
  - In each publisher, $\forall\ n \in \Theta$, there is mapping from $\Psi(n)$ to a multicast group, and
  - Some multicasting facility is available
- **In a network of N nodes it may require up to 2^N different groups (e.g. N = 10, # groups ≤ 1024)**
- **Even when IP Multicasting is available, each group has some network cost and therefore groups are not available in huge numbers**

18

Can we make ideal multicast practical ?

- Each publisher can in fact know all subscriptions - the real cost depends of the rhythm of subscription evolution
- Mapping Ψ is byproduct of the matching algorithm
- Mapping of Ψ(n) in an optimal multicasting diffusion tree is the main practical problem
- Solution (1) — still optimal
  - On-demand multicasting — the needed diffusion trees are lazily and dynamically computed
- Solution (2) — non optimal
  - Limit the # of required multicasting groups — introducing spam or repeating multicasts

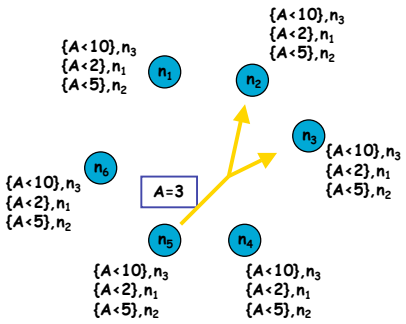
19

The Link Matching Algorithm

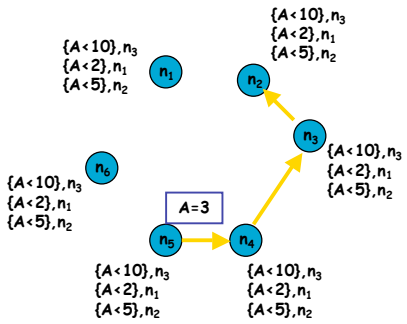
- G. Banavar et al. IBM T. J. Watson Research Center, Gryphon System, "An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems," IEEE DCS, 1999
- Each node has a global view of the network and computes as many Shortest Path Spanning Trees (SPST) as publishing nodes. One expects that in many scenarios there is a reduced number of publishers (e.g. Stock Exchanges). For each one, the local routing table contains the list of local links belonging to the SPST.
- Using an annotated Parallel Search Tree, a notification n and the list of links belonging to the publishers SPST, the algorithm determines a pruned SPT tree for each notification
- The algorithm has some resemblance with MOSPF, which is far from an ideal multicasting solution in terms of the backplane costs.

20

Ideal Multicasting versus the Link Matching Algorithm



Optimal multicasting at network level



"Matching" by every node in the path, and optimal multicasting at overlay level

21

MEDYM: Match-Early with Dynamic Multicast

- F. Cao and J. Pal Singh, Princeton's DADI Project, ACM/USENIX Middleware 2005
- Each node has a complete knowledge of the subscriptions and of the network as in the previews algorithm
- The publisher of n computes Ψ(n) and the corresponding list of matching nodes
- A variant of stateless multicasting is used to diffuse n. Stateless multicast sends messages containing a list of destinations in the header
- The algorithm performs almost as the optimal but stateless multicast does not scale and is not "popular" in the networking community

22

Non optimal solution with clustering (1)

- A. Riabov et al. "Exploiting IP Multicast Content-Based Publish-Subscribe Systems," in Middleware 2000 and L. Opyrchal et al. "Clustering Algorithms for Content-Based Publication-Subscription Systems", in ICDCS 2002, both in the context of IBM T. J. Watson Research Center's Gryphon System
- Solution 1 ⇒ send several multicasts to several groups. If k nodes are clustered in c mutually exclusive clusters, each cluster only needs $2^{k/c}$ groups for a total of $c.2^{k/c}$ groups. 20 nodes in 5 clusters only need $5.2^4=80$ groups instead of 2^{20}
- Clustering by region is the most effective strategy

23

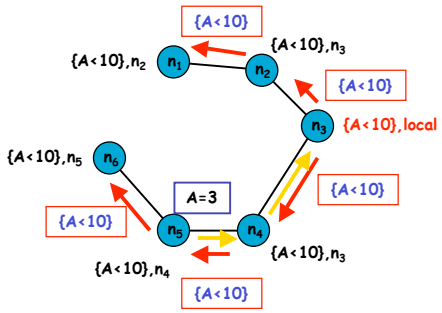
Non optimal solution with clustering (2)

- Solution 2 ⇒ in each cluster reduce group precision by aggregating several groups; if for example, a notification matches more than a threshold number of nodes, send it to the group of all nodes, or aggregate groups with very low matching rate
- All clustering solutions that reduce group precision are non-optimal since they trade multicasting control plane complexity for spam. Flooding is the ultimate spamming solution, a benchmark solution in what concerns simplicity
- Fine tuning the trading above is very dependent of the behavior of the notification, subscription distribution and network configuration

24

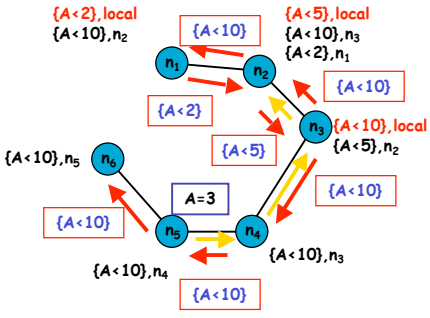
Learning by the reverse path

- A. Carzaniga et al. "Design and evaluation of a wide-area event notification service," in ACM Transactions on Computer Systems, 2001
- Solution adopted in most popular broker systems (Siena, Gryphon, Rebeca, Hermes, ...)
- The algorithm is in general used with a loop-free network overlay
- Bares some similarity to routing in IEEE 802.3 local area networks



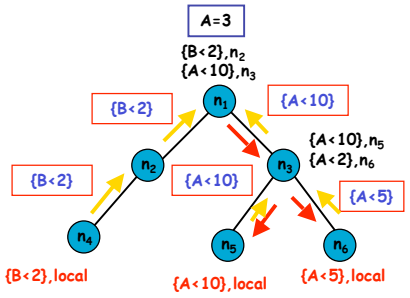
Some observations on the algorithm

- The routing table is made of entries of the form (S_1,link), (S_2,link), (S_3,local), ...
- To reduce matching complexity, subscriptions announced by the same link can be summarized using precise or imprecise summaries
- Precise summaries are often called a *covering subscription set (CSS)*. However, its coexistence with dynamic subscriptions is complex
- Imprecise summaries tend to the broadcasting solution



More aggressive summarization — using a rooted tree

- Publications and subscriptions flow in a well defined way and the subscriptions tables are reduced in size
- In a network of n nodes, each one with a different subscription, in an well balanced tree with O(log n) levels, the root will know n subscriptions, but the leaves, i.e. at least half of all nodes, will only know 1 subscription



Dynamic networks with cycles

- Carzaniga et al. "A Routing Scheme for Content-Based Networking," in Infocom 2004, proposed a more general "learning by the reverse path" algorithm supporting a network with loops and dealing with dynamic subscriptions
- Optimally can only be taken for granted with rooted shortest path trees
- Due to dynamic subscriptions, simple summarization management increases spam level with time
- The proposed algorithm leverages a pre-existing loop-free broadcasting protocol and introduces a new subscription managing protocol that periodically recalculates the subscriptions summaries

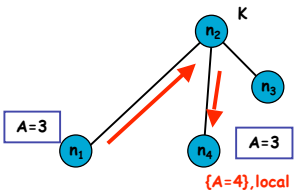
Mapping the Notification Space in a Key Space

- Yi-Min and Lili Qiu, et al. "Subscription Partitioning and Routing in Content-Based Pub/Sub Systems," IEEE DCS 2002
- Proposes a new way of making notifications meet the matching subscriptions when notifications and subscriptions only have equality operators and cover all attributes of the schema

n = {A=2, B=4, C="EDP", ...}
S = ...

Key(n) = Hash(n)
Key(S) = Hash(S)

Match(n, S) ⇒ Key(n) = Key(S)

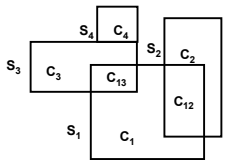


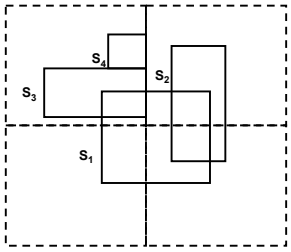
Sender and receiver meet at a rendezvous node in charge of a key range. The scheme introduces spam when there is no matching subscriber

29

Dealing with range subscriptions

- Tam, D. and Azimi, R. and Jacobsen, H.A., "Building Content-Based Publish/Subscribe Systems with Distributed Hash Tables," in DBISP2P, 2003
- Partitions the notification in cells and maps each cell in a key. Publishers and subscribers meet using the Pastry DHT and a variant of Scribe multicasting system for multicasting for nodes in the same cell
- The idea is appealing but the big challenges are: spam, load balancing due to hotspots, dealing with range and incomplete subscriptions and schema restrictions ....





30

More on range subscriptions and DHTs

- Gupta, A. et al. "Meghdoot: Content-Based Publish/Subscribe over P2P Networks," in ACM/Usenix Middleware 2005
- Aekaterinidis, I. and Triantafillou, P., "PastryStrings: A Comprehensive Content-Based Publish/Subscribe DHT Network," in ICDCS 2006

- CAN and Pastry DHTs are used to improve the approach. The first one improves on the load-balancing problem. The second one improves on ranges subscriptions with strings. Both require several multicasts for each notification

- Evaluations of these proposals are quite incomplete, to say it short, because spam and load are very dependent on the notification / subscription distribution, are not orthogonal of the schema and real network and churn costs are not considered

31

Semantic networks

- Ideal model: each node has its own subscription and nodes can choose whatever link they need
- Challenge: can the nodes be organized in a way that notifications find the matching nodes and optimally navigate through all of them ?
- There are several proposals based on
  - Clustering
  - The distribution of the parallel search tree
  - Gossiping biased by a proximity function

32

Sub-2-Sub — a gossiping example

- Voulgaris, S. Et al, "Sub-2-sub: Self-organizing content-based publish and subscribe for dynamic and large scale collaborative networks," in IPTPS 2006
- This gossiping system is based on three different types of views
- One view in each node to manage the global membership
- One view in each node based on a semantic proximity function
- Several exact views in each node, required to visit the nodes with subscriptions associated with each of the subscriptions groups of the optimal solution
- All gossiping systems require duplicate detection and spread all over the network a continuous evaluation of the matching algorithm
- Evaluations of the proposal consider churn but are otherwise limited in scope

33

Brief overview

Method	Routing	Matching executed by	Spam	Nodes routing a notification	Nodes storing a subscription
Flooding	Flooding	All nodes	Yes and duplicates	All	None
Ideal Multicast	Optimal at network level	Sender	No	None	All
Dynamic Multicast	Optimal at overlay level	Sender or sender and path nodes	No	Nodes of the optimal path	All
Learning by the reverse path	Optimal in specific scenarios	Sender and path nodes	Yes during certain periods	Nodes of the optimal path	Several
Key space-based or rendezvous-based	Depends	Sender and a subset of path nodes	Yes	Several	Several
Semantically driven gossiping	Random walk plus diffusion	Several and the destination nodes	No but duplicates	Several	Several

34

Lots of things remaining to be done

- Evaluation Methodology, Workloads and Network Models
- Test new realizations of "old ideas" (On-Demand Multicast)
- Real Internet wide scale and real situations
- Do new ideas like DHTs and gossiping are valuable or just "yet another product of the complexity factory" ?
- Consider mobile and wireless scenarios

35