Forwarding in a content based Network

Carzaniga and Wolf

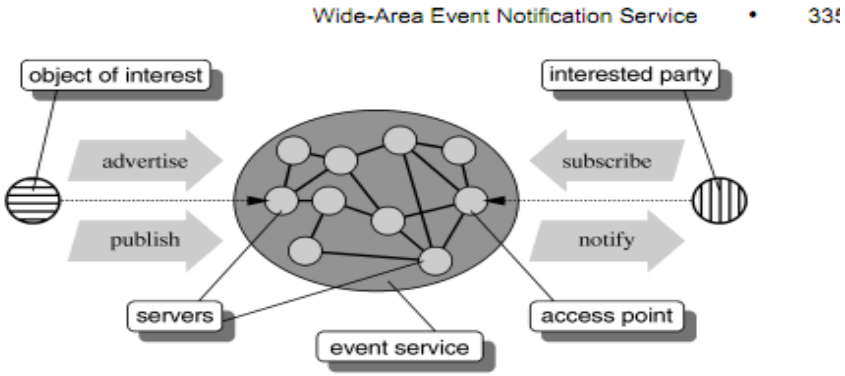
Introduction

- Content based communication
  - Instead of explicit address
- Receiver/Subscriber use selection predicate to express what they are interested in
- Sender/Publisher just push data into the network
- Model
  - attribute/value pair
  - Selection predicate logical disjunction of conjunction [alert-type="intrusion" Λ severity>2 V class="alert" Λ device-type="web-server"]

Message and Predicate

[class="alert", severity=6, device-type="web-server", alert-type="hardware failure"]

[alert-type="intrusion" ∧ severity>2 ∨ class="alert" ∧ device-type="web-server"]

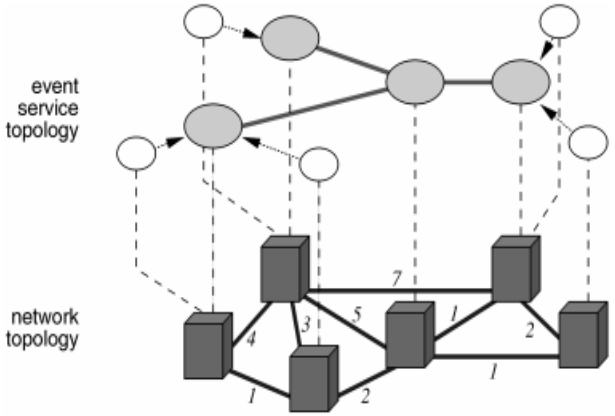


Application

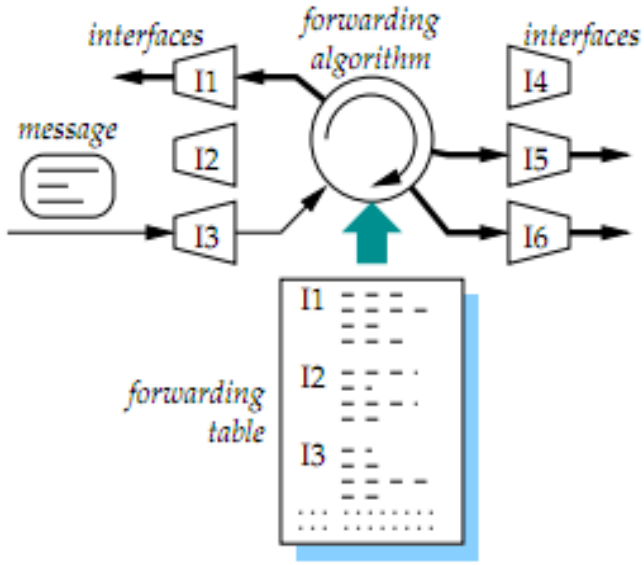
- Publish/Subscribe event notification
- System Monitoring Management
- Sensor Network
- Data Sharing

Overlay Network

- Build on top of an other network
- Connected by Logical Link
- Logical link can correspond to one or several physical link in the underlying network
- Example: JXTA (Sun)
  Solipsis (France Telecom)
  Gnutella
  DHT (Chord, Pastry, Tapestry)



- One-to-one association between filters and interfaces
- *Is* is an interface
- fs.t is a filter
- constraints in the third column
- Use the *counting algorithm* to determine which filters match the message



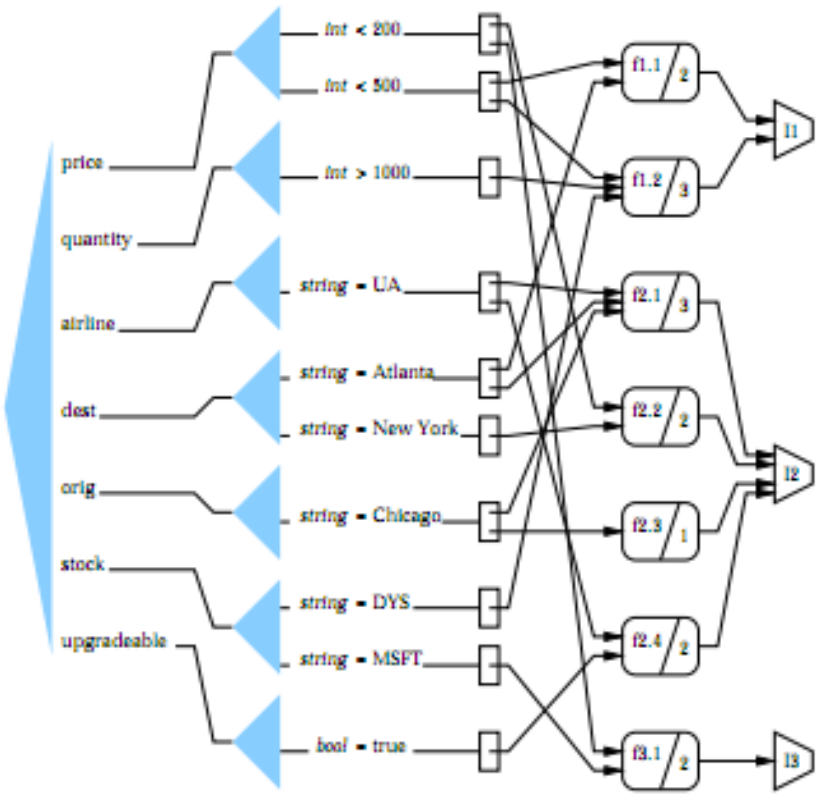
I_1	$f_{1,1}$	*string dest = Milano* *int price < 500*
	$f_{1,2}$	*string stock = DYS* *int quantity > 1000* *int price < 500*
I_2	$f_{2,1}$	*string airline = UA* *string orig = Denver* *string dest = Milano*
	$f_{2,2}$	*string dest = New York* *int price < 200*
	$f_{2,3}$	*string orig = Denver*
	$f_{2,4}$	*string airline = UA* *bool upgradeable = true*
I_3	$f_{3,1}$	*string stock = MSFT* *int price < 200*

Forwarding table

- Organized in two parts left-to-right
- Left side
  - all individuals constraints associated with neighbours router
  - connected to the right side as boolean input
- Right side
  - conjunction of constraints into filter
  - disjunction of filter into predicates of interface

Counting algorithm

- Iterates through all attributes
- Find constraints matched
- Iterate through matched constraints to find matched filters

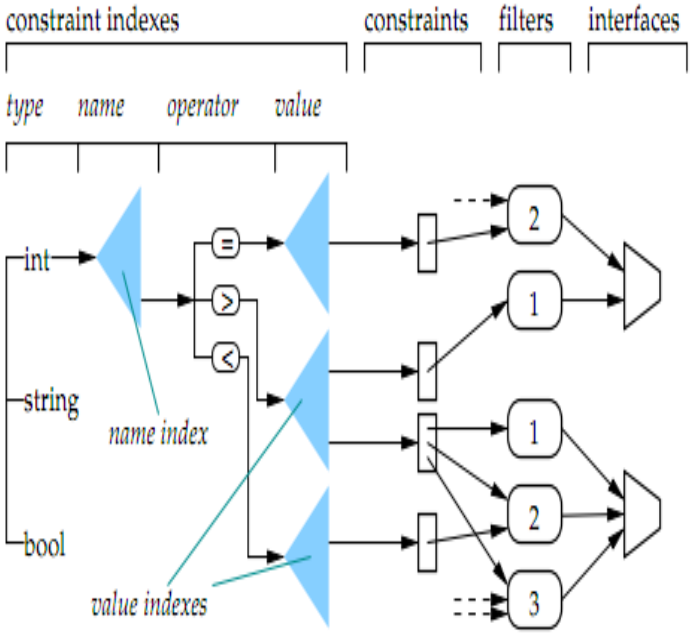


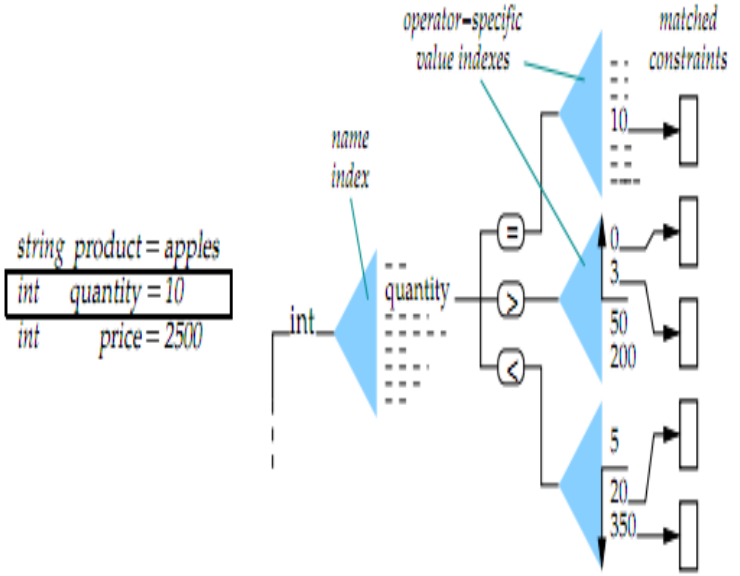
Counting algorithm

- Algorithm
  - use 2 data structures
    - tables of counters for partially matched filters
    - set containing interface to which the message should be forwarded
  - for each constraints found through the constraints index, the algorithm increments the counter of all filters linked to that constraint
  - when the counter equals the number of constraints linked to it, the filter is satisfied and the interface can be added to the set of matched interface

Index based optimisation

- Switch on the type of an attribute
- TST on the name of the attribute
- Specific optimization related to the type of the attribute are done to index constraint (lesser than for integer constraint sorted)





In the case of string

- Support operators =, <, >, suffix, prefix and substring all in a single index
- See the paper for more details

Attribute Selectivity

- Save time by eliminating from consideration ASAP
- Given an attribute a and an interface I, a is determinant if every filter of interface I contains at least one constraint on a
- Selectivity table associate attributes with the interface for which they are determinant
- Check this table as pre-process phase to eliminate interfaces that will not match (for example, if at

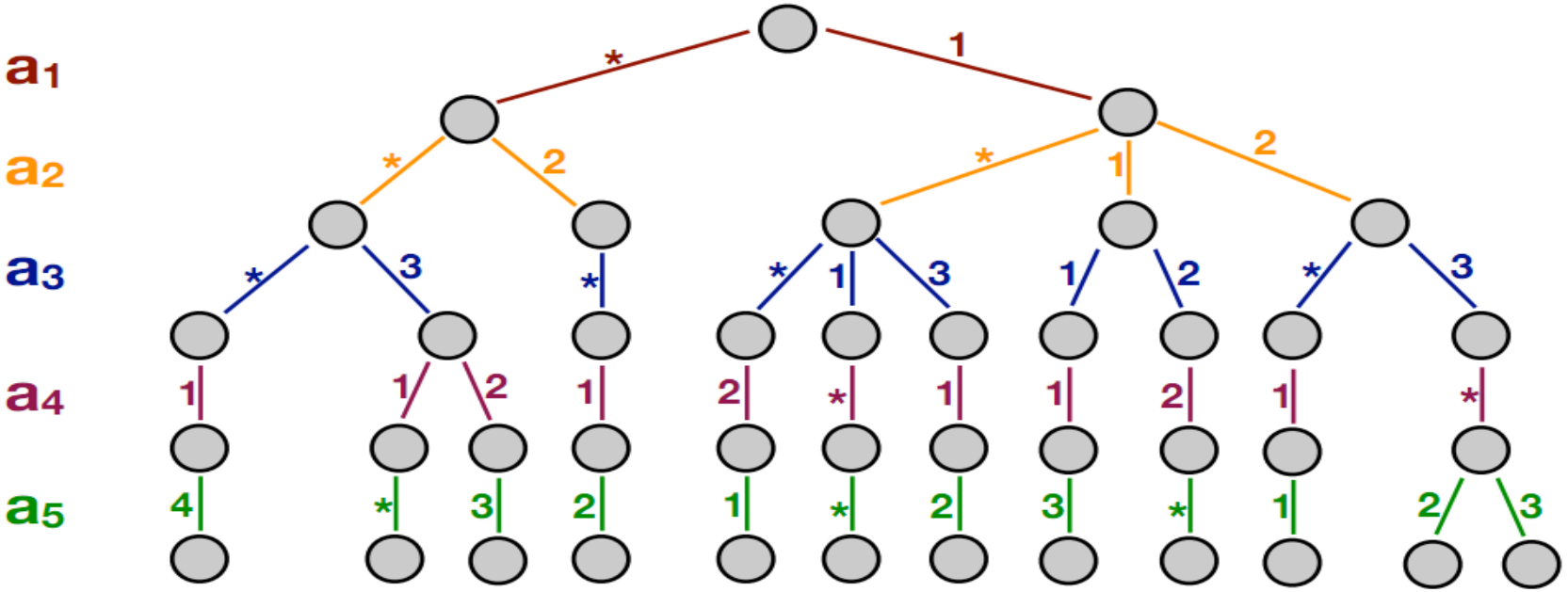
An other solution PST

Matching Events in a Content-based Subscription System

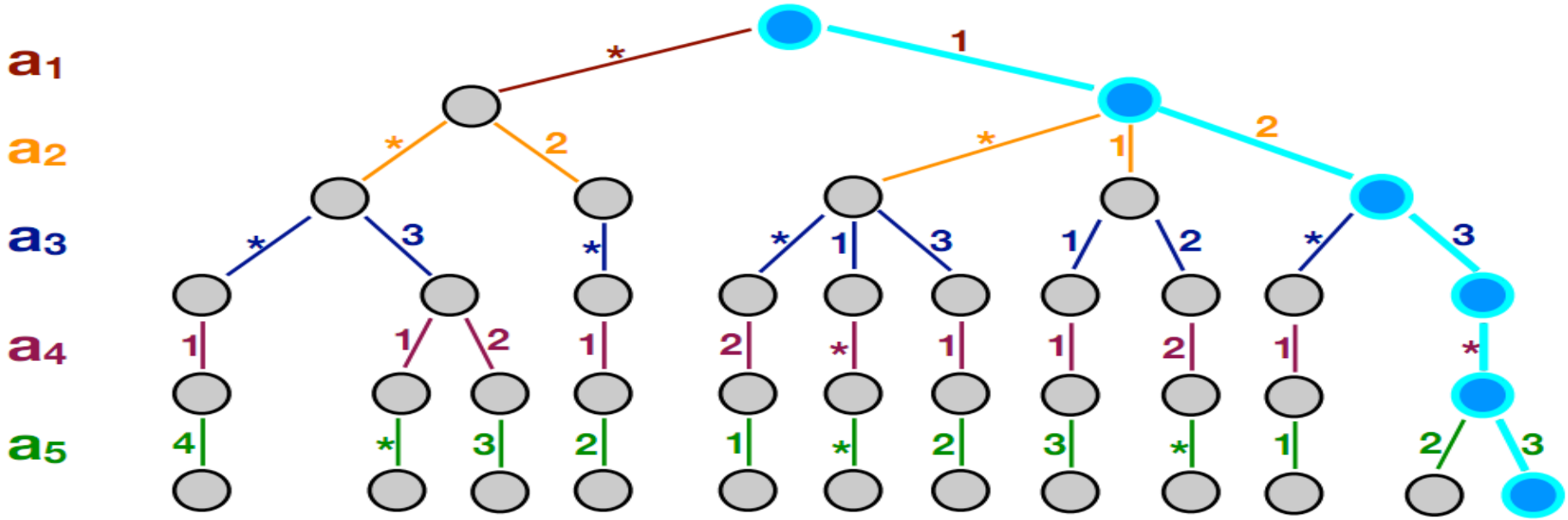
Marcos K. Aguilera,Robert E. Strom, Daniel C. Sturman, Mark Astley and Tushar D. Chandra

Existing Approach

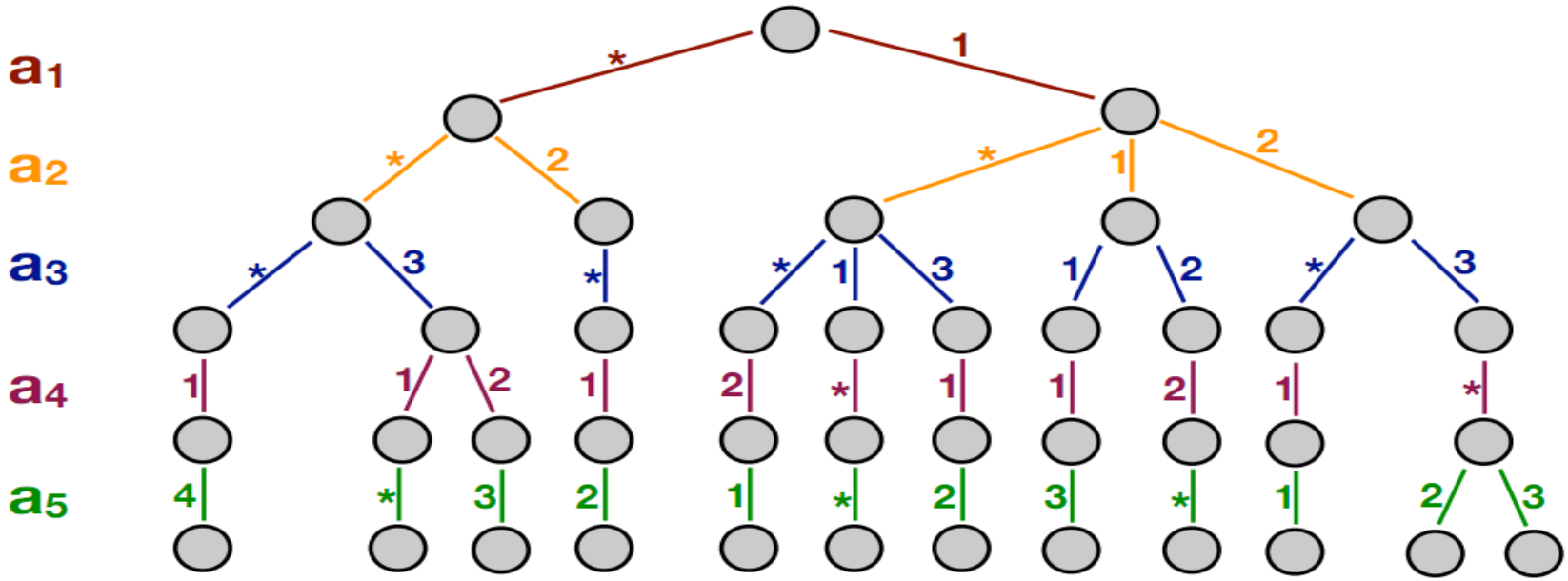
- Flooding approach: message are send to everyone and clients filter what they receive
- Match first approach: message is matched against all filter list to generate a destination list and then the message is routed to all entry on this list



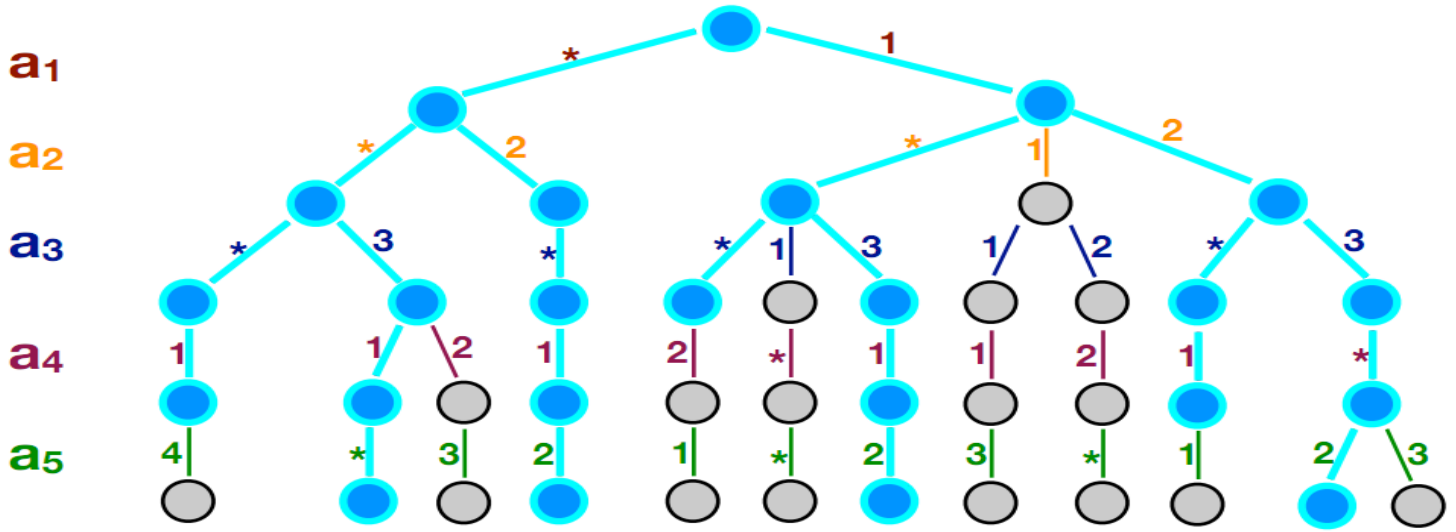
a1
a2
a3
a4
a5



$(a_1=1 \;\&\&\; a_2=2 \;\&\&\; a_3=3 \;\&\&\; a_5=3)$



a1
a2
a3
a4
a5



a = <1, 2, 3, 1, 2>

Notes

- Other type of test than equality can be performed
- The order in which the attribute are ordered for root to leaf is arbitrary
  - however, the algorithm performs better if the attribute near the root have the fewest "*"

Distributed Improvement

An Efficient Multicast Protocol for Content-Based Publish-Subscribe System

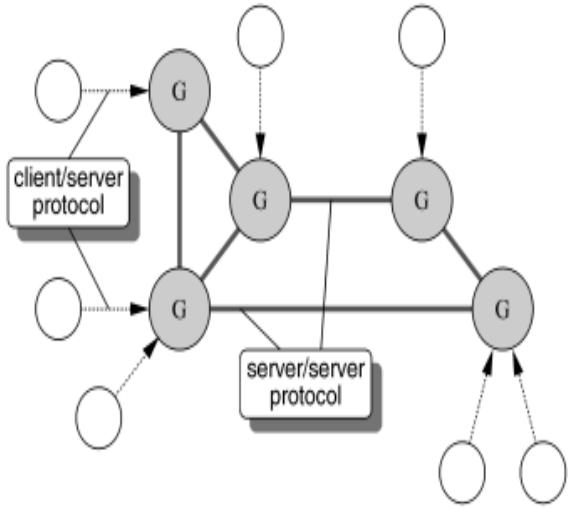
Guruduth Banavar, Tushar Chandra, Bodhi Mukherjee, Jay Nagarajarao, Robert E. Strom, and Daniel C. Sturma

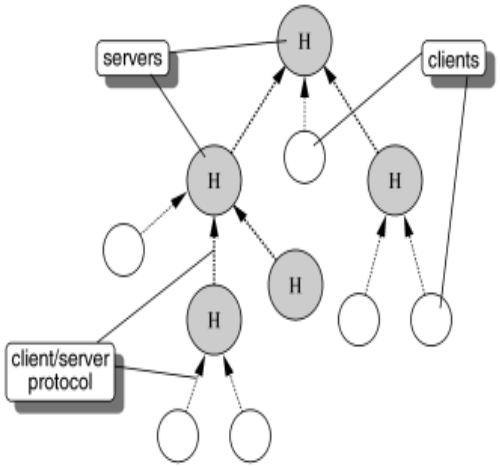
Related consideration

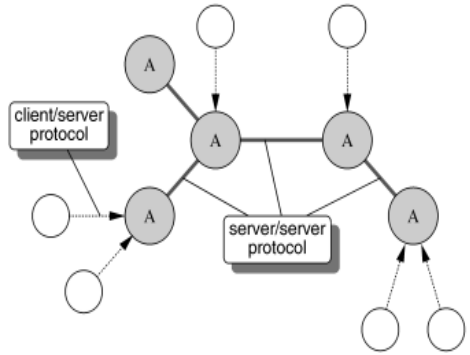
Design and Evaluation of a Wide-Area Event Notification Service

ANTONIO CARZANIGA, DAVID S. ROSENBLUM and ALEXANDER L. WOLF

Architecture

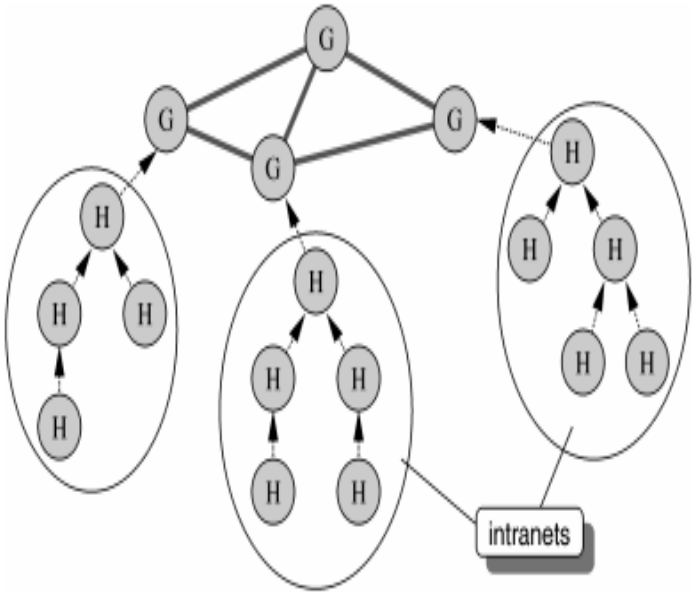


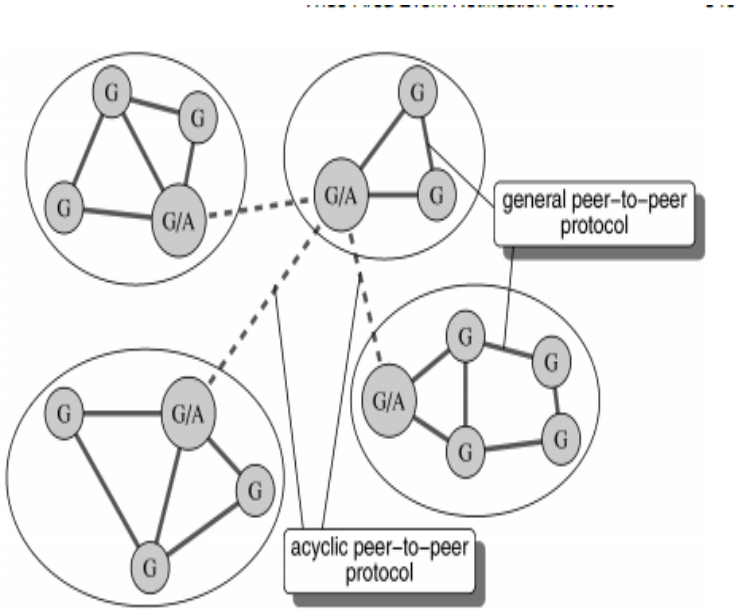




The paper presented assumed an acyclic topology

Hybrid Architecture





The routing or forwarding algorithm to be used depends on the topology

Question ?

Merci

tfjmp2@cam.ac.uk