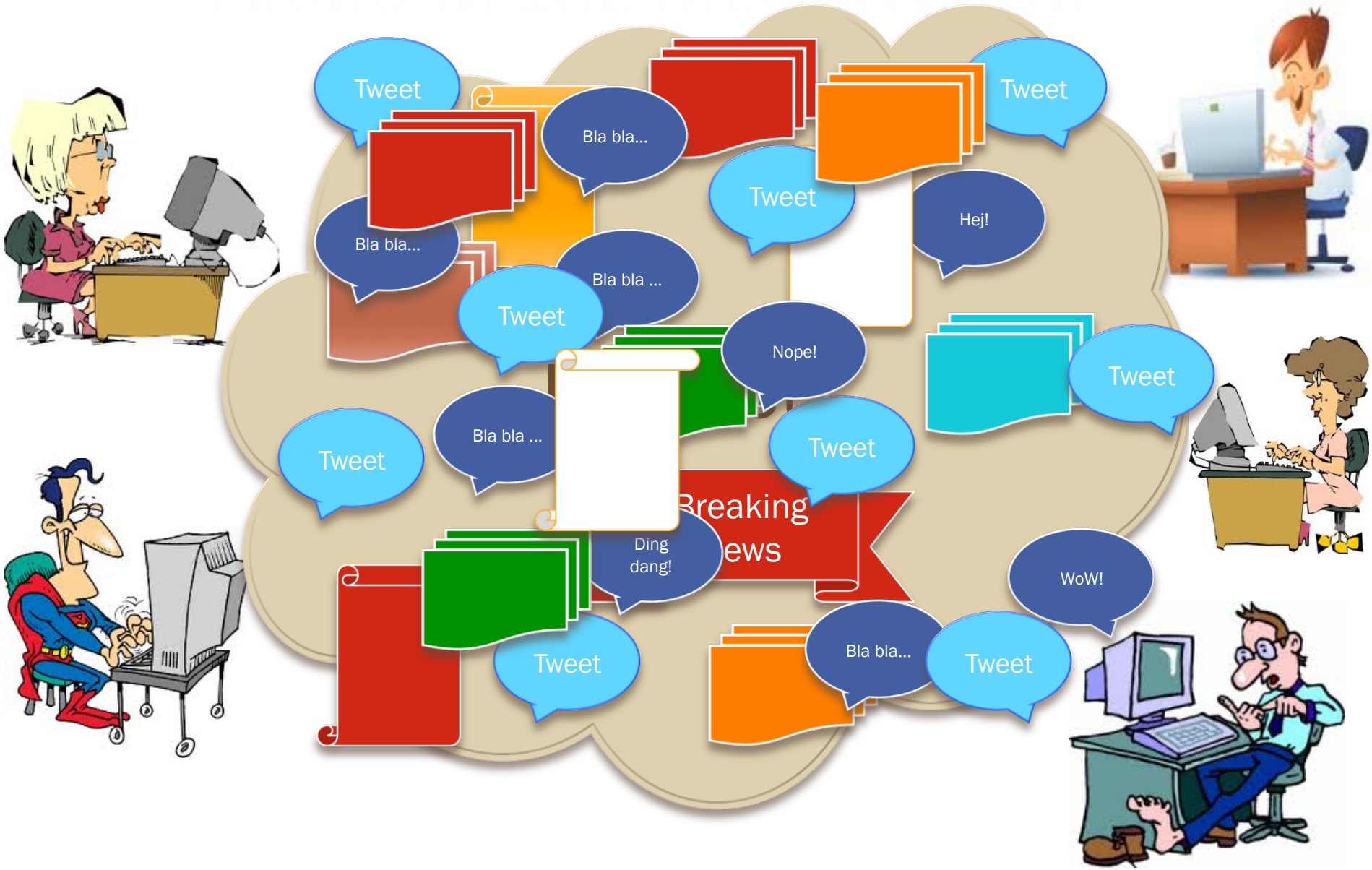
DISTRIBUTED PUBLISH/SUBSCRIBE SYSTEMS: A QUICK SURVEY

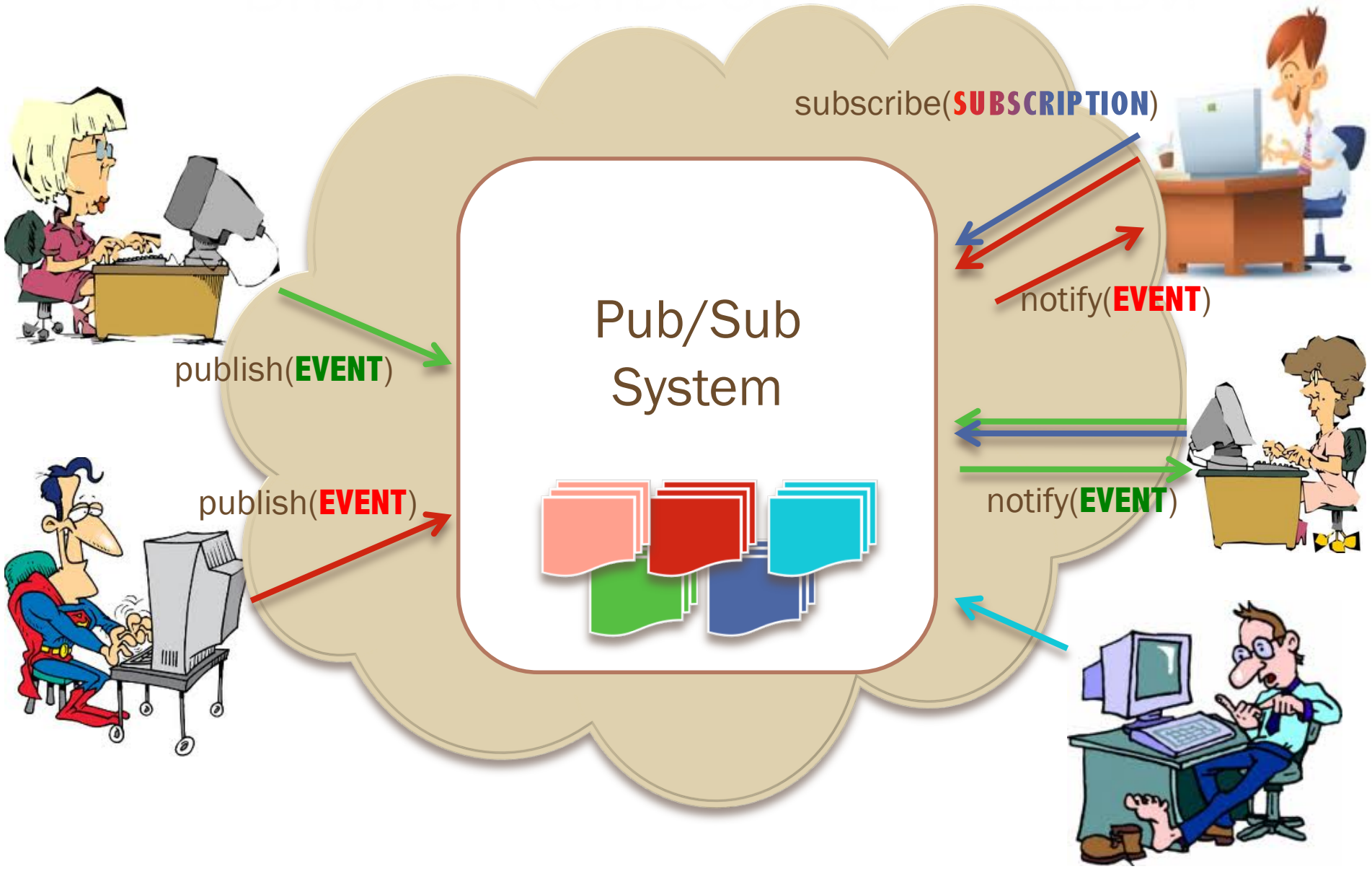
ID2210: LECTURE 5

FATEMEH RAHIMIAN

LIVING IN THE INFORMATION ERA



PUBLISH/SUBSCRIBE PATTERN



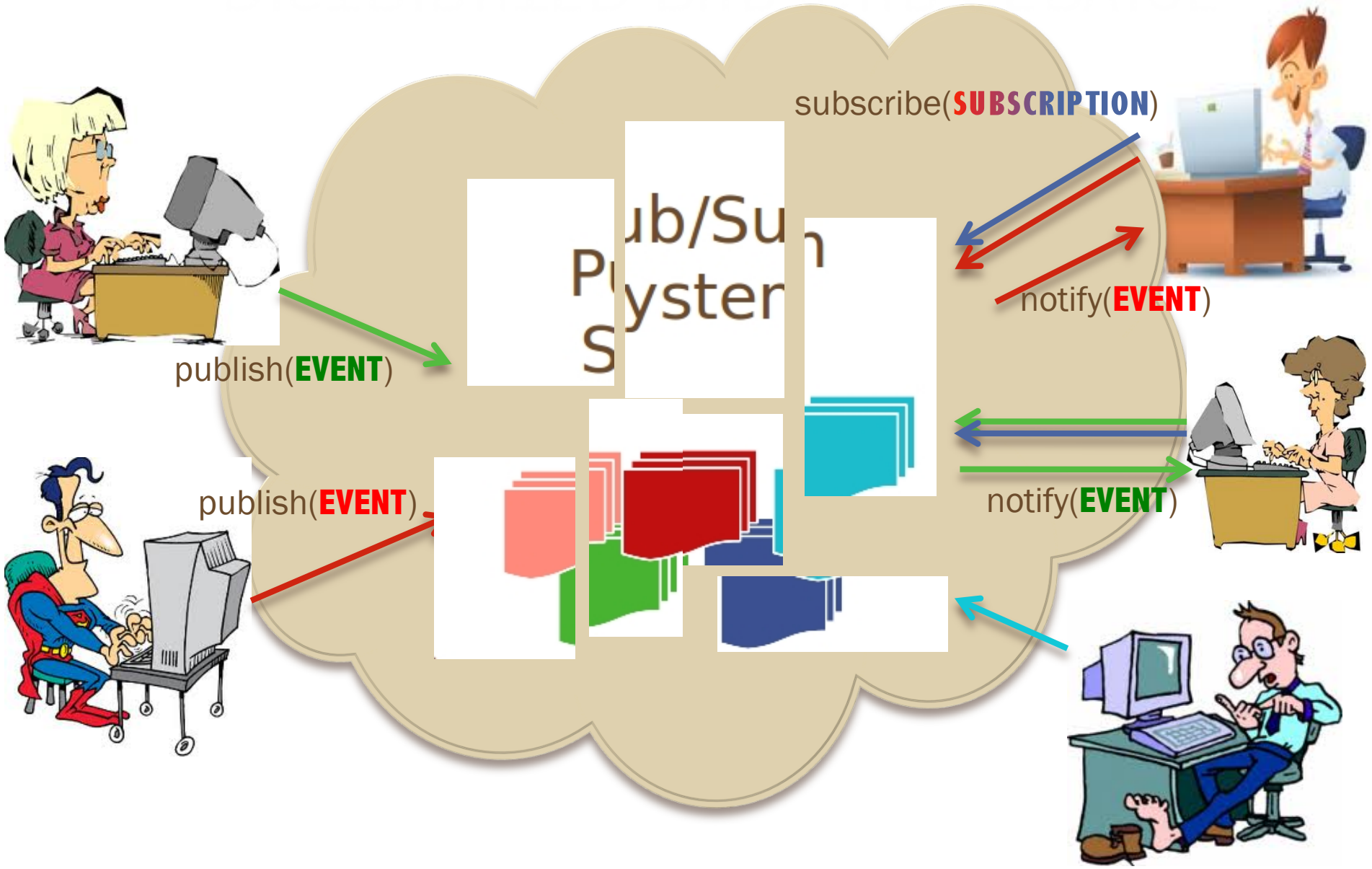
PUB/SUB PATTERN PROPERTIES

- Space decoupling
- Time decoupling
- Synchronization decoupling

SUBSCRIPTION MODELS

- Topic-based
  - Events are classified into predefined topics. Subscriptions can include any number of these topics.
    - International Film Festivals in Stockholm
    - Weather in Stockholm
- Content-based
  - Events are structured in form of multiple attributes. Subscriptions can define a range over any of these attributes.
    - Weather between -5 and 0
    - Location: Stockholm

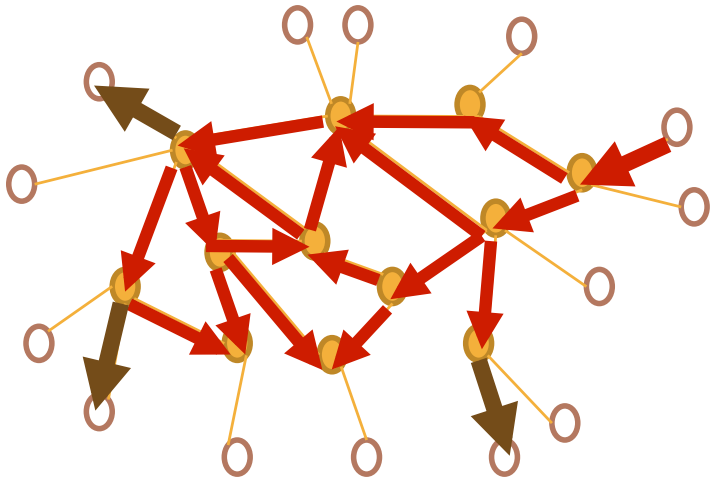
DISTRIBUTED PUB/SUB SERVICE

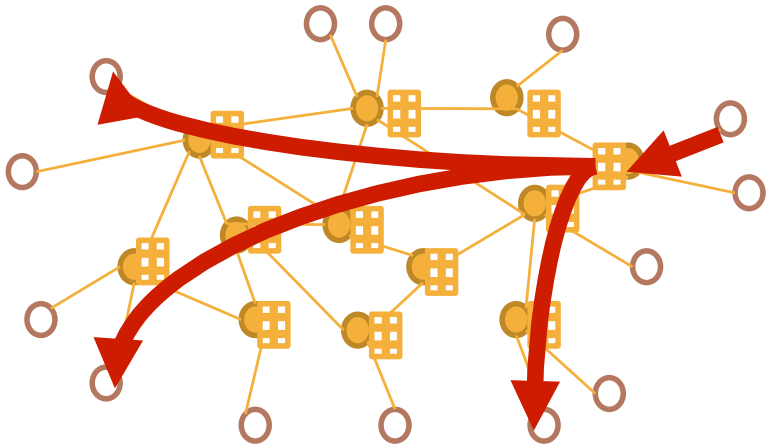


BROKER-BASED SOLUTIONS

Event Flooding

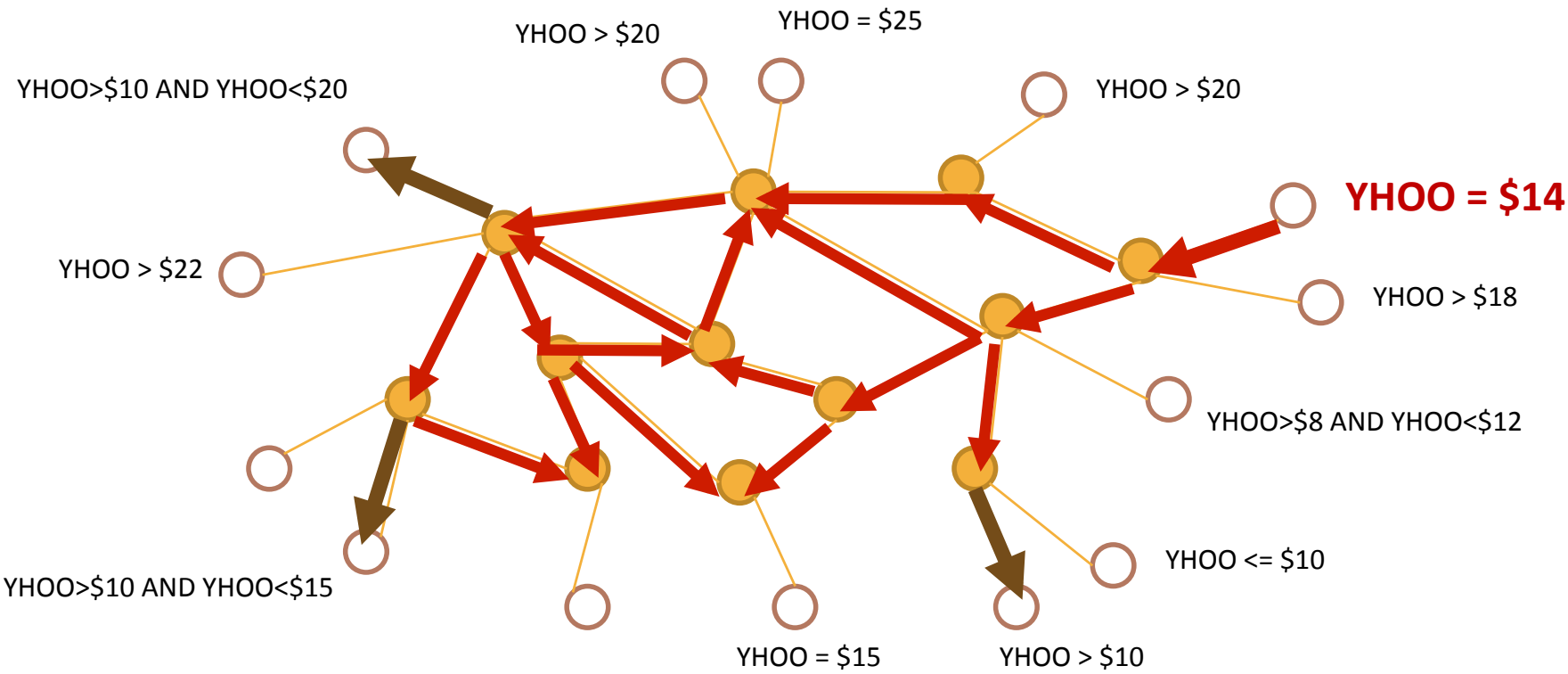
Subscription Flooding





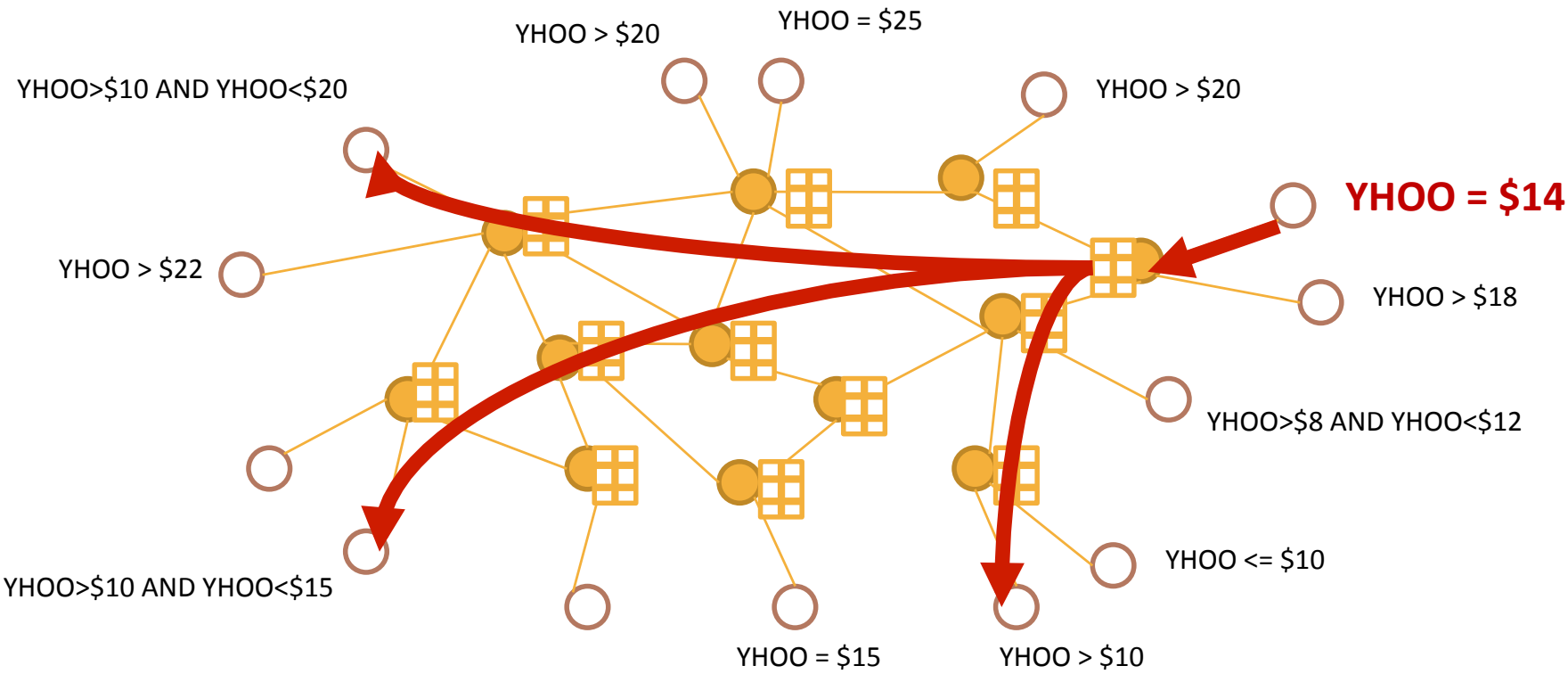
BROKER-BASED PUB/SUB

- Event Flooding



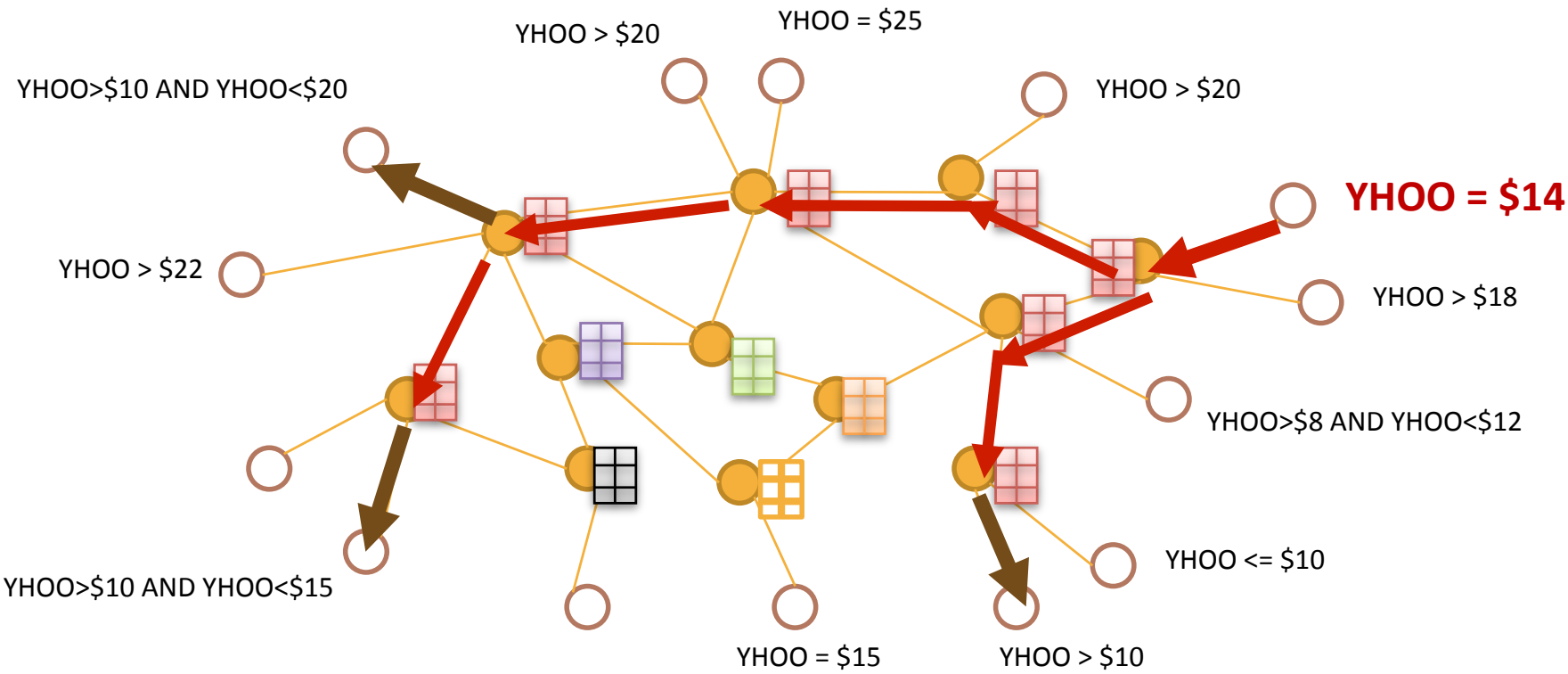
BROKER-BASED PUB/SUB

- Subscription Flooding



BROKER-BASED PUB/SUB

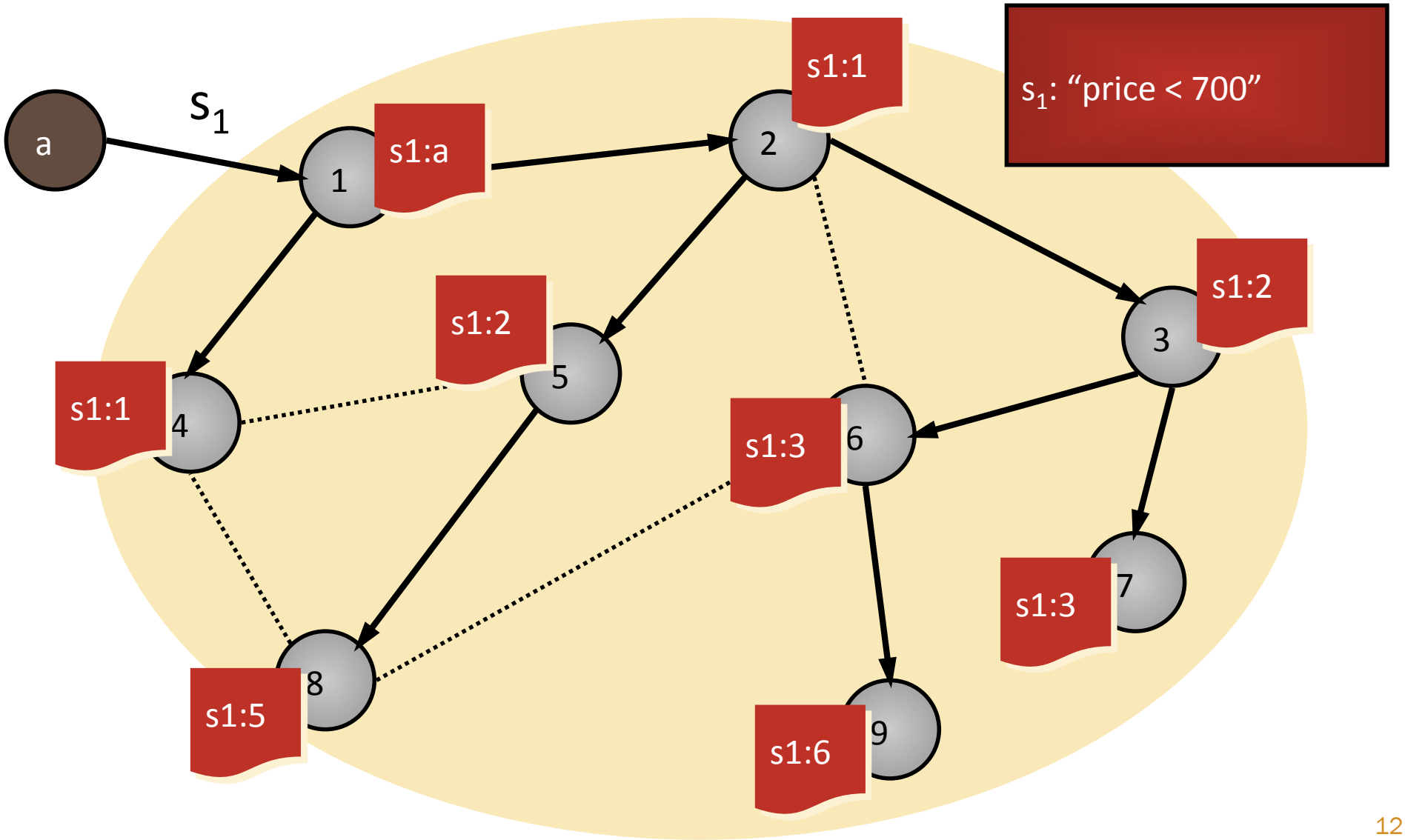
- Filter-based Routing



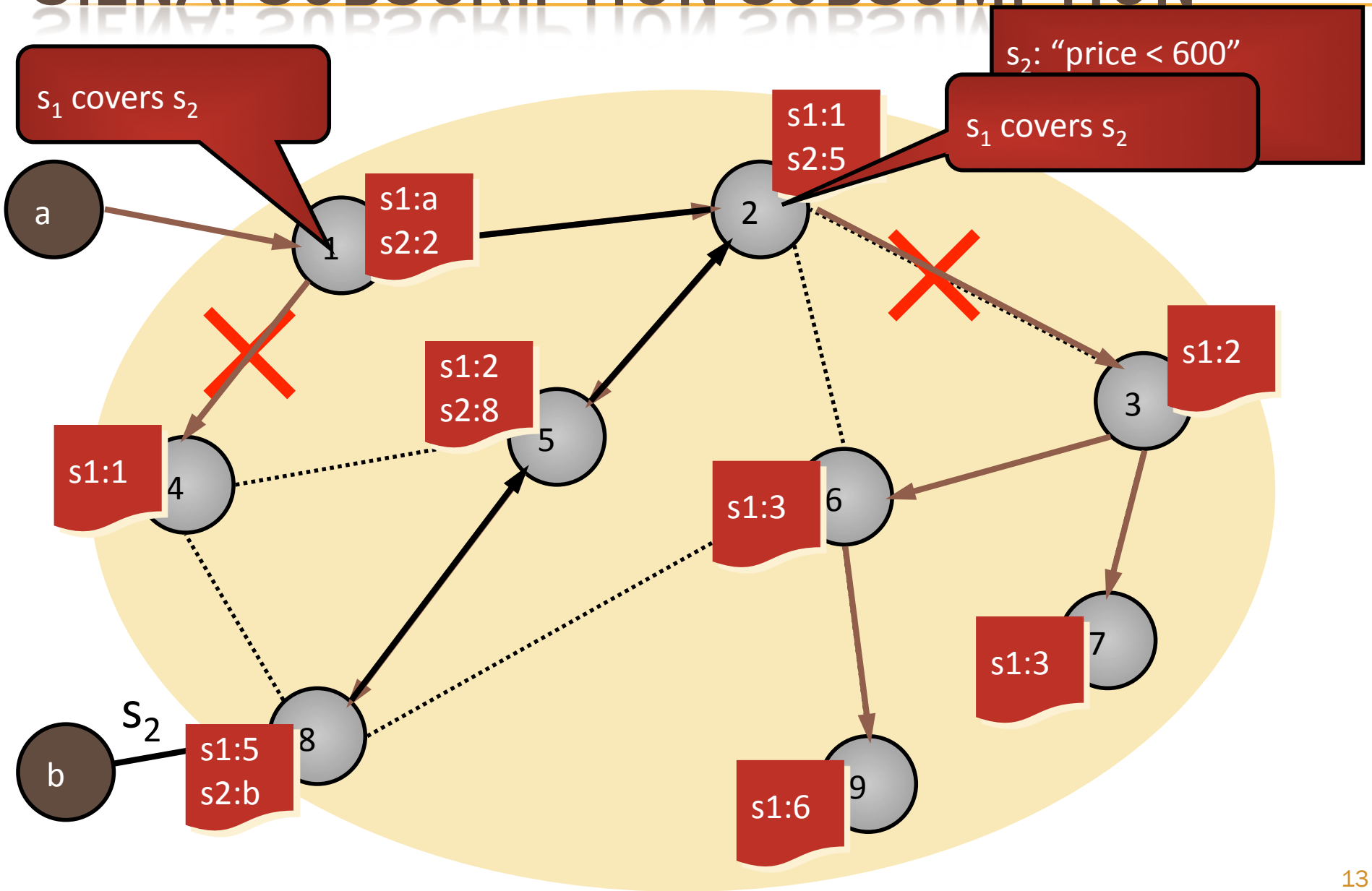
A Broker-based Content-based Pub/Sub System

SIENA

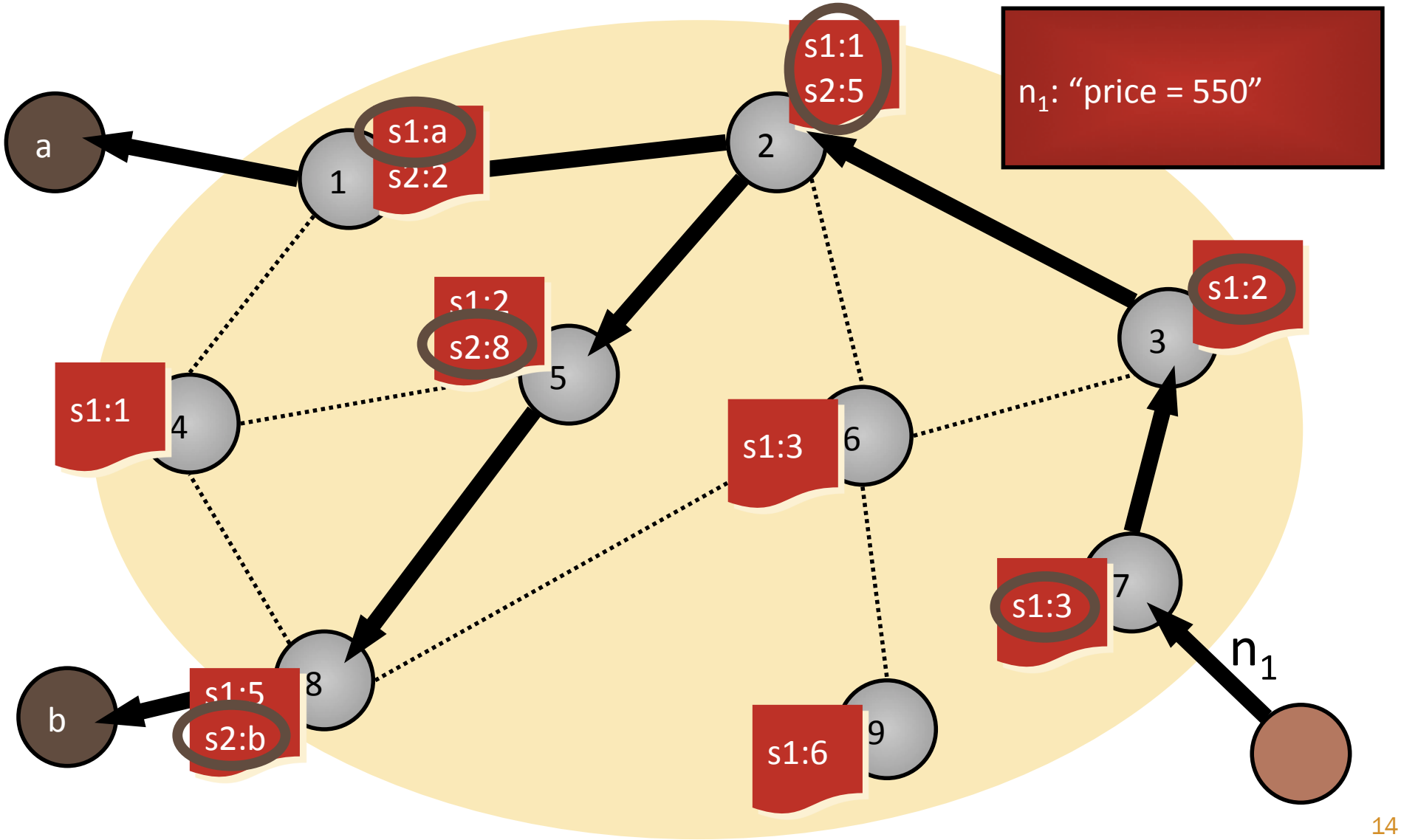
SIENA: SUBSCRIPTION FORWARDING



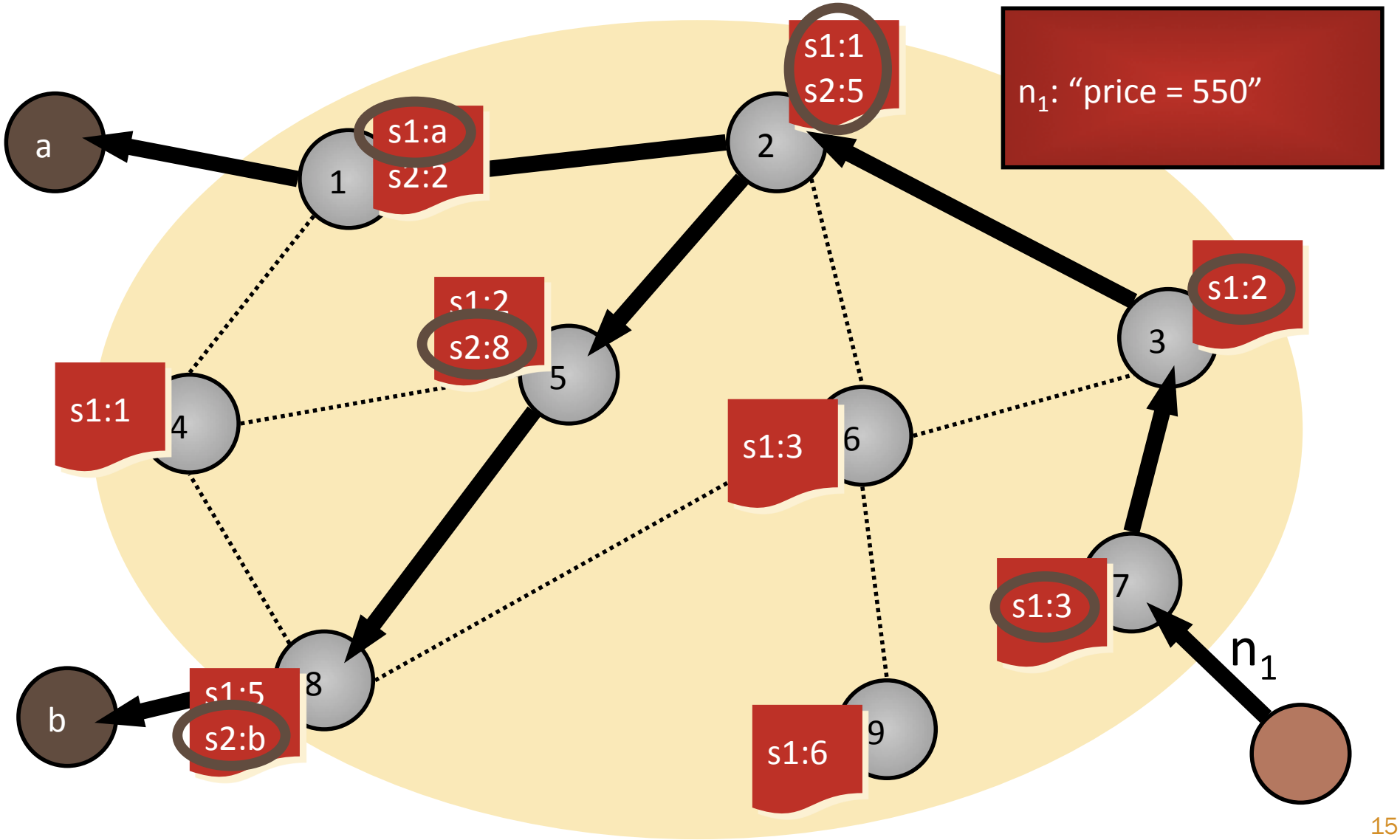
SIENA: SUBSCRIPTION SUBSUMPTION



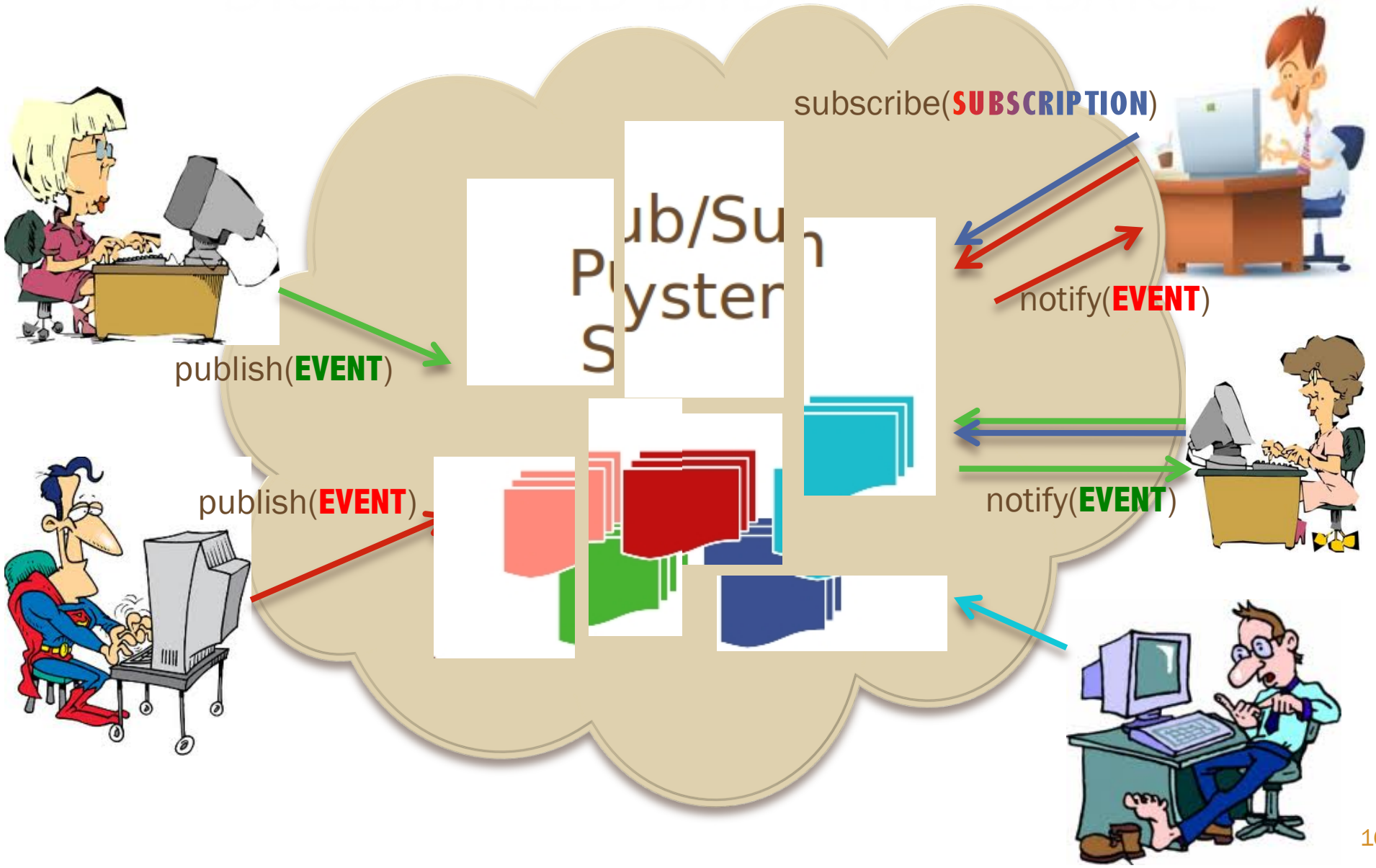
SIENA: EVENT DELIVERY



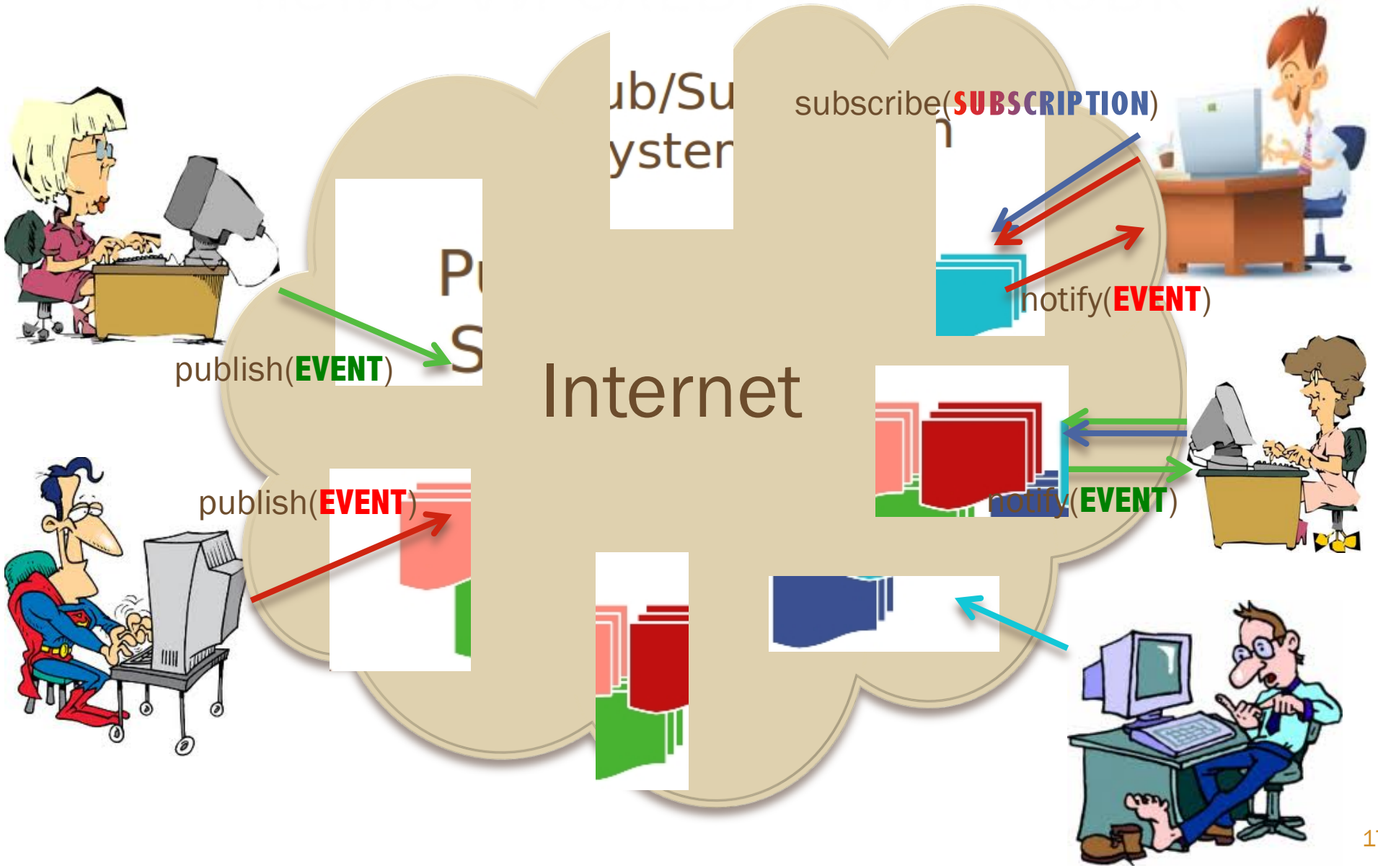
PROBLEMS?


a
1
s1:a
s2:2
2
s1:1
s2:5
n1: “price = 550”
s1:2
3
s1:2
s2:8
5
s1:1
4
s1:3
6
7
s1:3
n1
b
s1:5
s2:b
8
9
s1:6

DISTRIBUTED PUB/SUB SERVICE

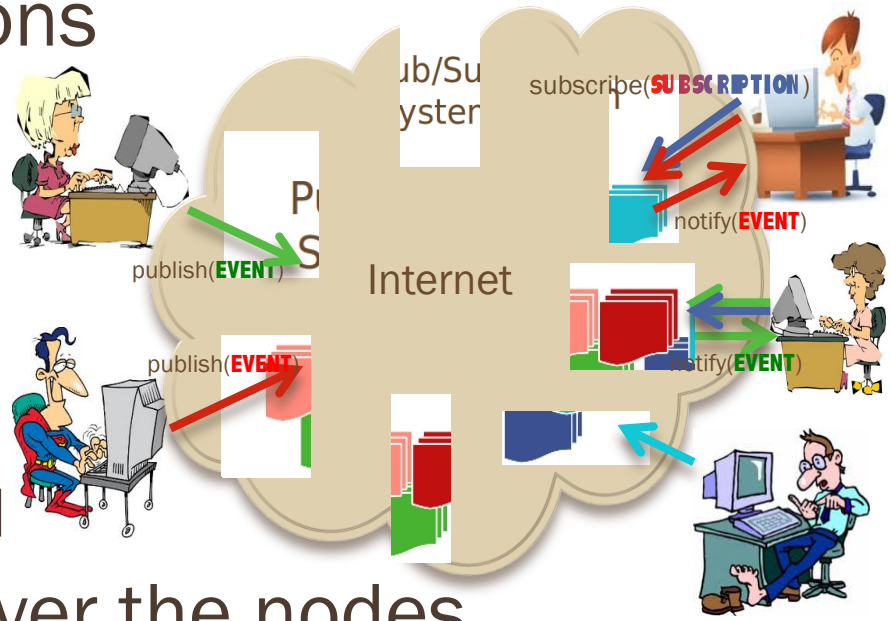


USING AN OVERLAY NETWORK



REQUIREMENTS

- Scale to large network sizes
- Guarantee event delivery
- Handle massive publications
- Tolerate failures
- Provide fast delivery
- Generate low overhead
  - Maintenance overhead
  - Data propagation overhead
- Distribute the load fairly over the nodes

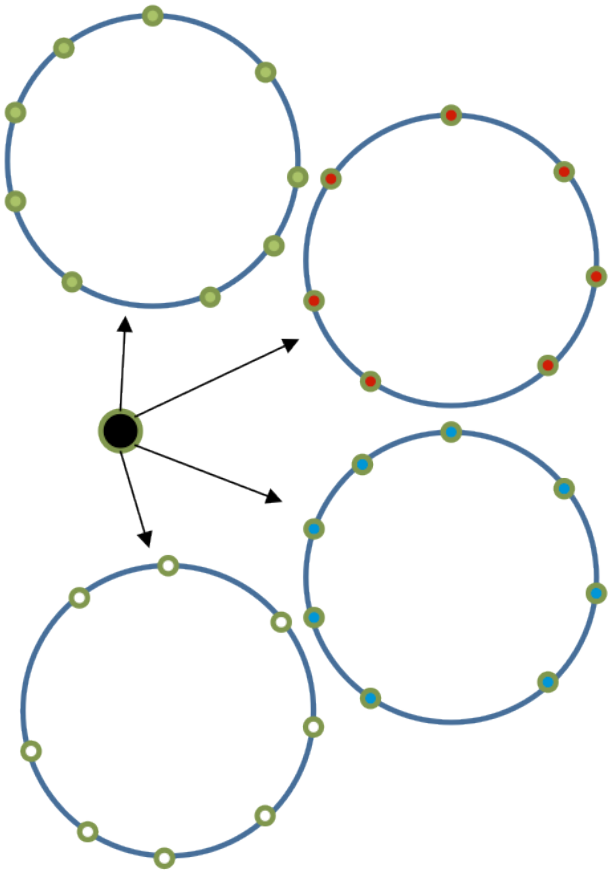


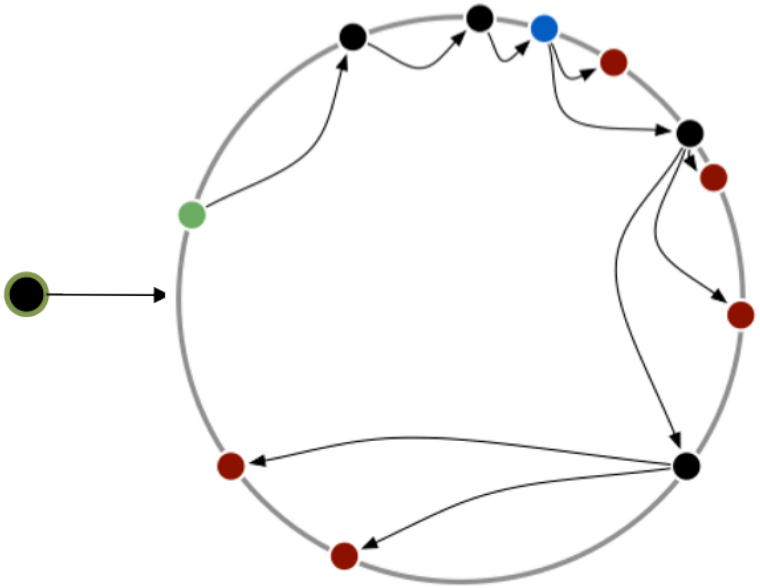
P2P TOPIC-BASED PUB/SUB SYSTEMS

THE RANGE OF TOPIC-BASED SOLUTIONS

An Overlay Per Topic

Rendezvous Routing



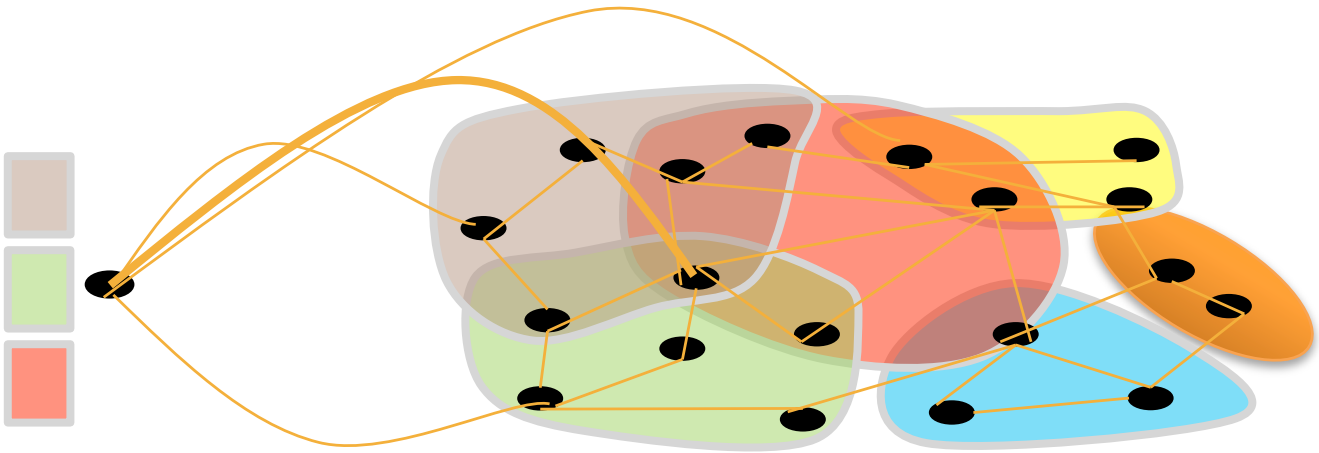


A Topic-based Pub/Sub System

TERA

OVERLAY-PER-TOPIC PUB/SUB

- Overlay is topic-connected if the sub-graph induced by every topic is connected

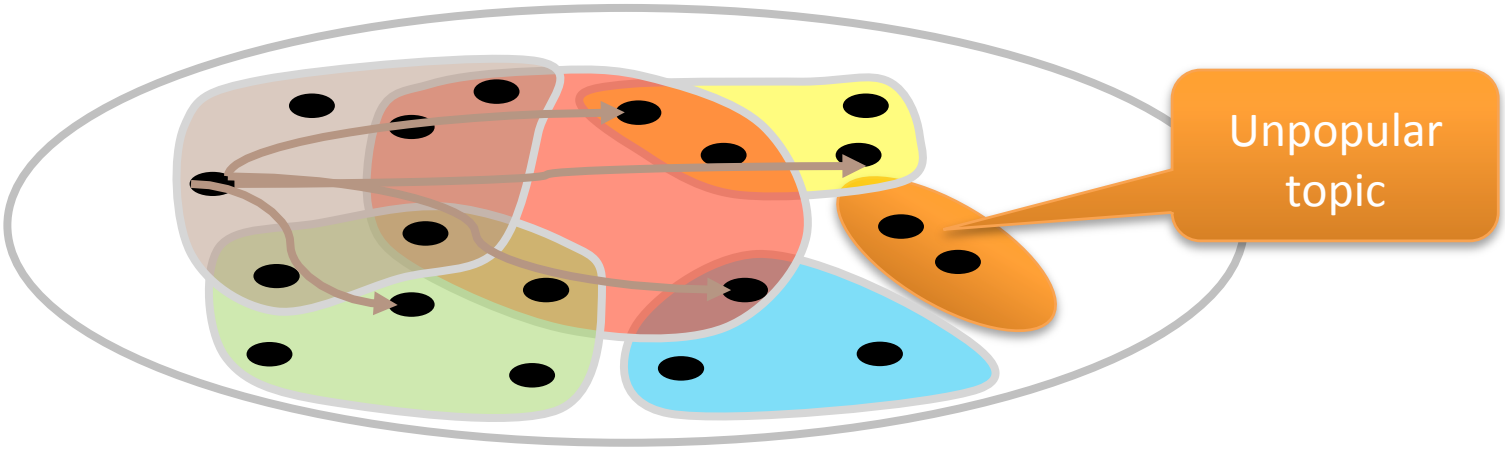


TERA

- Baldoni et al. “TERA: topic-based event routing for peer-to-peer architectures”, 2007
  - Publishers do not have to be interested in the topic they are publishing
  - The basic requirements:
    - Interest Clustering
    - Inner-Cluster Dissemination
    - *Outer-Cluster Dissemination*

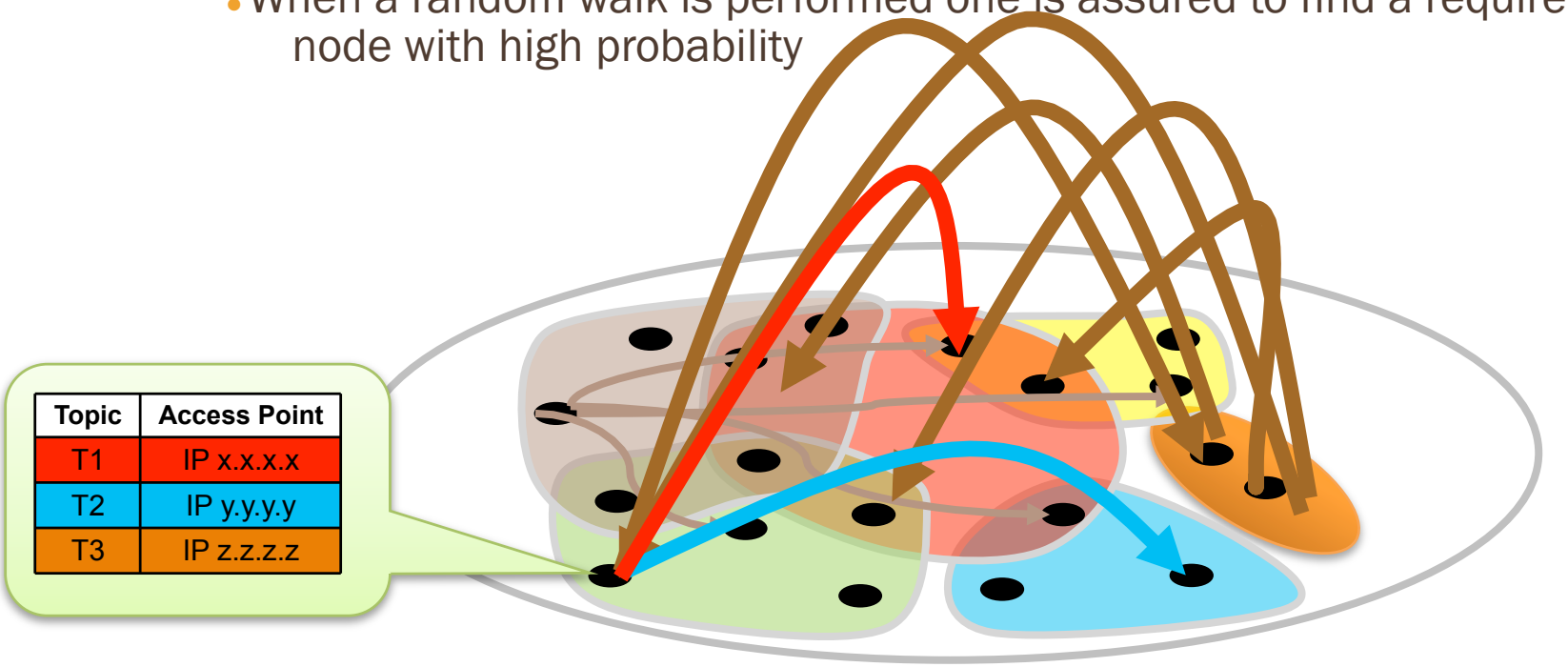
SYSTEM OVERVIEW

- Interest Clustering
  - Instantiates a dedicated overlay network per topic
- Inner-Cluster Dissemination
  - Uses a simple flooding scheme (which can be replaced by more sophisticated routing algorithms, if required)
- How to find a specific cluster in a decentralized fashion (Outer-Cluster Dissemination)?
  - *Random walks* in the network
  - What is the probability of finding?
  - What if a cluster (topic) is very unpopular?



OUTER CLUSTER ROUTING

- Solution:
  - Every topic advertizes itself on the random nodes in the network (via gossiping)
  - BUT (!) with the *probability inversely proportional to the size of the cluster* (topic population)
  - Every node keeps *Access Point Table* with pointers to *access points* to different clusters (topics)
  - When a random walk is performed one is assured to find a required access node with high probability



OUTER CLUSTER ROUTING: AN EXAMPLE

- Populating APT:
  - Node X contacts node Y
- Outer Cluster Publishing

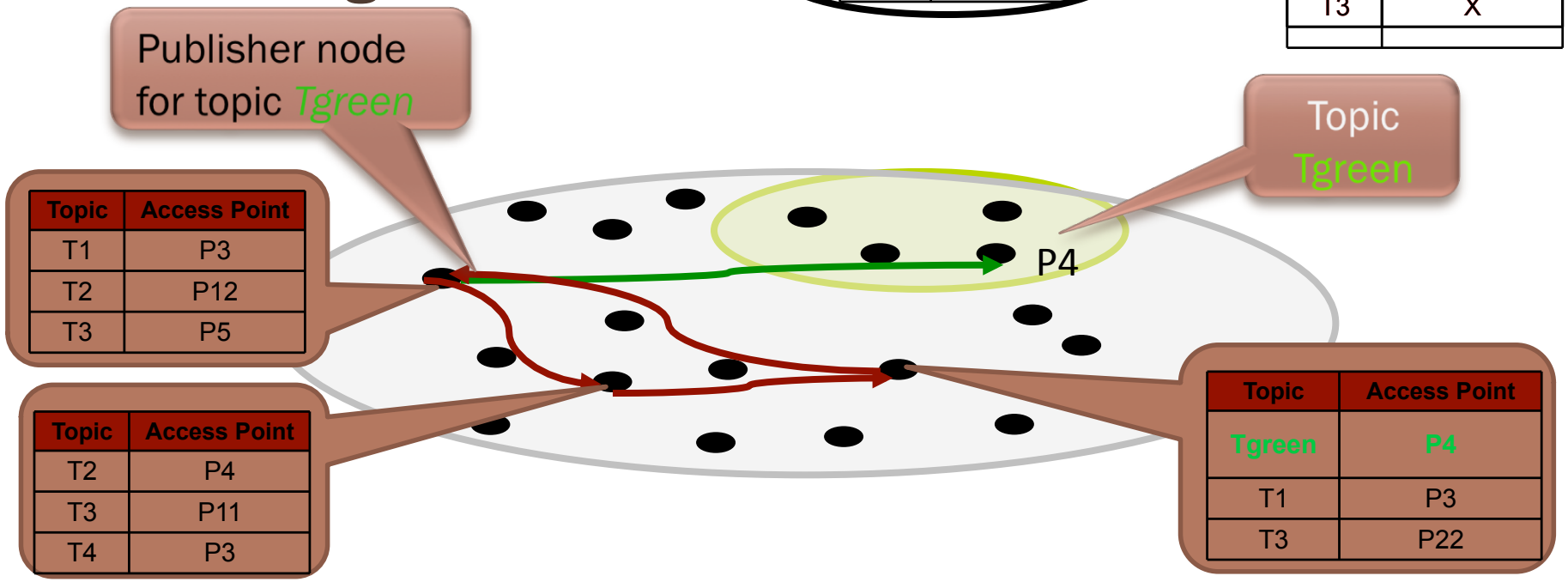
Subscriptions of node X

Topic	Popularity
T2	456
T3	123
T4	678
T5	15

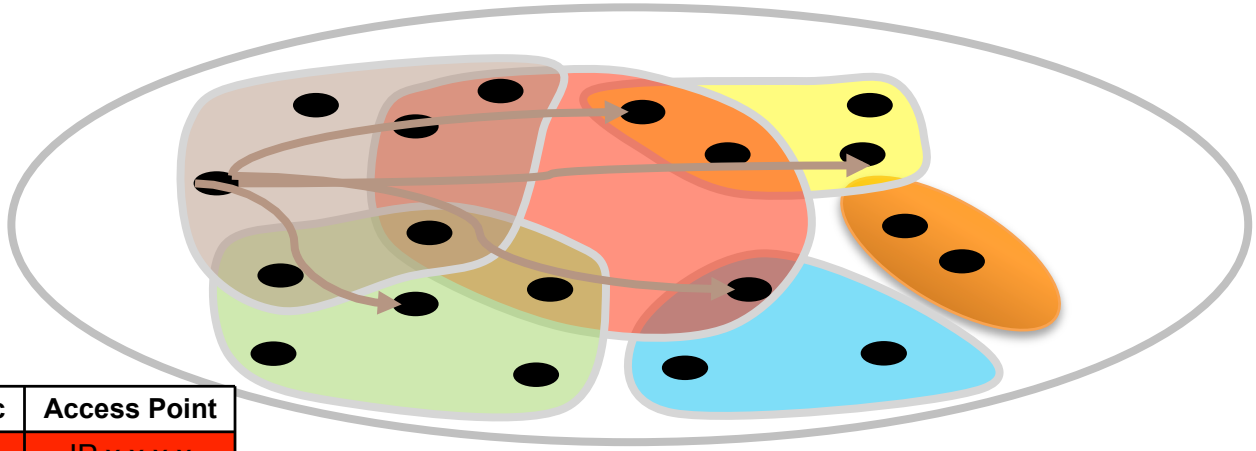
APT in node Y

Topic	Access Point
T1	Q
T2	R

Topic	Access Point
T1	Q
T2	X
T3	X



PROBLEMS?



Topic	Access Point
T1	IP x.x.x.x
T2	IP y.y.y.y
T3	IP z.z.z.z

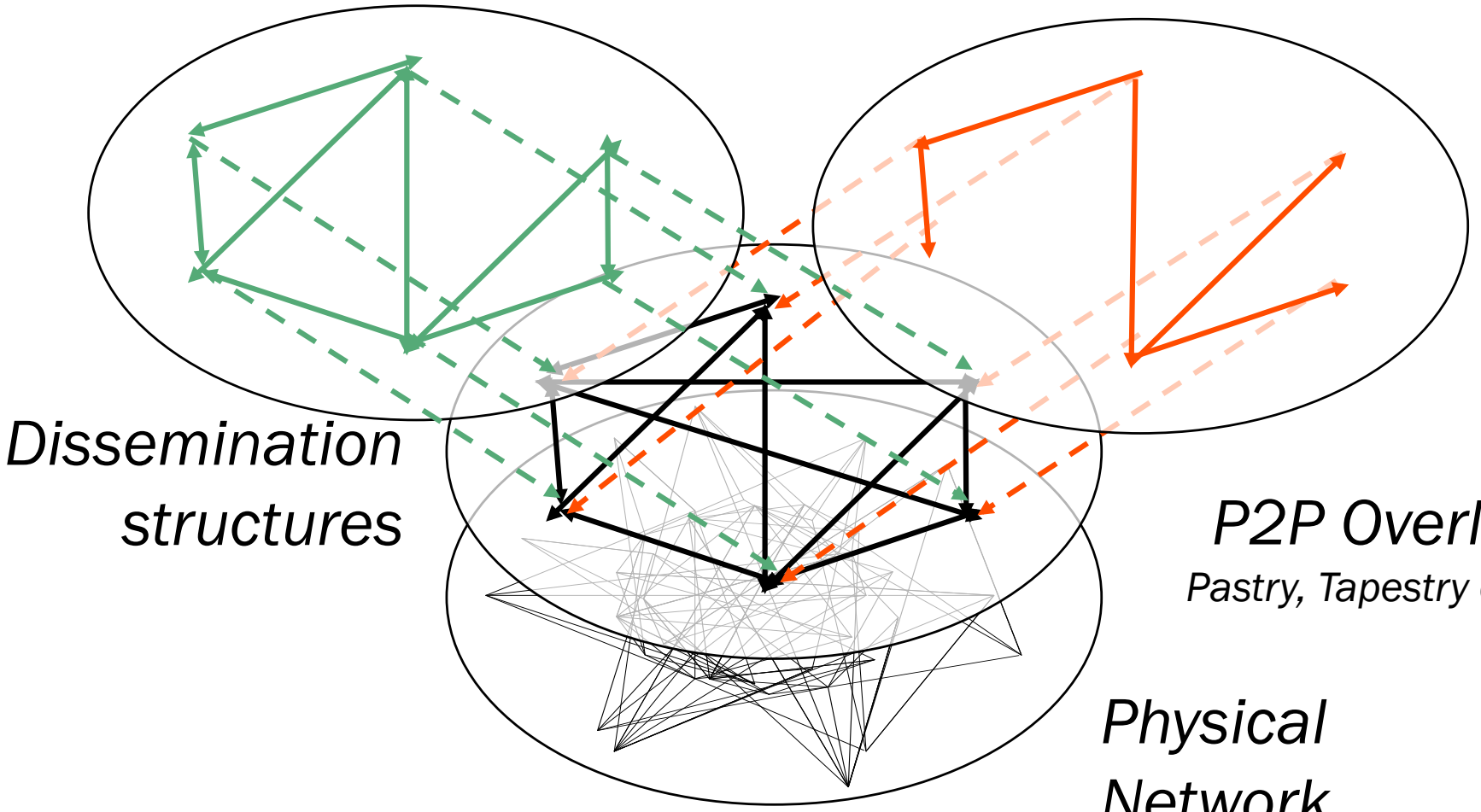
SCRIBE

RENDEZVOUS BASED PUB/SUB

- Castro et al. "*SCRIBE: A large-scale and decentralized application-level multicast infrastructure*", *2002*

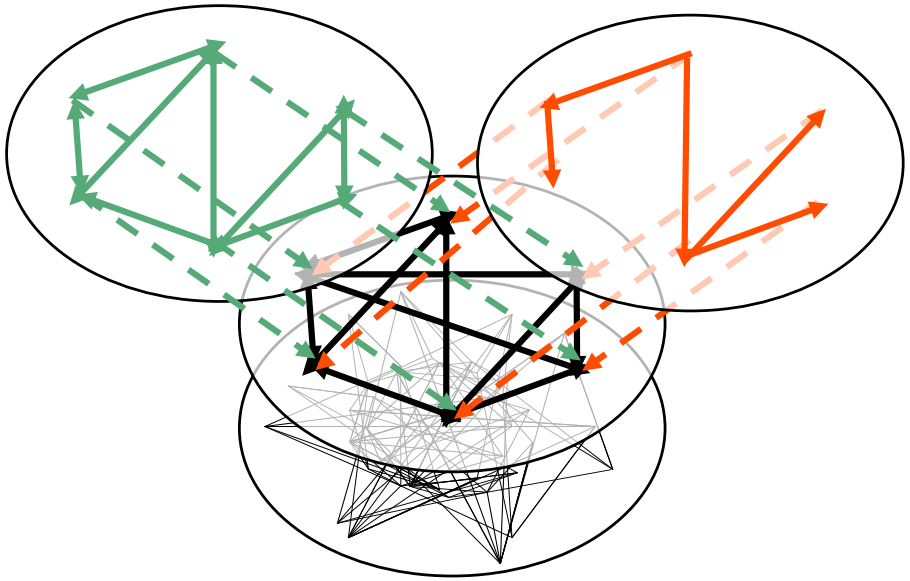
- Zhuang et al. "*Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination*", *2001*

SCRIBE: ON TOP OF A STRUCTURED OVERLAY



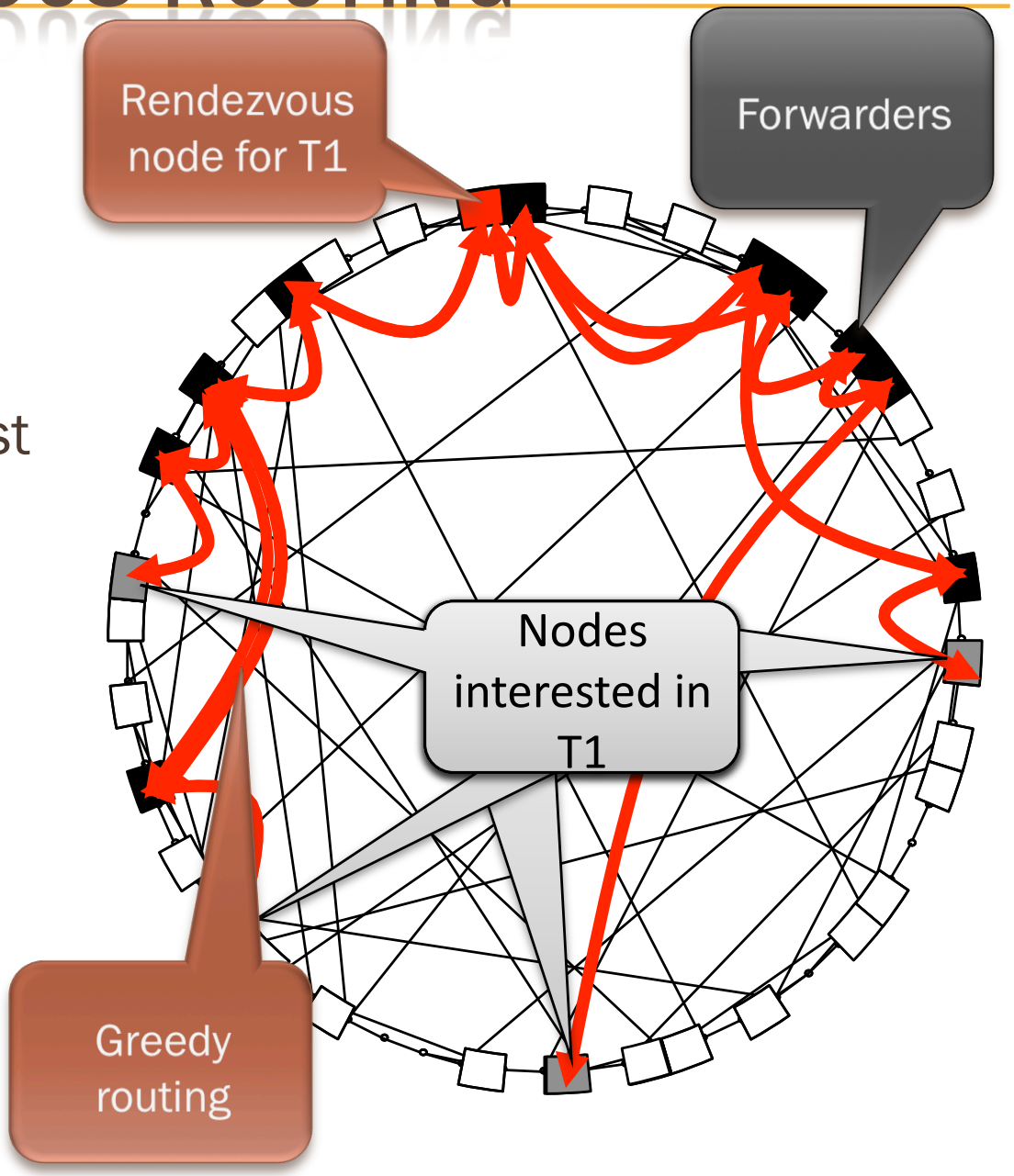
SCRIBE: ON TOP OF A STRUCTURED OVERLAY

- Why structured overlays for pub/sub?
  - Decentralized construction
  - Scalability, Connectivity, Low diameter
  - Node degree is not blown up
- Scribe is based on Pastry Overlay



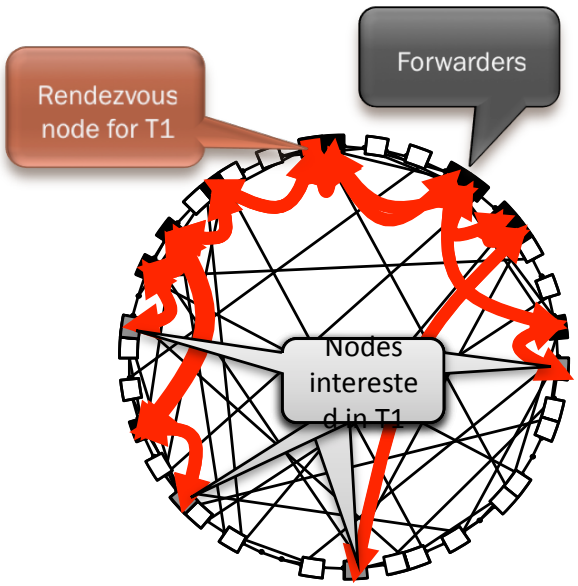
SCRIBE: RENDEZVOUS ROUTING

- Each group has a unique group-Id.
- The Scribe node with a node-Id numerically closest to the group-Id acts as the rendezvous point for the associated group.



SCRIBE: THE MULTICAST TREES

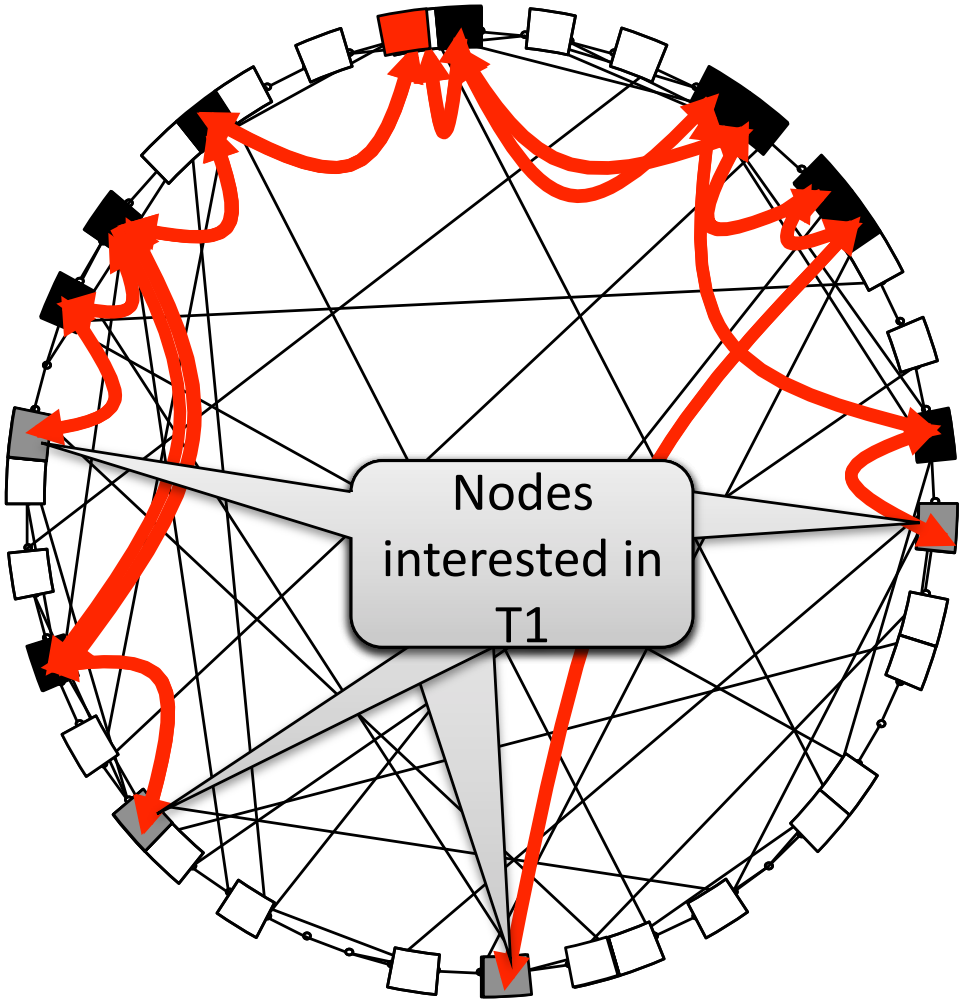
- The rendezvous point is the root of the multicast tree created for the group.
- Forwarders may or may not be members of the group.
  - Each forwarder maintains a children table for each group
- The properties of Pastry routes ensure that this mechanism produces a tree without the loops



REPAIRING MULTICAST TREES

- Exchange heartbeat messages
- A child suspects that its parent is faulty when it fails to receive heartbeat messages
- Upon detection of the failure of its parent, a node rejoins the tree.
  - The node calls Pastry to route a JOIN message to the group's identifier.
  - Pastry will route the message to a new parent, thus repairing the multicast tree.

PROBLEMS?

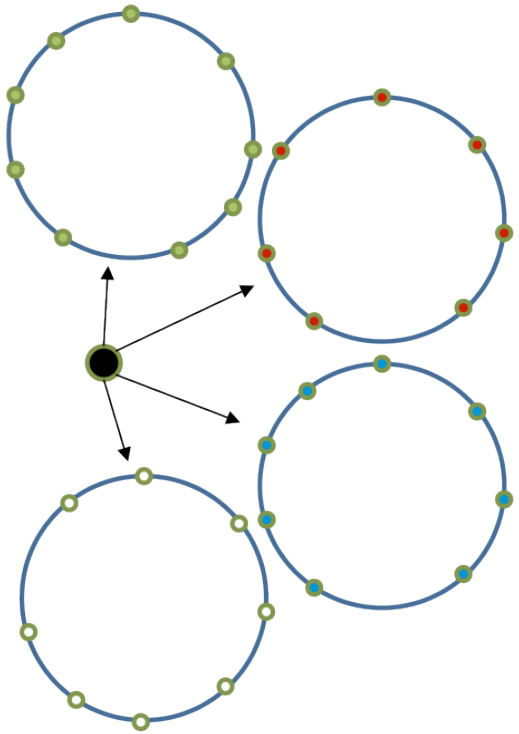


THE RANGE OF TOPIC-BASED SOLUTIONS

An Overlay Per Topic

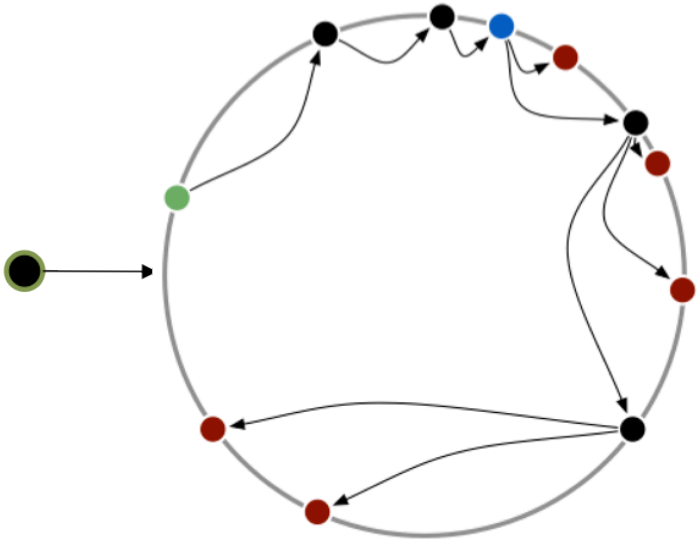
Rendezvous Routing

Unbounded node degree



Huge traffic overhead

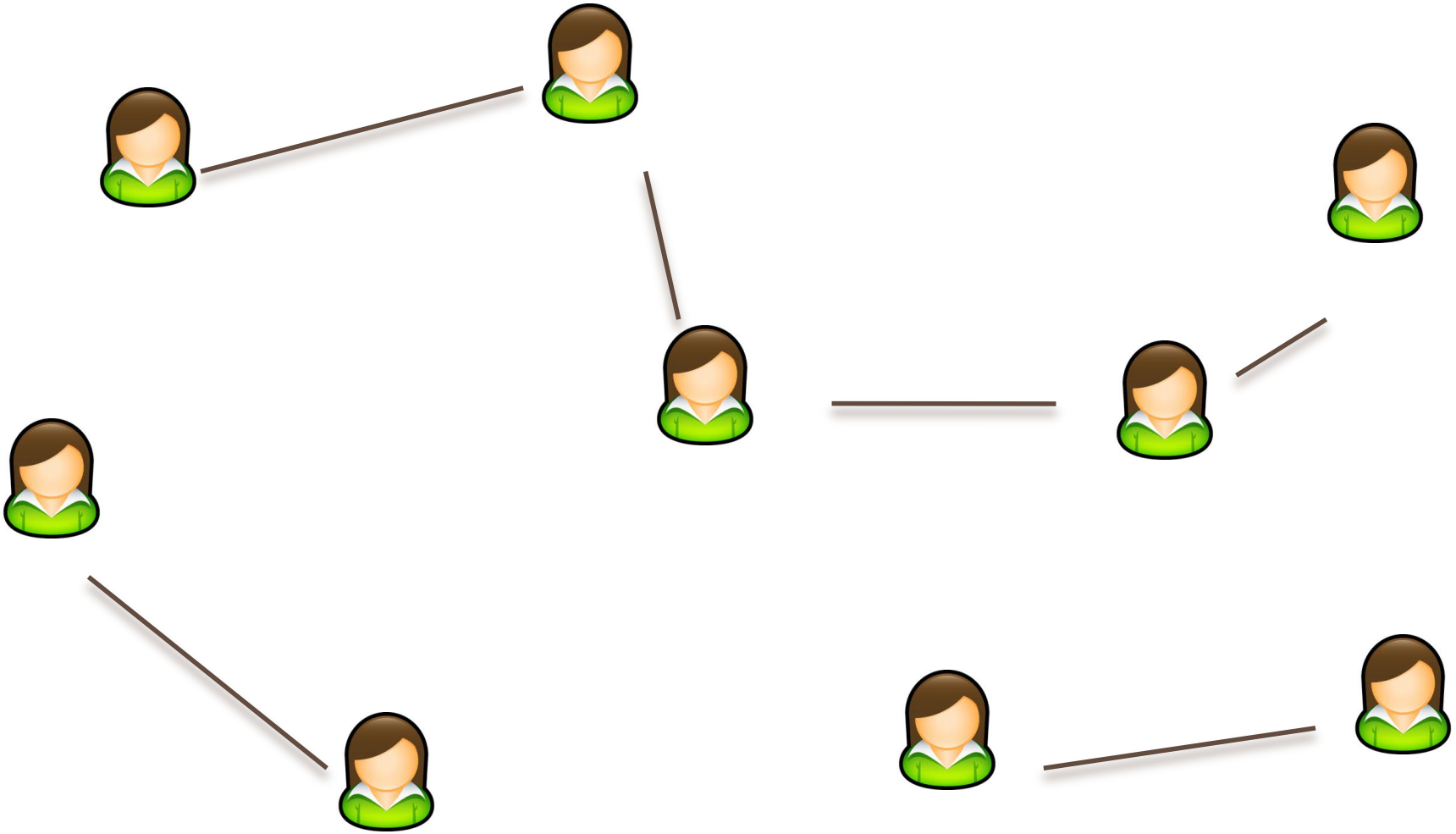
Unbalanced load



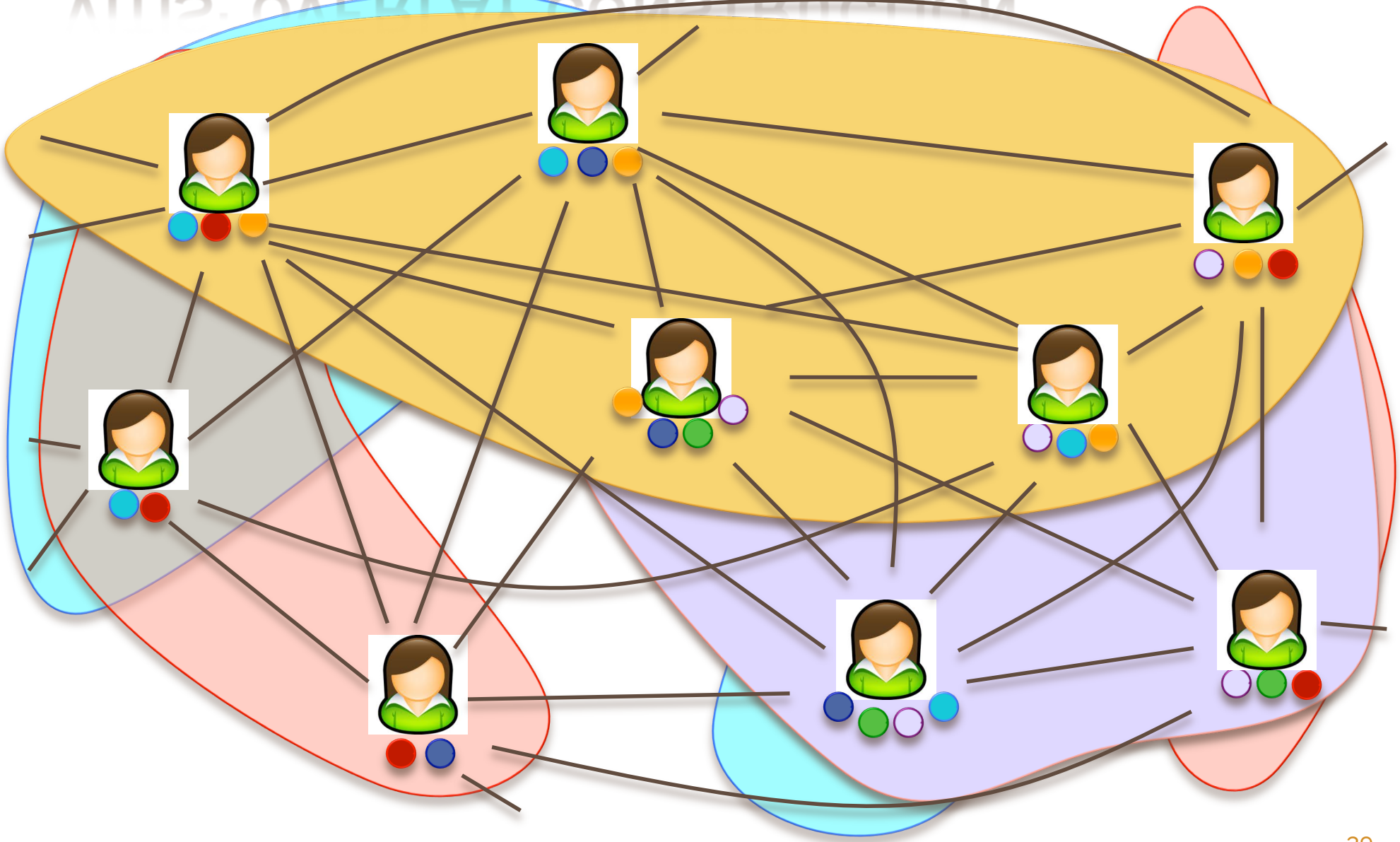
A Gossip-based Hybrid Overlay for Topic-based Pub/Sub

VITIS

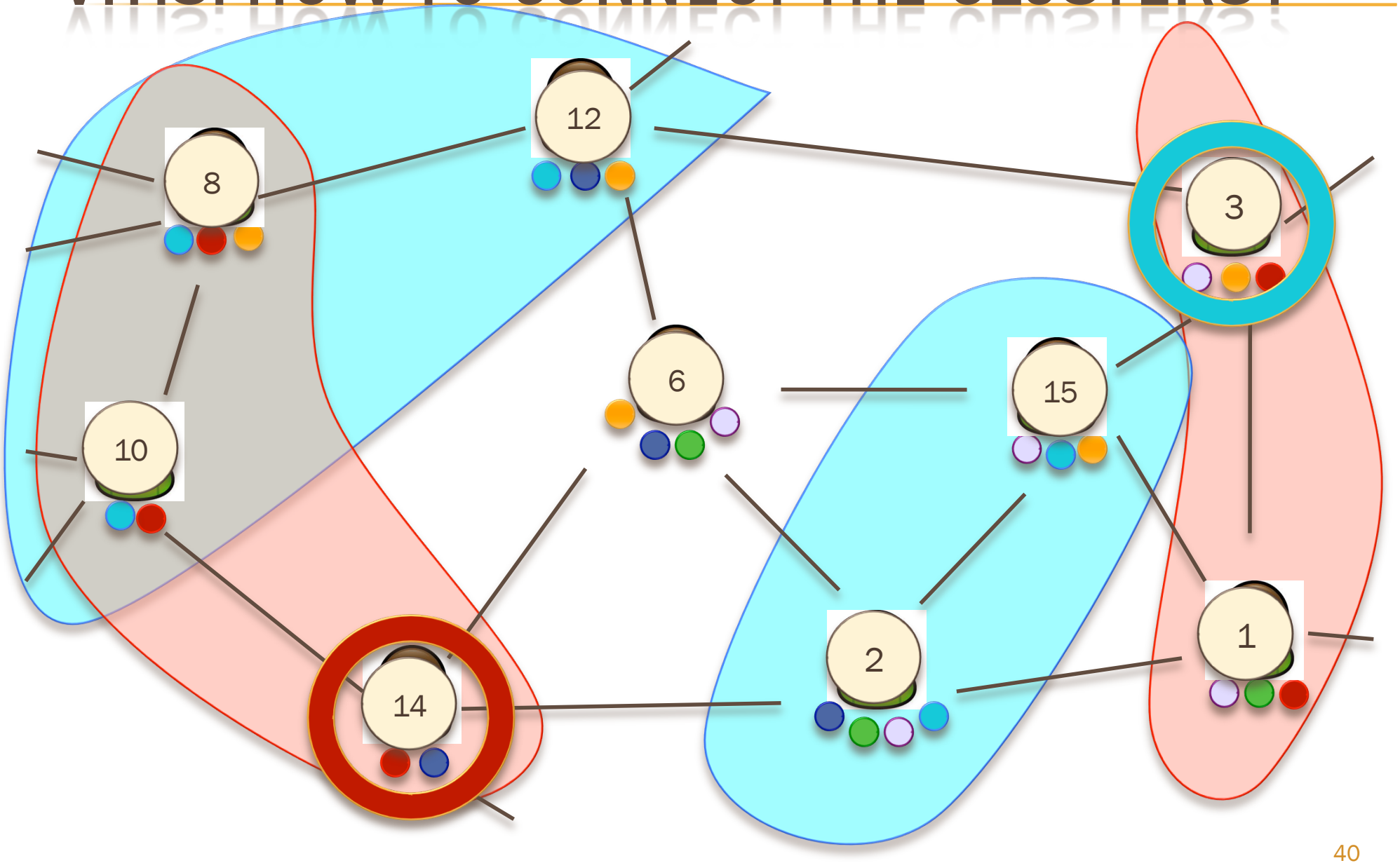
VITIS: GOSSIPING



VITIS: OVERLAY CONSTRUCTION

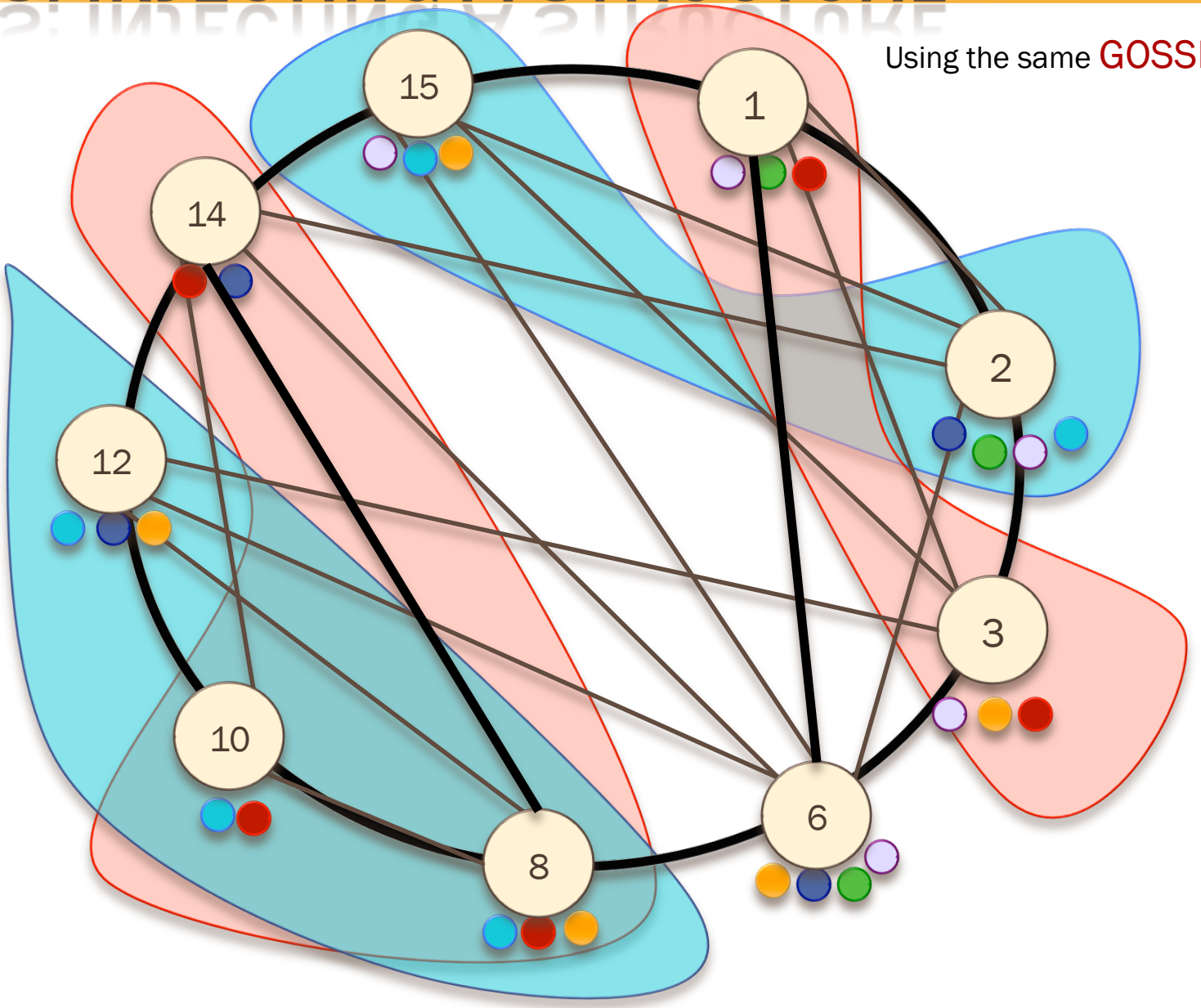


VITIS: HOW TO CONNECT THE CLUSTERS?

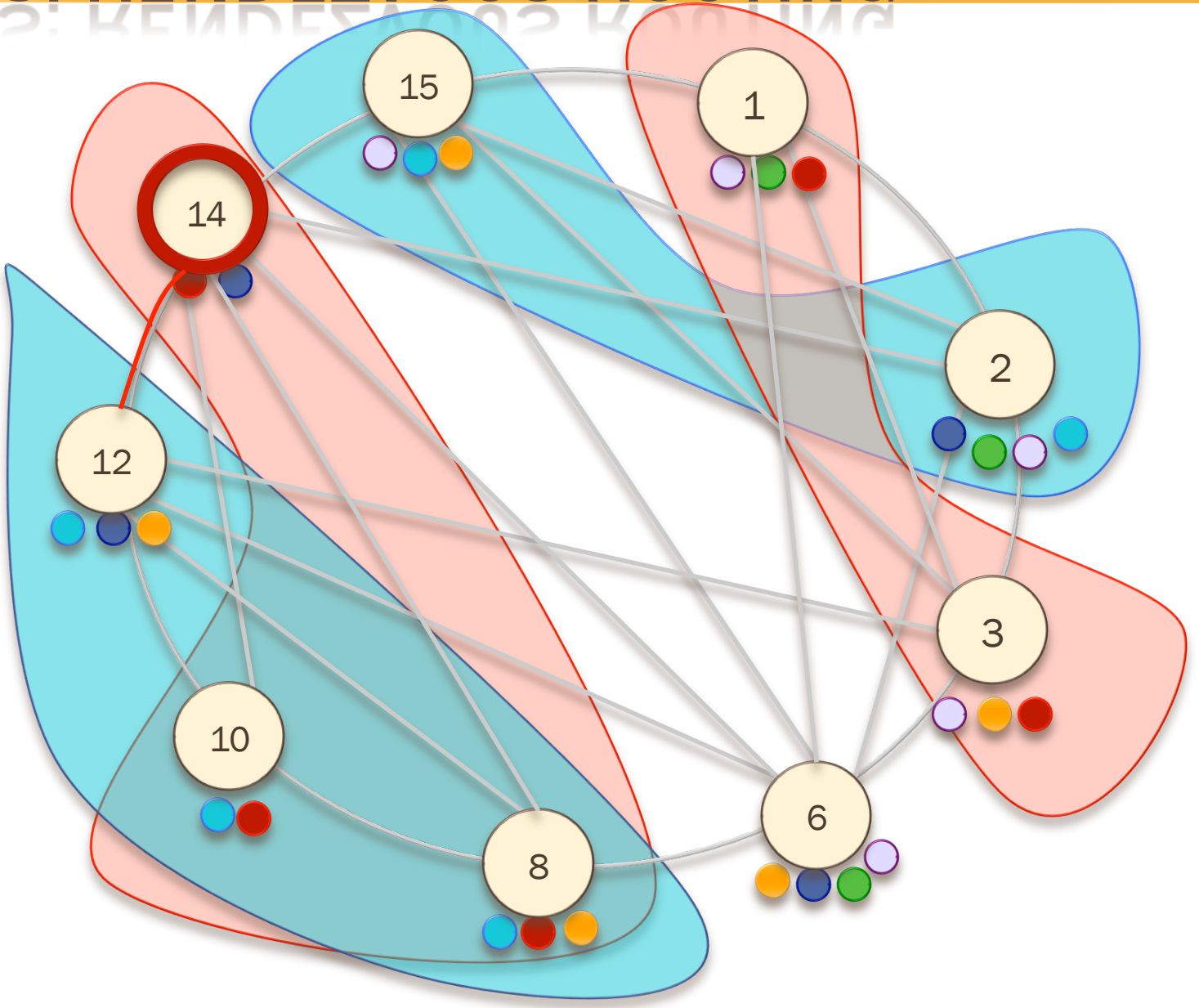


VITIS: INJECTING A STRUCTURE

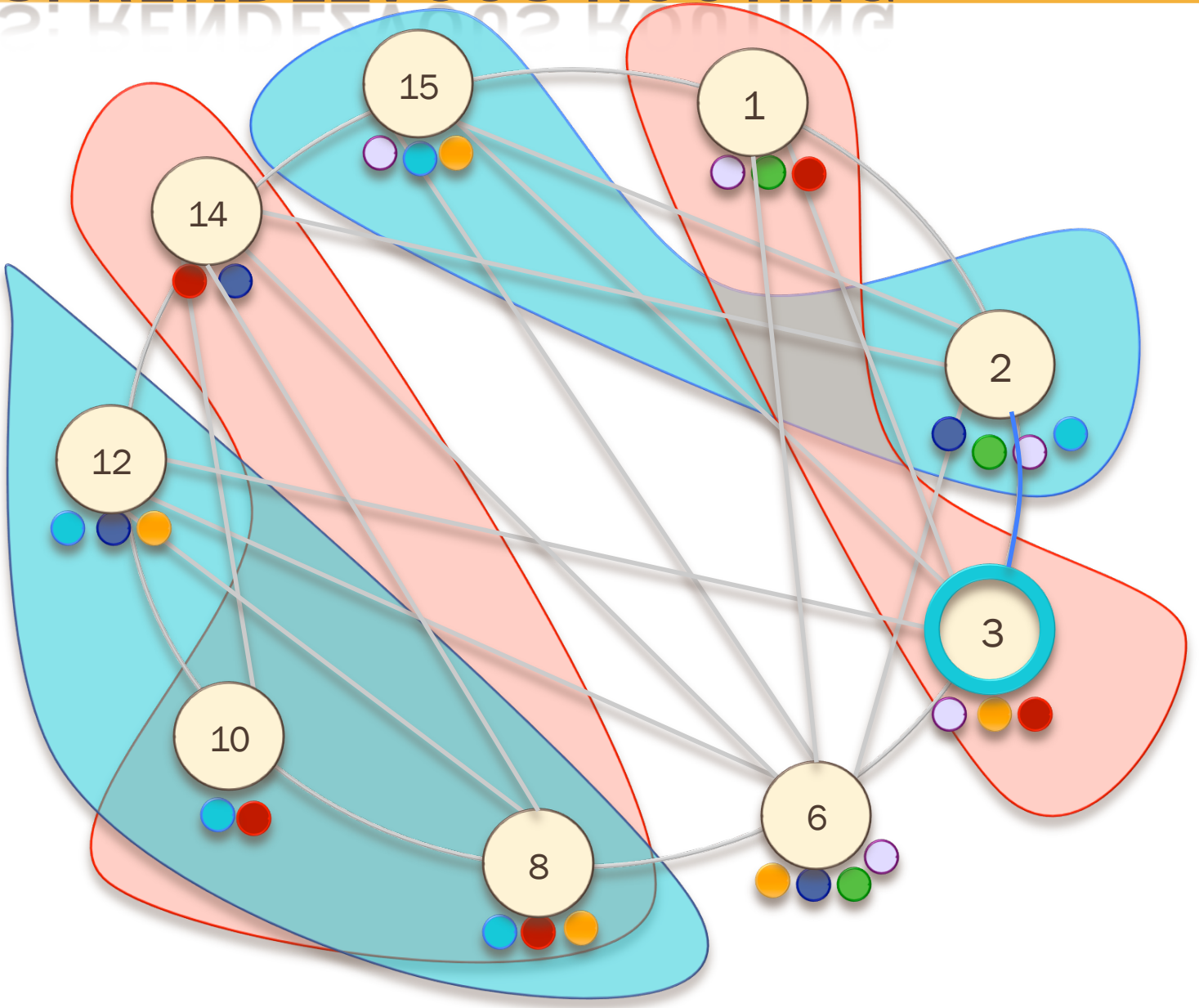
Using the same GOSSIPING protocol



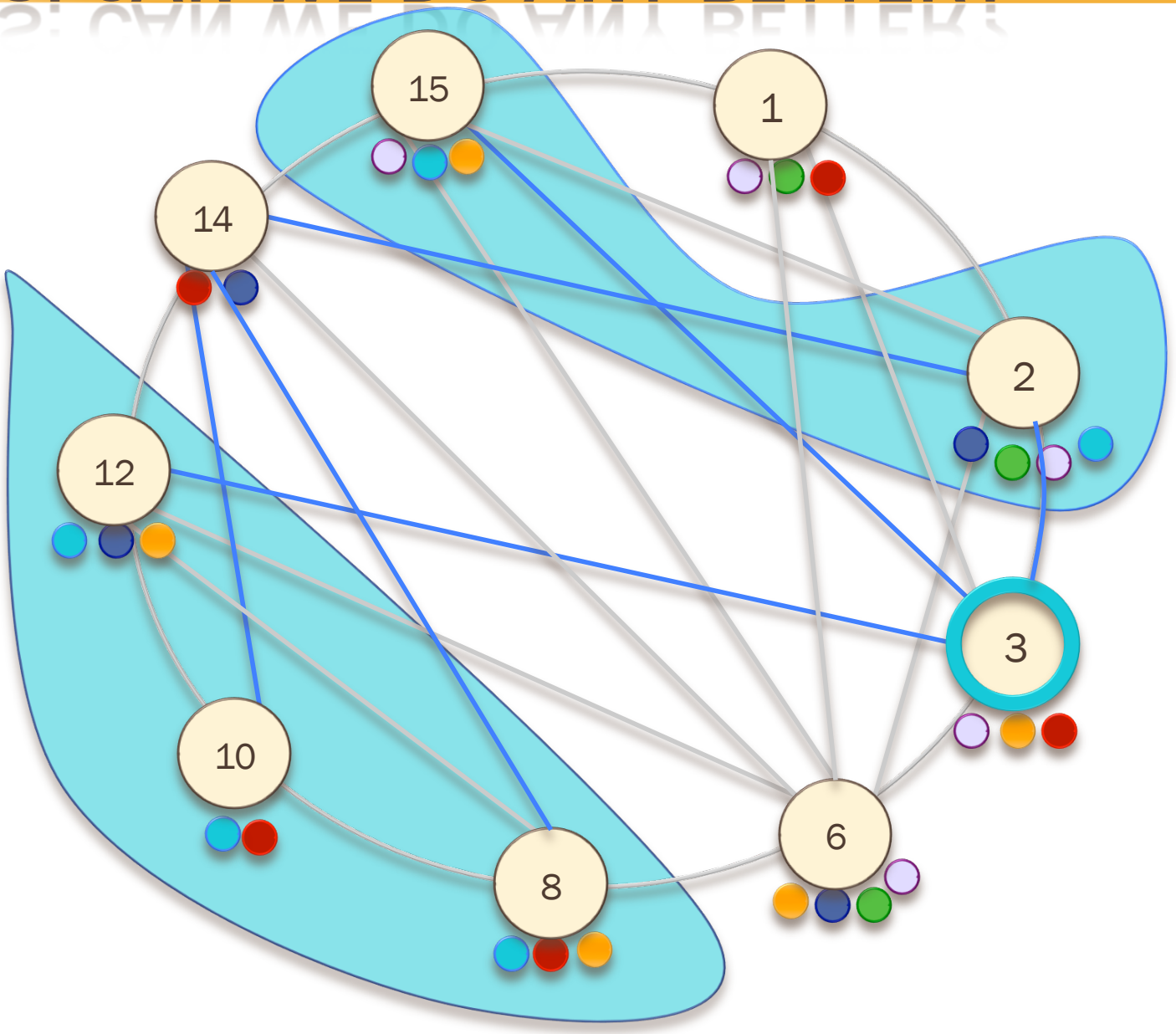
VITIS: RENDEZVOUS ROUTING



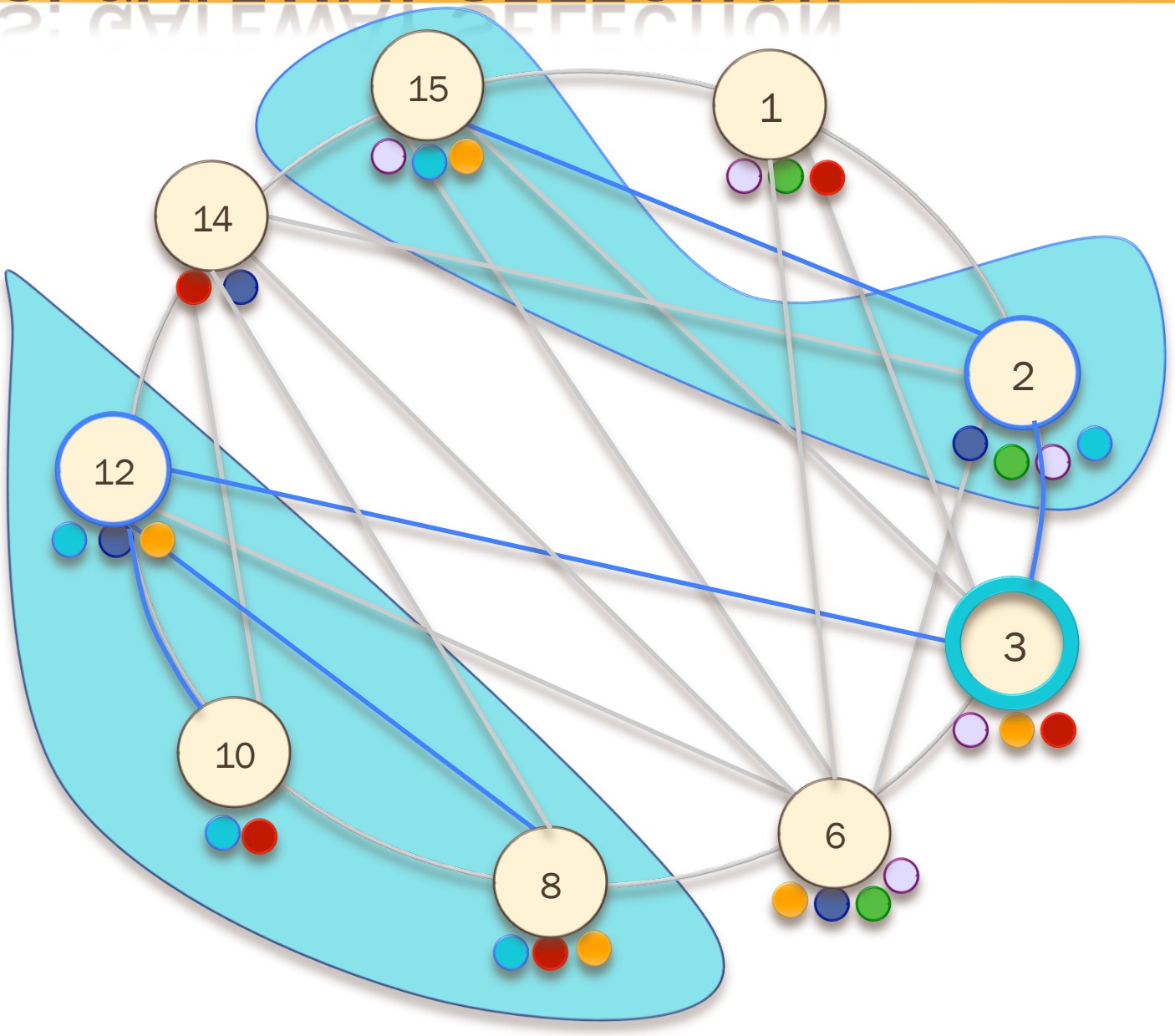
VITIS: RENDEZVOUS ROUTING



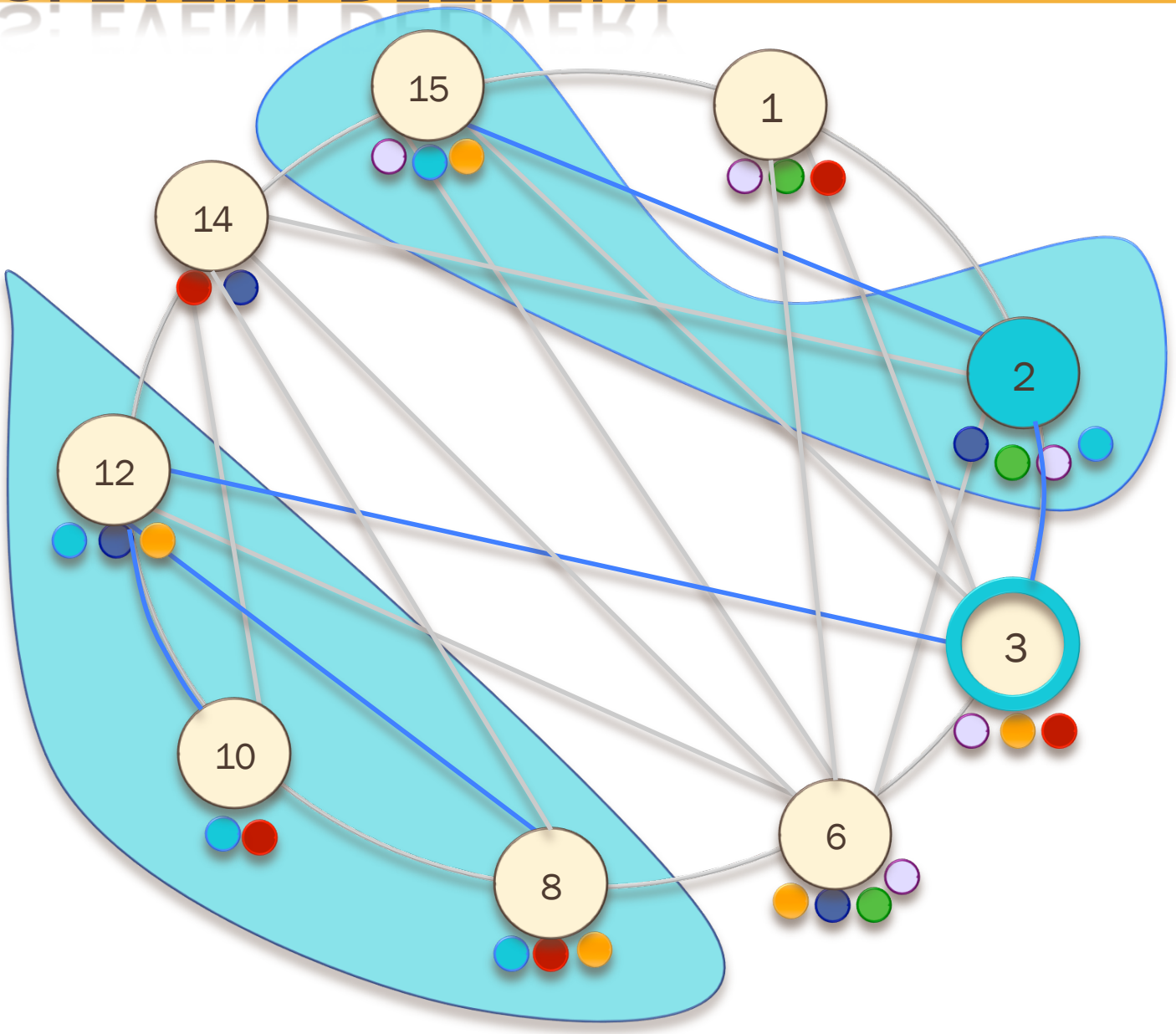
VITIS: CAN WE DO ANY BETTER?



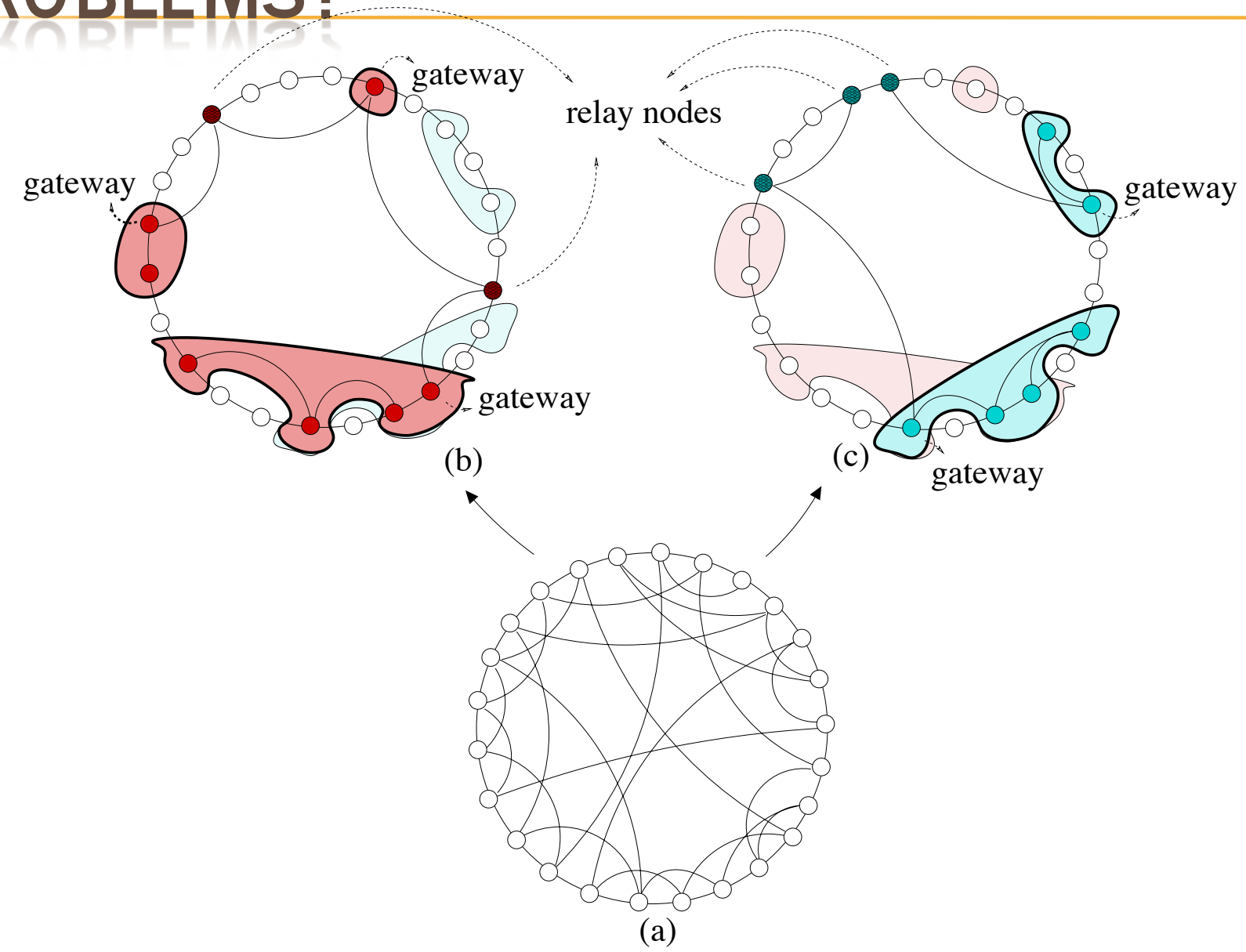
VITIS: GATEWAY SELECTION



VITIS: EVENT DELIVERY



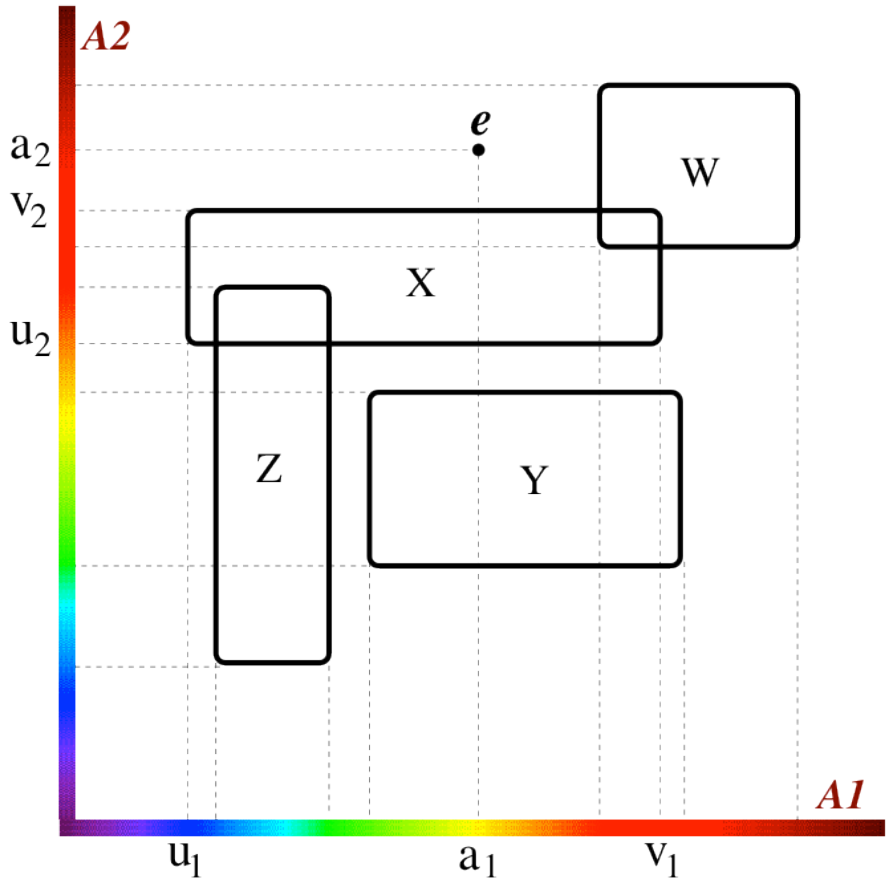
PROBLEMS?



P2P CONTENT-BASED PUB/SUB SYSTEMS

CONTENT-BASED SUBSCRIPTION MODEL

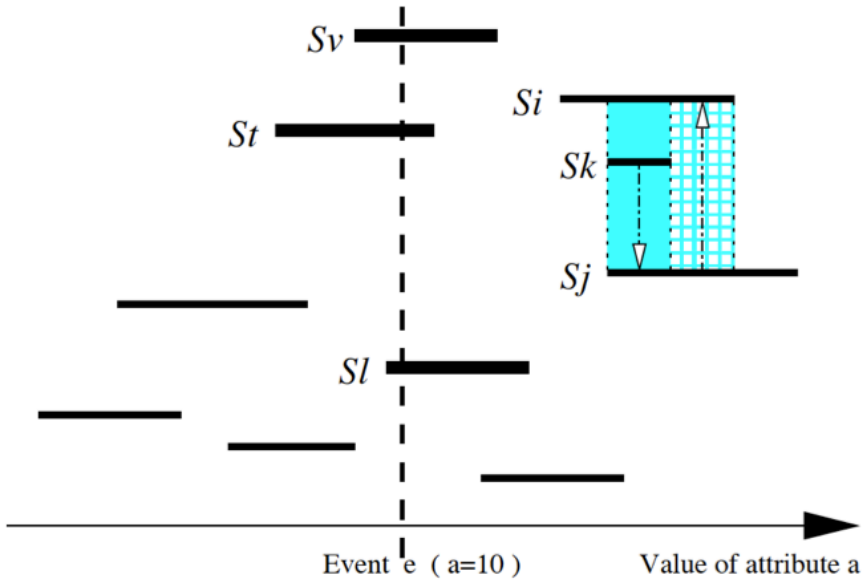
- Every node subscribes to some ranges over multiple attributes.
- An event is a point in the subscription space.

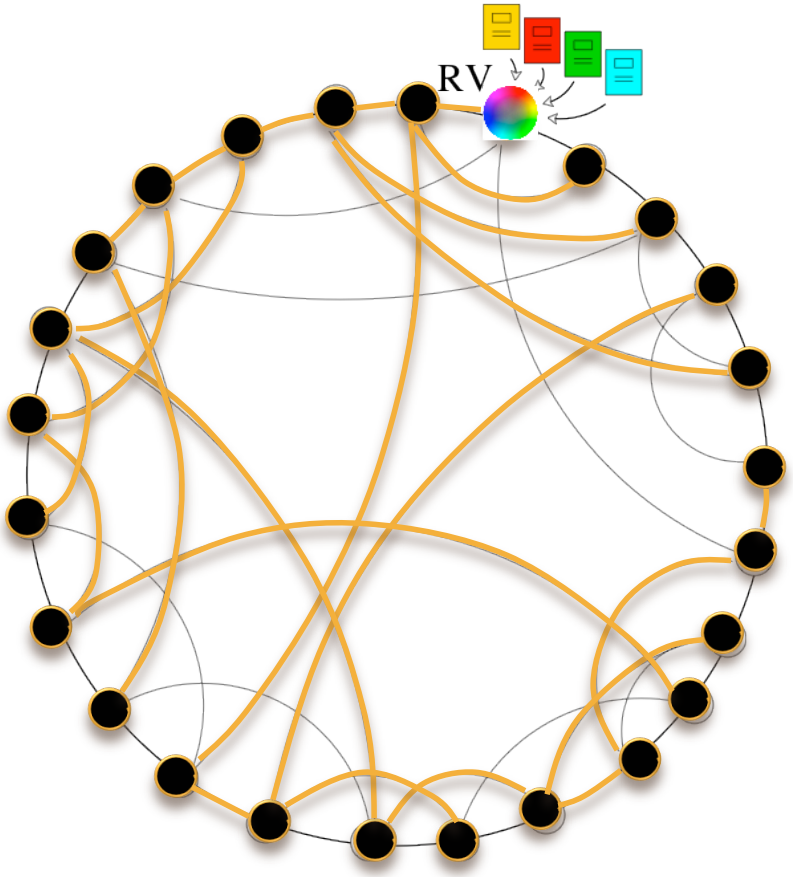


THE RANGE OF CONTENT-BASED SOLUTIONS

Sub-2-Sub

Ferry





SUB2SUB

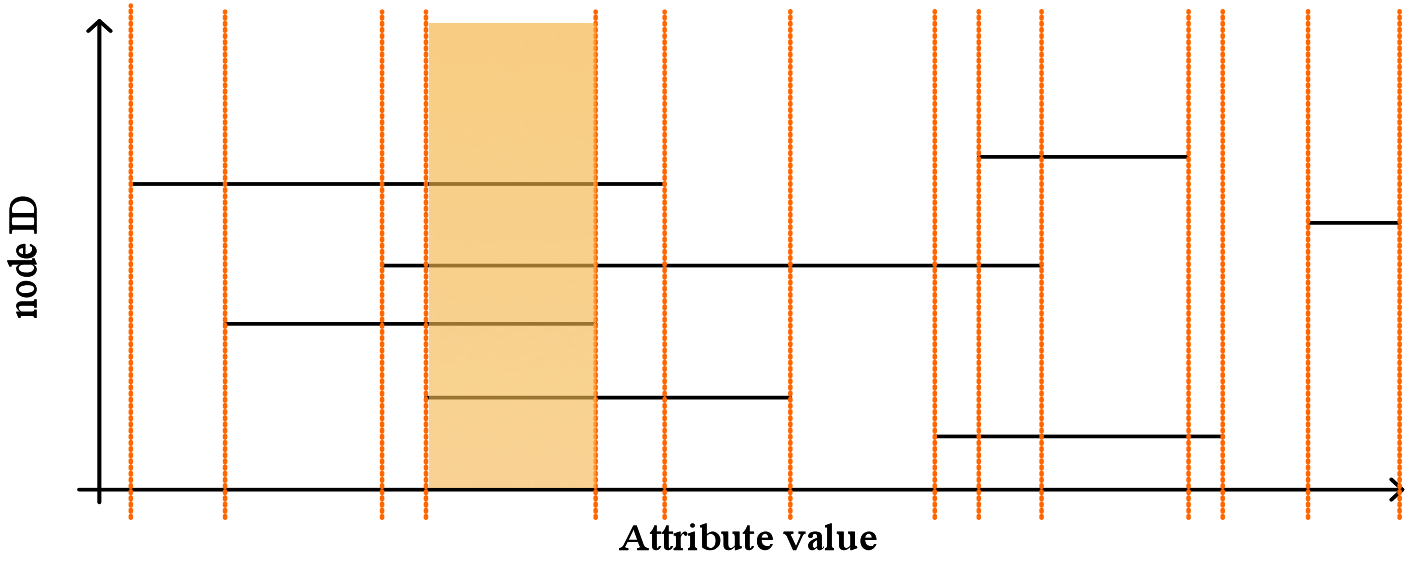
SUB-2-SUB

- Voulgaris et al. ”Sub-2-sub: Self-organizing content-based publish and subscribe for dynamic and large scale collaborative networks”, 2006

SUB-2-SUB: KEY CONCEPT

"Partition event space in homogeneous subspaces"

(homogeneous subspace: all its events have the same subscribers)

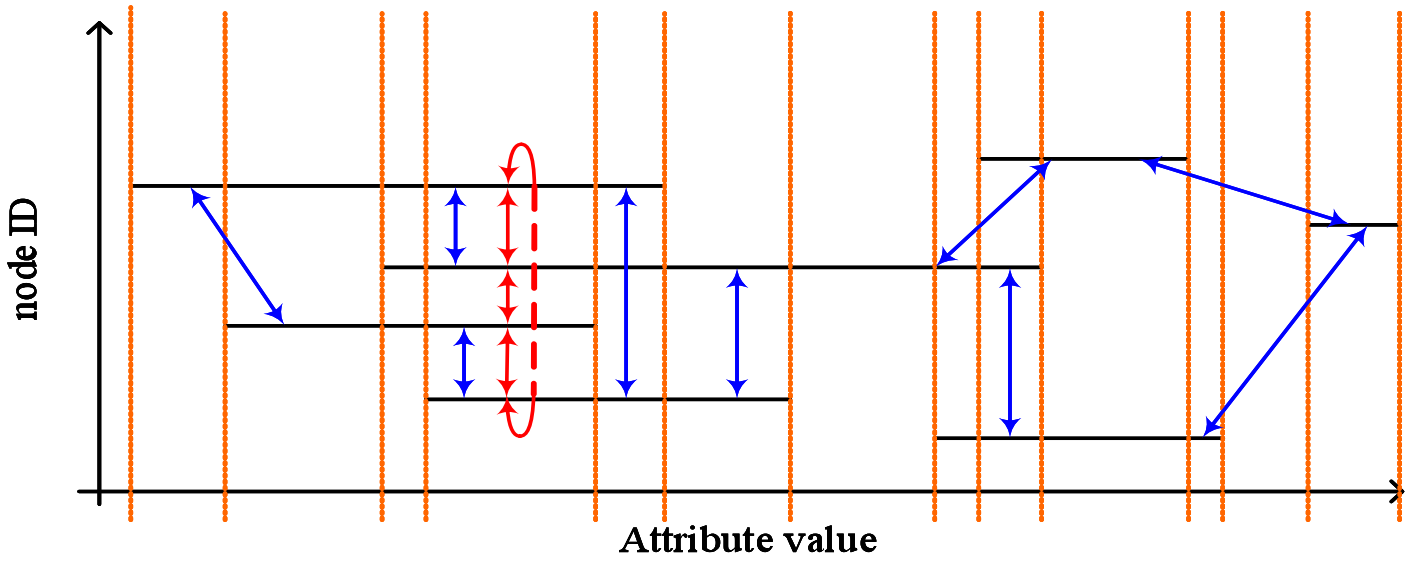


HOW DO WE BUILD SUB-2-SUB?

- Each node → 3 sets of links to other nodes (Managed by gossiping protocols based on Cyclon & Vicinity)
  - *Random links*, i.e., links to randomly selected peers in the overlay, are needed to discover nodes, and to keep the whole overlay connected in a single partition.
  - *Overlapping-interest links* reflect the similarities between subscriptions and are used to send published events to random other interested peers (and to speed up event dissemination).
  - *Ring links* are used to build a ring of nodes for each set

SUB-2-SUB: OPERATION

- Let subscribers of **“near” subspaces** discover each other
  - **Distance function**: based on Euclidean distance between two subscriptions (Distance 0 denotes overlap)
  - “VICINITY” protocol (Nodes gossip similarly to CYCLON and **keep neighbors of minimum distance**)
- Organize subscribers of the each subspace in a **ring**
  - Again “Vicinity” is forming a ring (similar to T-man)
- To publish an event, navigate to the target subspace, and hand the event to any one subscriber
  - Event reaches all and only interested subscribers, autonomously!

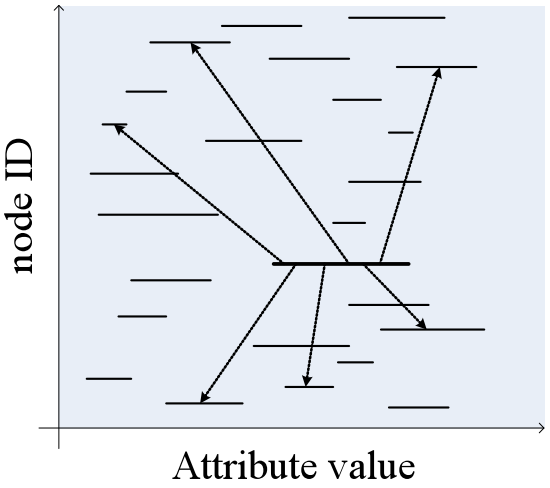


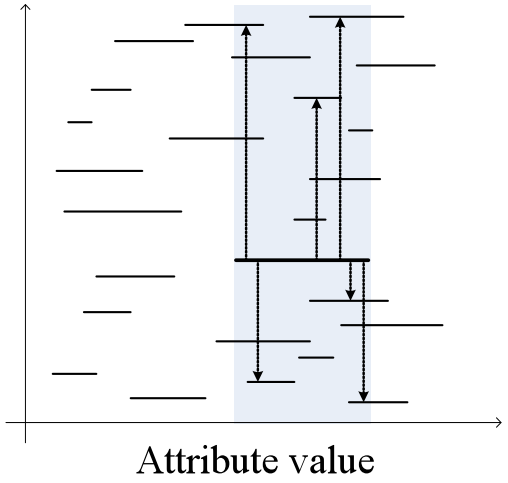
PROBLEMS?

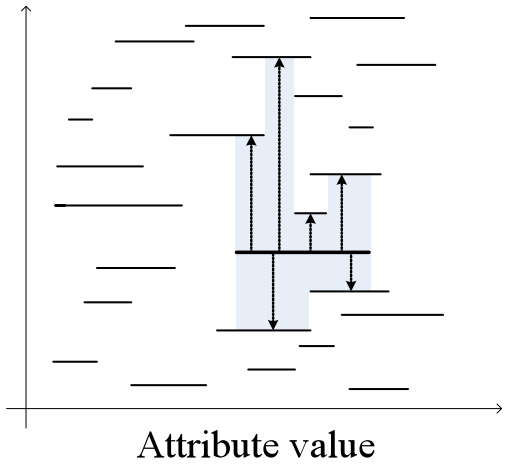
Random subscribers (CYCLON)

Overlapping subscr. (VICINITY)

Ring links (VICINITY)





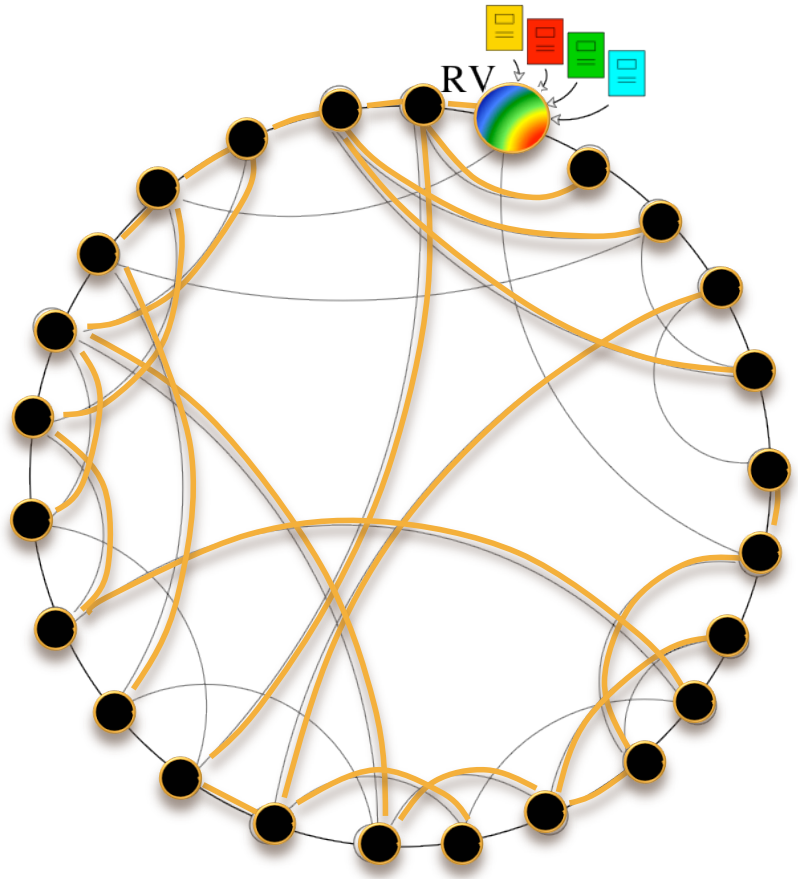


- Depending on subscription distribution a large number of ring structures must be maintained
- Node degree might grow very large
- A peer might need to participate in very large number of overlays (independent to the number of subscriptions it stores).

FERRY

FERRY: SIMILAR TO SCRIBE

- One rendezvous node per attribute

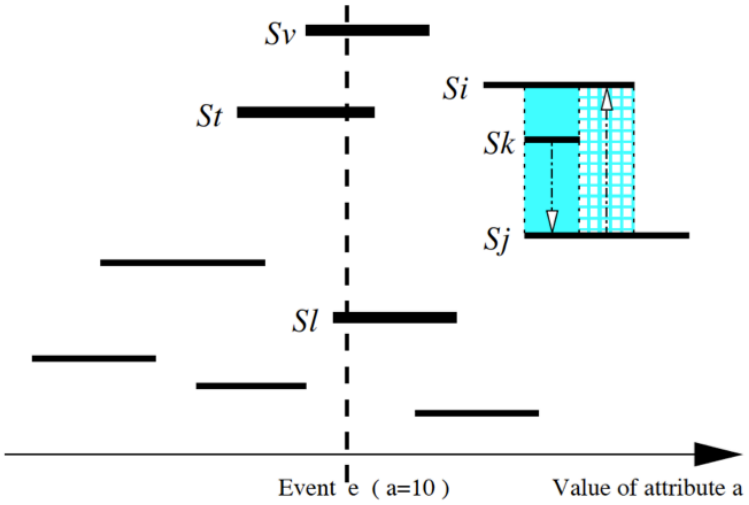


THE RANGE OF CONTENT-BASED SOLUTIONS

Sub-2-Sub | Ferry

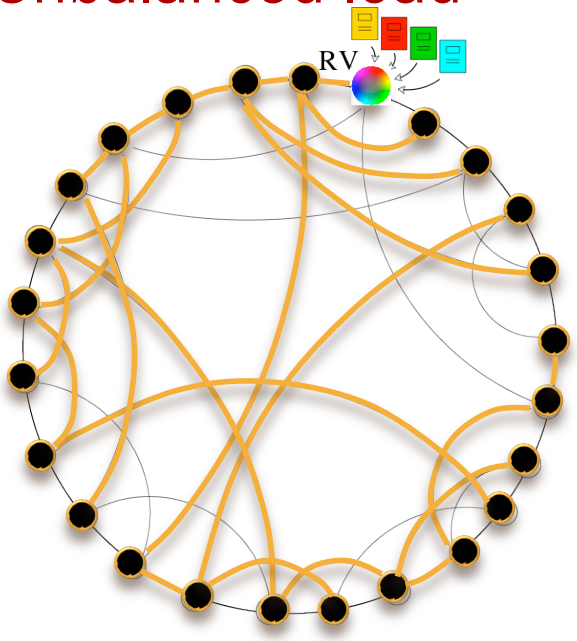
Unbounded node degree

Inefficient routing



Huge traffic overhead

Unbalanced load



Acknowledgment:

Some slides are delivered from the lecture notes by Sarunas Girdzijauskas