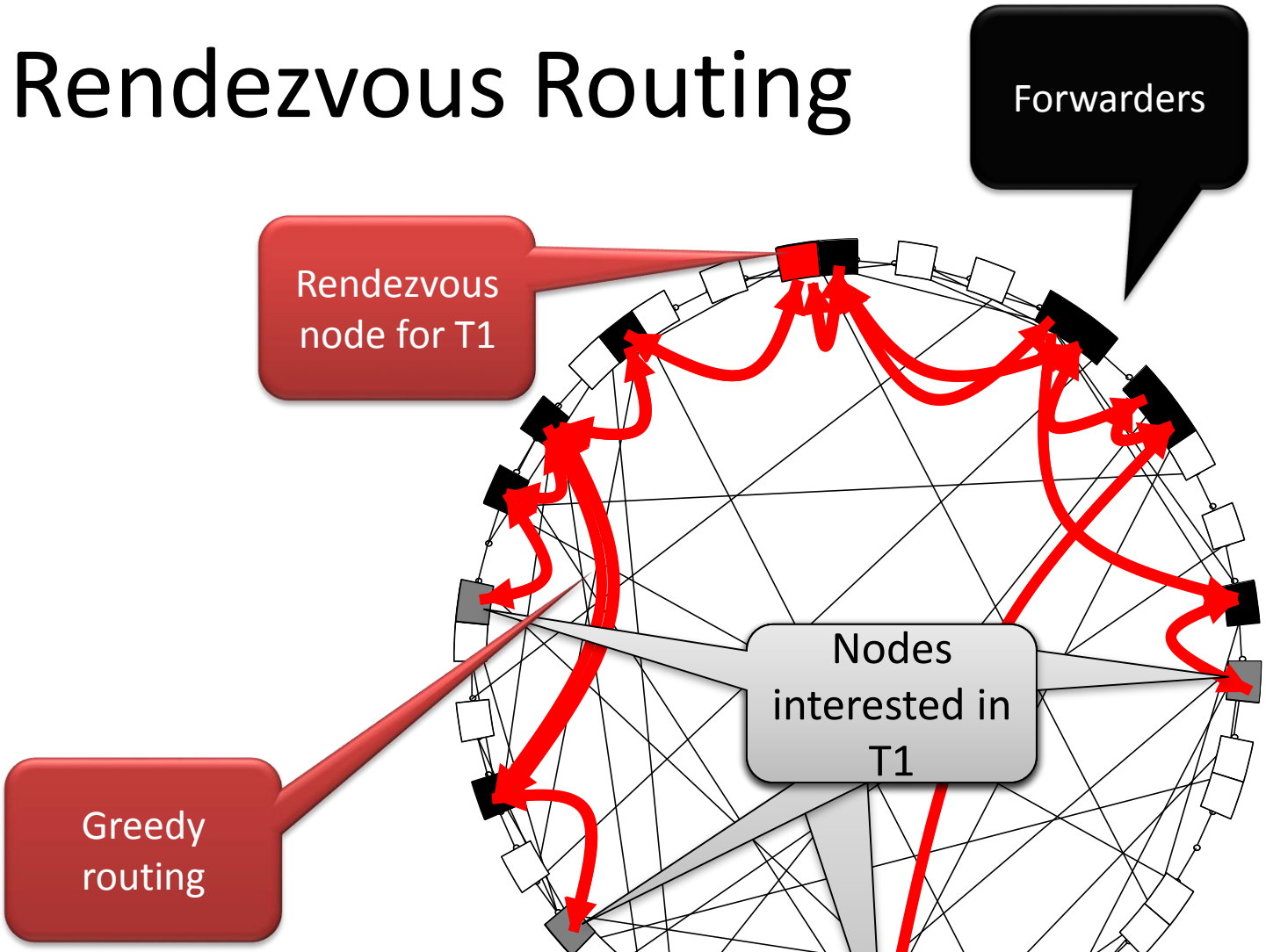
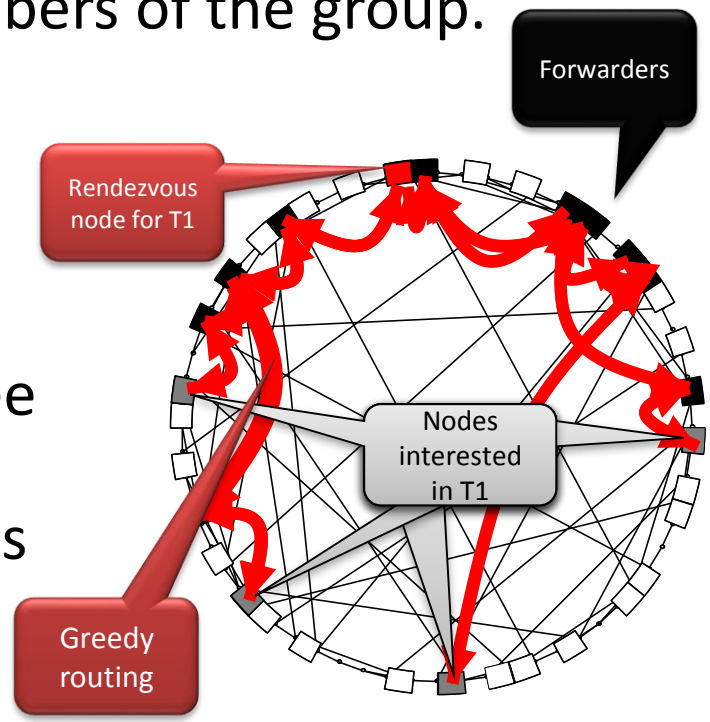
Scribe: Rendezvous Routing



- Scribe is based on Pastry Overlay
- Rendezvous Routing can be based on any navigable structured overlay (e.g., Chord, Symphony etc).

Scribe: Group Management

- Each group (topic) has an **unique *groupId***.
  - The Scribe node with a *nodeId* numerically closest to the *groupId* acts as the rendez-vous point for the associated group.
- The rendez-vous 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 the group containing an entry
  - (**IP address** and **nodeId**) for each of its children in the multicast tree
- The properties of Pastry routes ensure that this mechanism produces a tree ***without the loops***



Repairing multicast trees

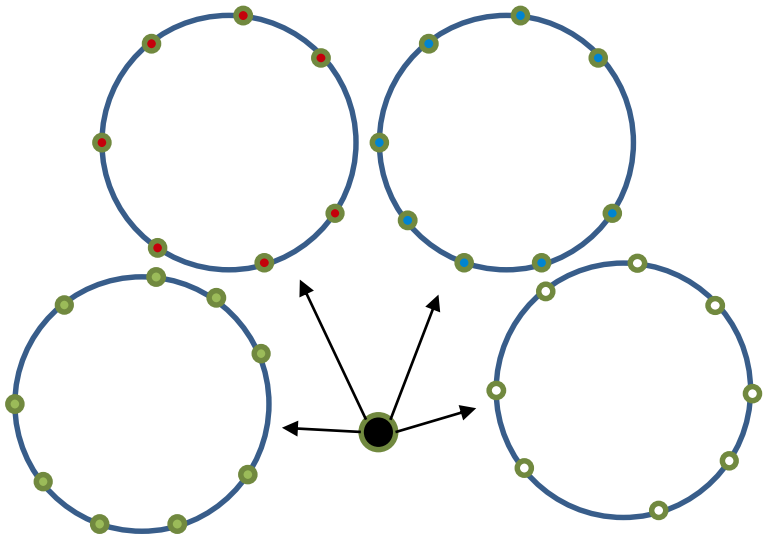
- ***What if a node fails (e.g., one of the forwarders)?***
- ***Heartbeat*** mechanism to node's children
- A child suspects that its parent is faulty when it fails to receive heartbeat messages
- Upon detection of the failure of its parent, a node rejoins the tree.
  - The node calls Pastry to **route** a JOIN message to the **group's identifier**.
  - Pastry will route the message to a new parent, thus repairing the multicast tree.

Scribe: Pros/cons

- ☺ Gets all the good properties of underlying structured P2P:
  - Decentralized construction
  - Scalability, Connectivity, Low diameter, Robustness
  - **Node degree is not blown up**
- ☹ Down sides:
  - Might involve **many non-interested** nodes
  - Does not take into account subscription similarities

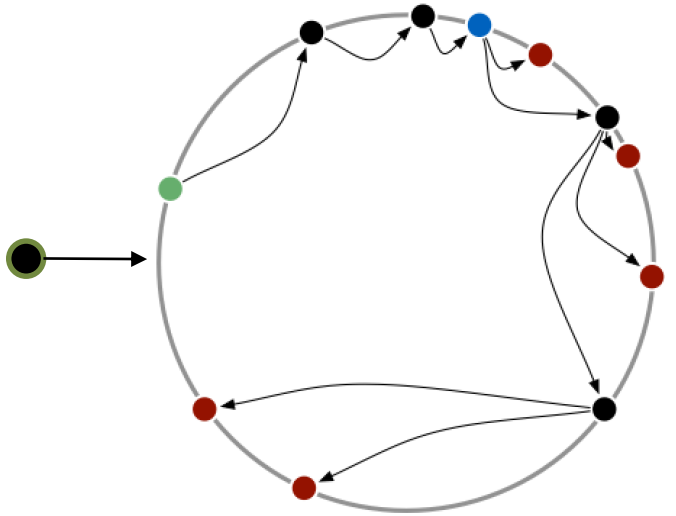
Building Decentralized Pub/Sub

- One overlay per topic



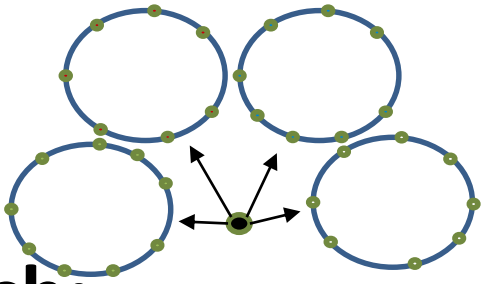
Large node degree

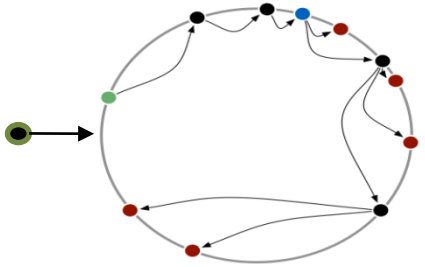
- A shared overlay for all topics



High relay traffic overhead

Vitis [IPDPS 2011]





- **Hybrid Approach:**
  - **Cluster similar nodes together** (unstructured overlay – like Spidercast)
    - Minimize the number of relay nodes by exploiting user **subscription correlation** & event **publication rates**
    - Account for the **underlying topology (bandwidth & cost) [DAIS 2012]**
  - Employ **Rendezvous routing** to enable sub-clusters find each other (structured overlay – Like Scribe)
  - **Fixed** node degree
  - **Purely gossip driven**

Overlay Creation by Gossip

Think: Spidercast by T-Man

- Gossiping enables us to **find and cluster** peers with **similar interests** connected by **cheap and low-latency** links
  - A node starts with a local fixed size view
  - Performs a bidirectional exchange of the view with a random node ⇒ 2 views
  - Keeps the only the ***preferred*** (preference function favors **similar peers** on **cheap and low-latency** links) nodes in the view ⇒ 1 view
  - Repeat

Clustering Similar Nodes Together

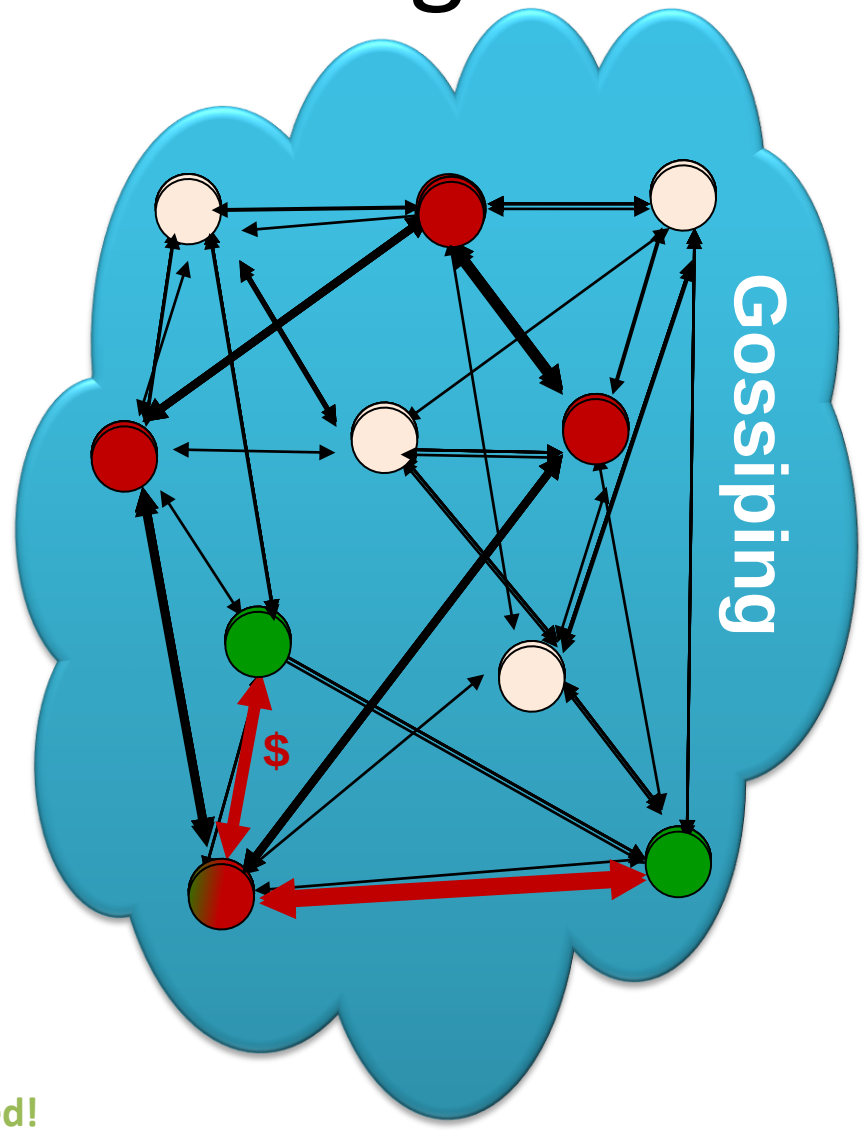
– Peer **interest similarity** metric (e.g., preference function in Tman)
  - i.e., Jaccard index

Node subscriptions s1, s2 ⊆ T

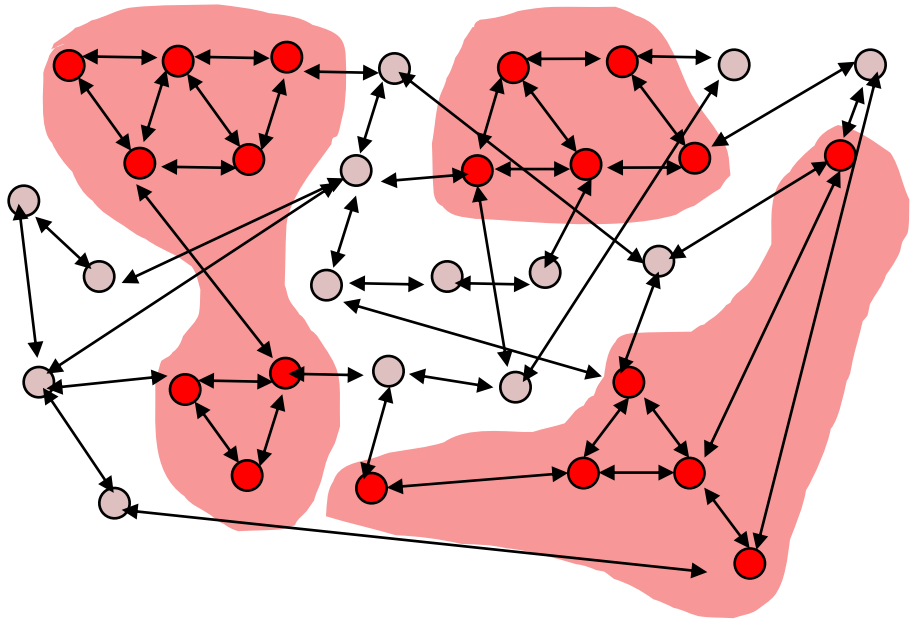
sim(s1, s2) = |s1 ∩ s2| / |s1 ∪ s2|

– Locality-aware [DAIS 2012]
  - Weighted by link **cost** (bandwidth and $)
  - Weighted by Topic **publication rates**

– Number of neighbors is **limited**!
  - **Topic-connectivity (might) be not preserved!**

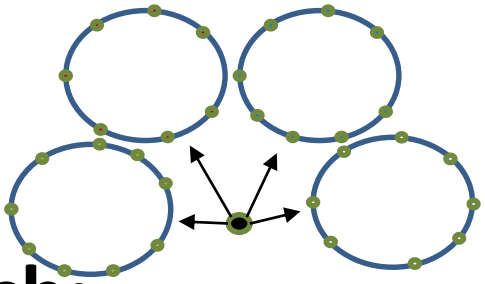


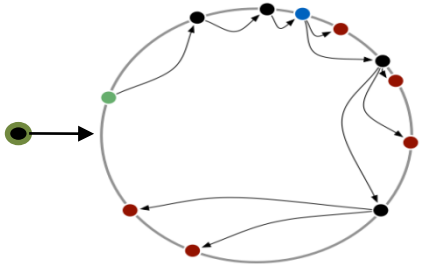
Problem: How to publish?



- Clustering peers of **similar interests** into **bandwidth** and **cost** effective clusters
  - Clusters might (will) be disjoint
  - Publishing requires connected components for each topic

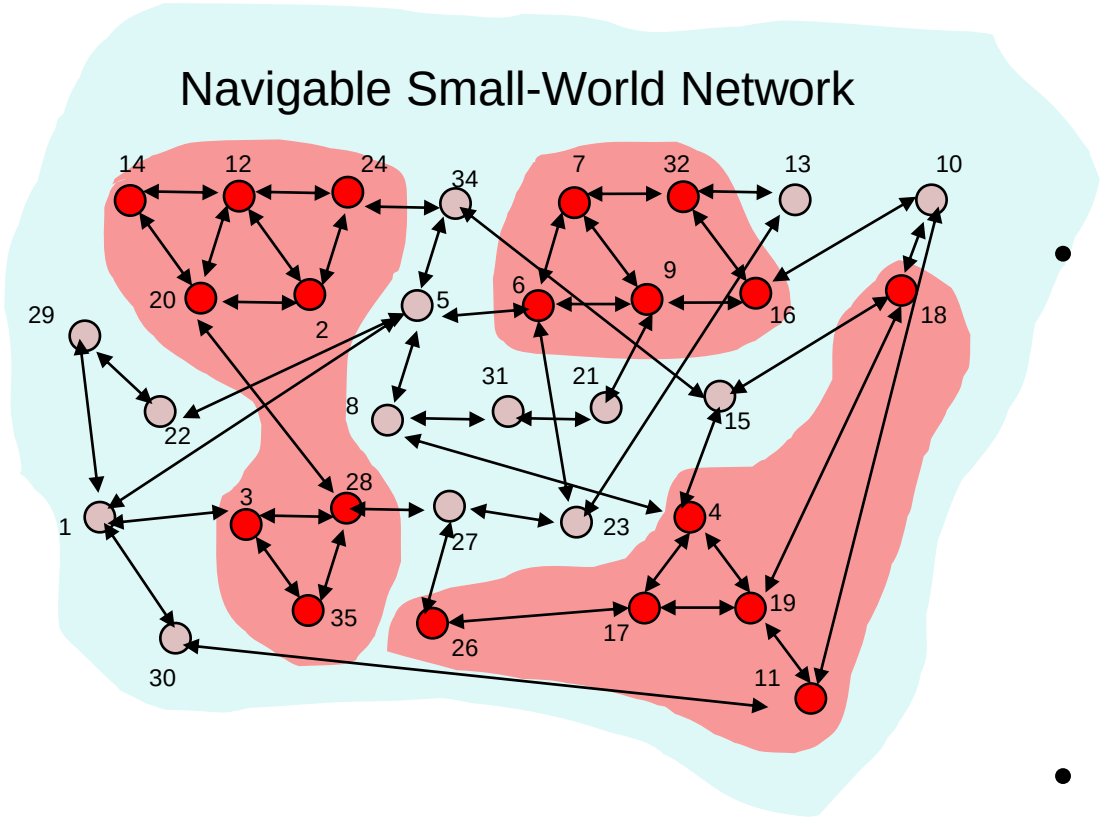
Vitis [IPDPS 2011]





- **Hybrid Approach:**
  - **Cluster similar nodes together** (unstructured overlay)
    - Minimize the number of relay nodes by exploiting user **subscription correlation** & event **publication rates**
    - Account for the **underlying topology (bandwidth & cost) [DAIS 2012]**
  - Employ Rendezvous routing to enable sub-clusters find each other (structured overlay)
  - **Fixed** node degree
  - **Purely gossip driven**

From Unstructured to Structured

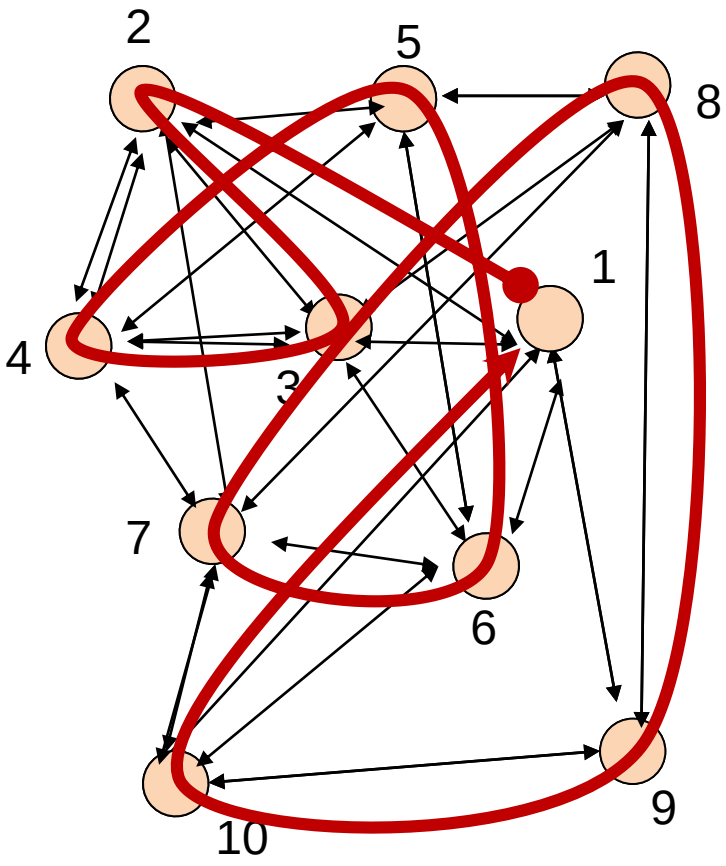


- Structure is added (Navigable Small-World made by gossiping)
- Making greedy routing possible

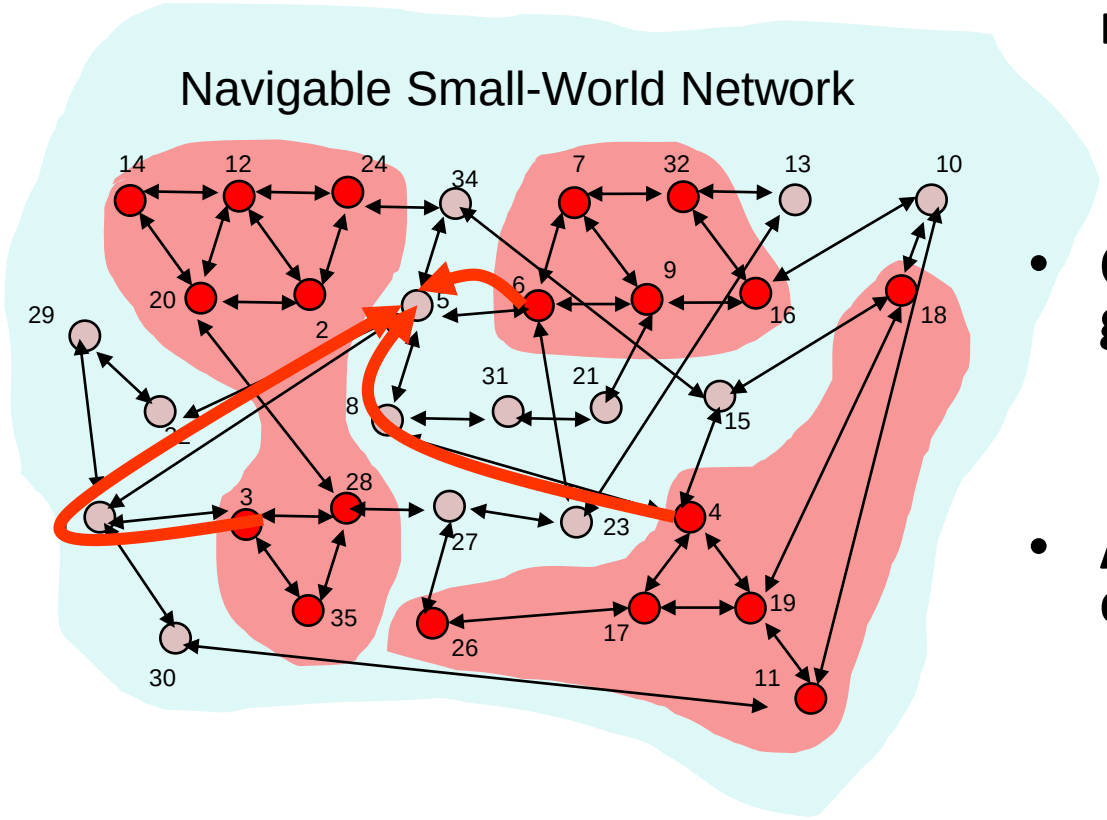
  Two type of links:
  - Ring Link
  - Long-Range link
- Do it with the same **gossiping technique**!

Building Navigable Structure

- Using gossiping again: think T-man!
- Every peer decides on random ID
- Updating ranking function for choice of neighbors:
  - Ring Link(s)
  - Long-Range link (Small-World style) for polylogarithmic routing performance



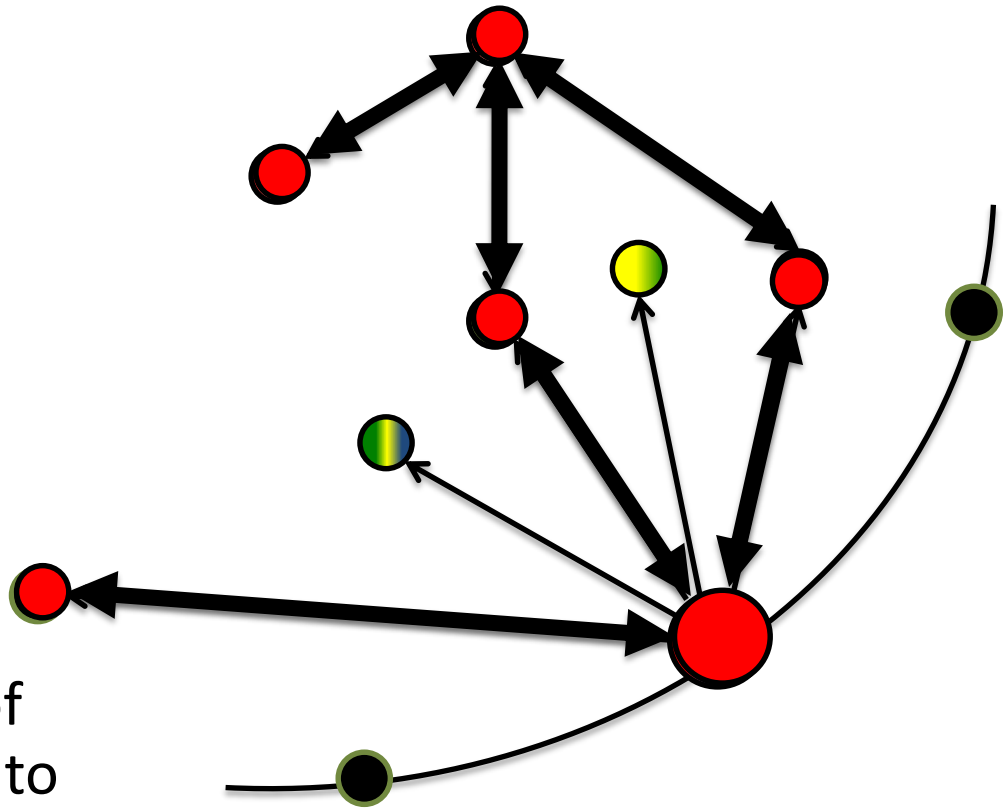
Inter-Cluster Connectivity



- **Structure is added (Navigable Small-World made by gossiping)**
  - Ring Link
  - Long-Range (finger) link(s)
  - Clustering (friend) links
- **Clusters are connected by greedy routes**
  - Rendezvous node for each topic (think: Scribe!)
  - Gateway for each cluster
- **All topics become connected**
  - For publishing “flood the topic”, or
  - Choose a rendezvous node to publish

Recap: Neighbor Selection

- Ring
- Small-World
- Similarity Clusters
- Forming Topic sub-clusters



– Up to **10 fold** reduction of relay traffic as compared to existing approaches (e.g., Scribe, Bayeux)

Vitis Recap

- Large scale pub/sub for heterogeneous environments
  - Huge potential of combining two different paradigms:
    - **unstructured similarity based overlays**, and
    - Navigability enabled by **structured small-world overlays**
- Dissemination structures are **self-organizing** and highly **robust** due to **gossiping**

Publish/Subscribe: recap

- **Topology**
  - Structured/Unstructured/Hybrid
- **Event Routing (dissemination)**:
  - Event flooding/random-walk/Rendez-vous
  - What are the advantages/disadvantages?
- How would you compare different approaches?

Acknowledgements:
Some slides were derived from the lecture notes of G. Chockler (IBM Research Haifa)