Routing Indices

For Peer-to-Peer Systems

presented by
Wei Xie
University of Toronto

Outline

- Introduction to Peer-to-Peer System
- Routing Indices
- Cycles in Peer-to-Peer Network
- Parallel Alterative of Routing Indices
- Conclusion

Peer-to-Peer System

- Formed by a large number of nodes that can join or leave the system at any time
- Distributed computing nodes of equal roles or capabilities exchange information directly with each other

Famous P2P Systems

- Napster
- Gnutella
- Freenet
- GIOSS

Mechanism of Searching in P2P system

- mechanisms without an index
  - flooding network with queries
- mechanism with specialized index node
  - index node becomes bottleneck
- mechanism with indices at each node
  - routing indices
- mechanism with super nodes
  - every node stores metadata about its neighbors within depth n

Routing Indices

Select the best neighbors to send query to

Key factor is how to define and compute the goodness suitable to a specific query

- **Compound Routing Index (CRI)**
- **Hop-count Routing Index (HRI)**
- **Exponentially Aggregated Routing Index (ERI)**

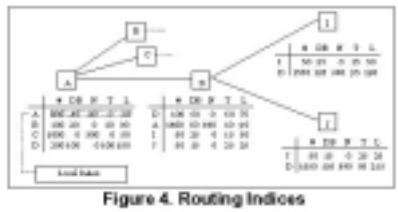


Figure 4. Routing Indices

Compound RI

- **summary of detailed index**
- **compact data structure**
- **goodness= the number of documents that could be found in a path**
- **send query in depth-first fashion**
- **savings in the number of messages**
- **modest storage space**
- **introduce errors of overcount or undercount**

$$NumberOfDocuments \times \prod_i \frac{CRI(i,i)}{NumberOfDocuments}$$

- *maintenance?*
- *effect of errors?*

Path	# docs	Documents with topics DB	N	T	L
B	100	20	0	10	10
C	1000	0	900	0	80
D	200	100	0	100	180

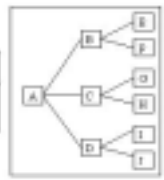


Figure 3. A Sample Compound RI

Hop-count RI

- take into account of numbers of hops
- regular-tree model to compute goodness
- need local network structure to decide value of *horizon*
- higher storage and transmission cost
- potential loss in accuracy
- complicated maintenance algorithm

$$goodness_{hc}(Neighbor_i, Q) = \sum_{j=0...h} \frac{goodness(N_i[j], Q)}{F^j}$$

	1 Hop					2 Hops				
Node	#	DB	N	T	L	#	DB	N	T	L
X	60	13	2	8	10	20	30	10	4	17
Y	20	0	3	15	12	80	31	0	25	20
Z	5	2	0	5	5	70	30	40	20	50

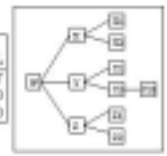
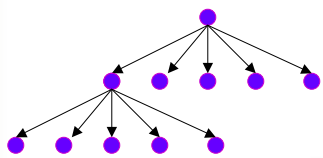


Figure 6. A sample Hop-count RI for node W

Exponential RI

- keep info. for all nodes accessible from each path
- dynamic data structure
- need complete information of local network
- unrealistic assumptions of regular tree model



Regular Tree

Path	#	DB	N	T	L
X	66.67	16.33	5.33	6.33	15.67
Y	46.67	10.33	5.00	20.00	18.67
Z	28.33	5.33	13.33	9.67	19.67

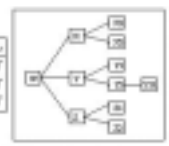


Figure 7. A sample Exponential Routing Index for Node W

Algorithms for creating/update RI, answering queries

```
RI Creation/Update Algorithm
Input: I
Output: An updated RI (as a side effect)

// Creation phase:
RI[0] = Summary(I)
For each neighbor j:
    Send(j, RI[0])

// Update phase:
OldRI = RI
While (true):
    Wait for a batch of aggregated RIs to arrive and/or
      for the local index to change
    For each aggregated RI (if any), A_k, received from neighbor k:
        RI[k] = A_k
    If local index changed:
        RI[0] = Summary(I)
    If OldRI is different enough than RI:
        OldRI = RI
        For each neighbor j in RI:
            A = Aggregate(RI[0..j-1, j+1..k])
            Send(j, A)
```

```
Answer Query
Input:  A query Q, a client C, and a node F
Output: Query answers, Query forwards

Send local answers for Q to C
If StopCondition():
    return
Rank = [ ]
For each neighbor j except F:
    Rank.Append([j, Estimator(RI[j])])
    Sort tuples in Rank by second attribute
for f in Rank:
    Send query to neighbor f[0]
    If StopCondition():
        return
```

Figure 7: RI Usage Algorithm

Effect of errors

assumption:

- **documents are organized into categories**
- **index is hash table of categories**

results show:

- **loss is modest**
- **similar results to undercounts and mixed errors**

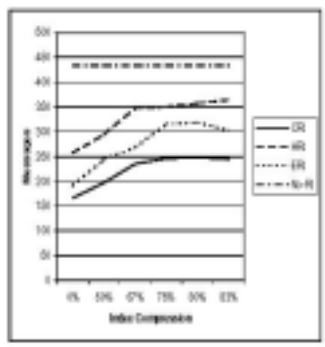
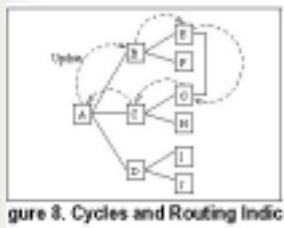


Figure 15: Effect of Overcounts

Cycles in the P2P System



gure 8. Cycles and Routing Indices

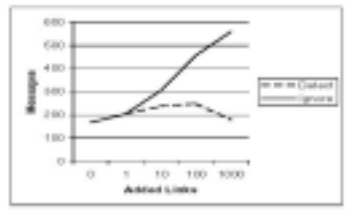


Figure 16. Effect of Cycles

- No-op
- Cycle avoidance
- Cycle detection and recovery

- signature: unique message id
- ignore some update messages

Performance Evaluation

Topology

- Tree
- Tree with added cycles
- Power-law graph

Data Distribution

- Uniform
- 80/20

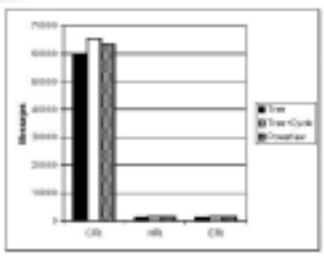


Figure 18. Updates and Network Topology

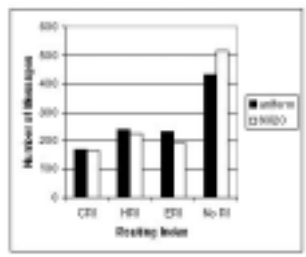
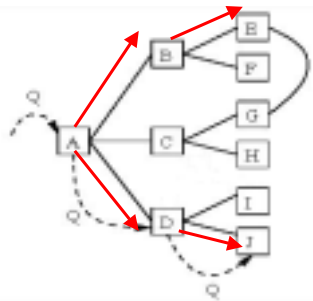


Figure 10. Comparison of CRI, HRI, and ERI

Parallel Alternative of Routing Indices

for each node i in data structure
if $|goodness_{highest} - goodness_i| <= threshold$
{
send Q to node i
apply RI algorithm
}

- *threshold* **represents the possibility similarity of finding results in different paths**
- **value of** *threshold* **is decided by many factors:**
  number of results (stop condition)
  response time limitation



Better response time!

Conclusion

- **Propose simple and effective strategies**
- **Compact data structures and algorithms**
- **Do not need to know global network architecture**
- **Reduce the number of messages greatly**
- **Sequential algorithms, but can be changed to parallel ones easily**
- **Response time is worse than non-index systems**
- **Paper is easy to understand**

Thanks for your patience!

Happy Halloween!