Structuring P2P networks for efficient searching

Rishi Kant

and

Abderrahim Laabid

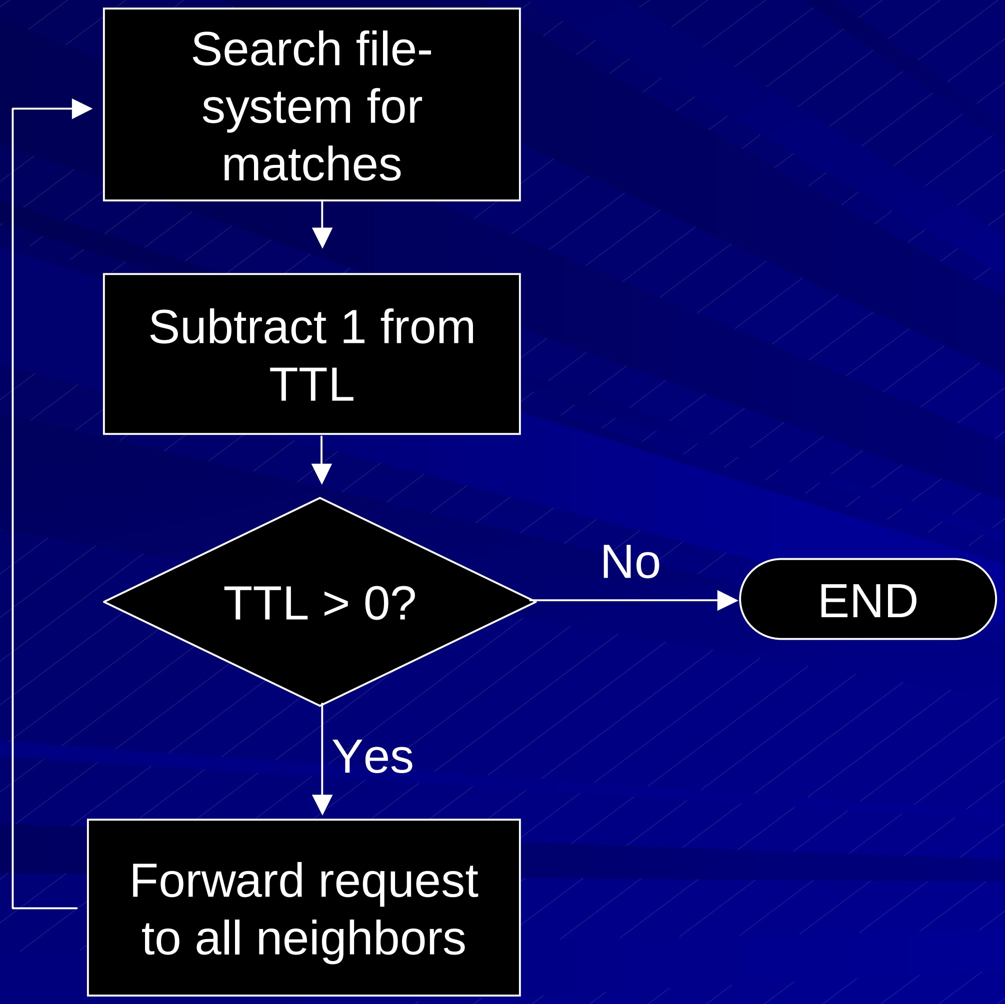
Motivation

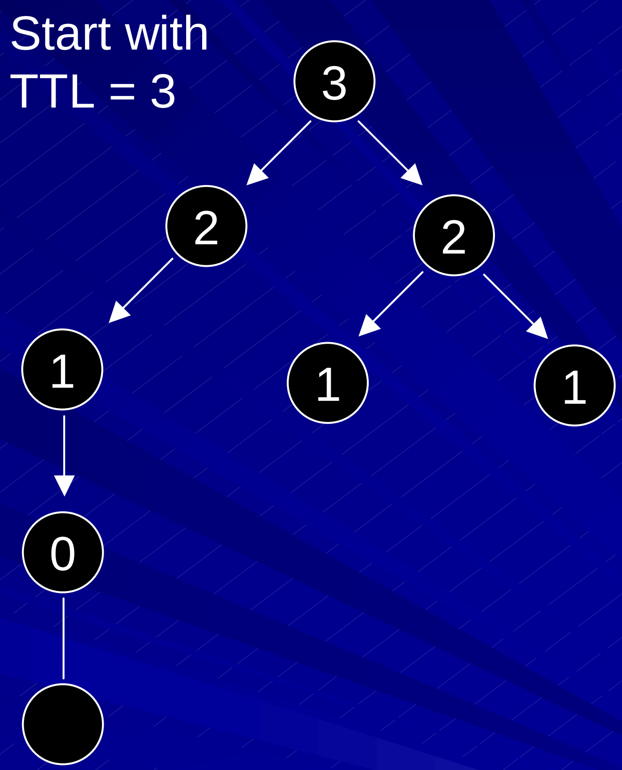
- File-sharing is most popular use of P2P networks
- For file-sharing, searching is the most important operation
- Structuring P2P networks for efficient searching has not been widely explored

Present P2P search methods

Network	Search method	Disadvantage
Gnutella	BFS with TTL	High network overhead
Kazaa	Super-nodes	Processing and memory overhead borne by few nodes
Chord	None	N.A.

BFS with TTL (Gnutella)



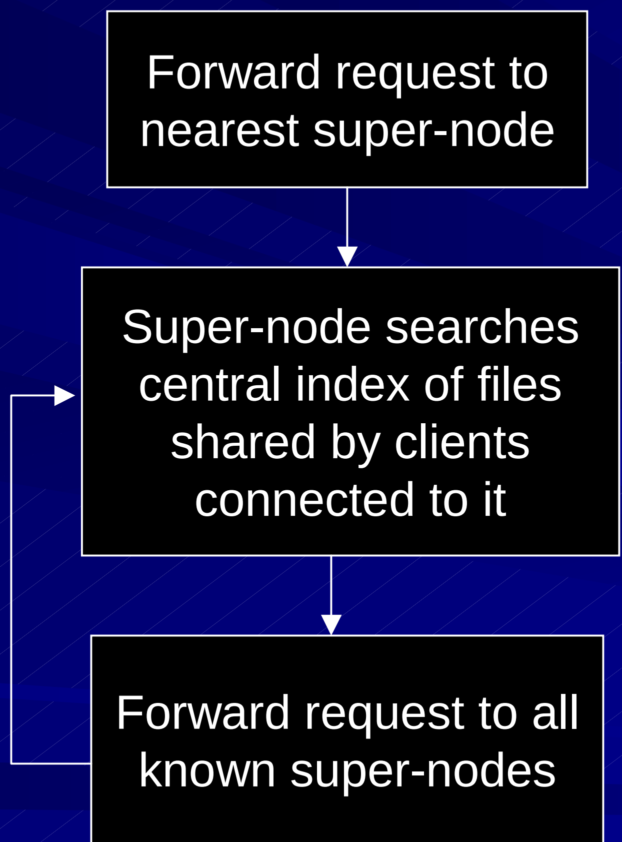


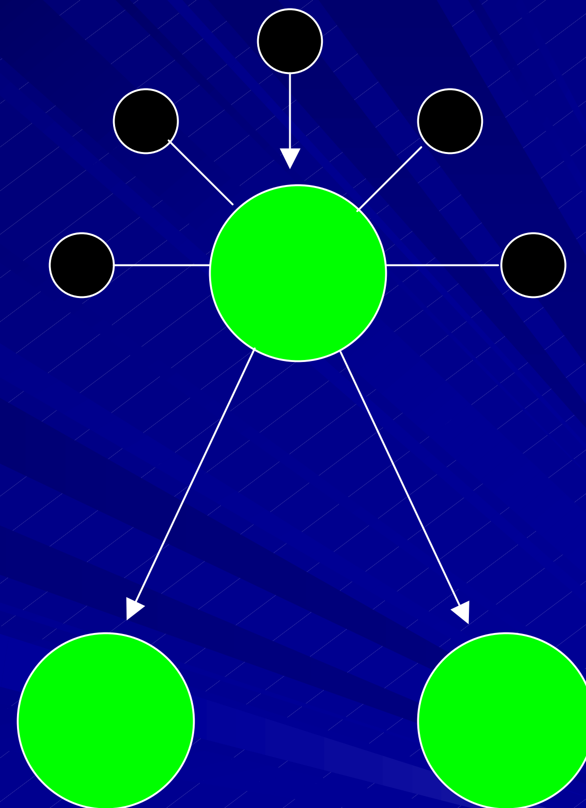
O (min (log N, TTL))

BFS with TTL

Advantages	Disadvantages
No need for central server	Search request may loop back
Nodes share processing overhead	Nodes which have no matches also participate in search
Robust to nodes joining and leaving network	Nodes may receive same request from alternate routes

Super-nodes (Kazaa)





O (log N – log n)

Super-nodes

Advantages	Disadvantages
No need for central server	Search requests may loop back
Nodes share processing overhead	Search overhead shared by select nodes only
Faster searches compared to BFS	Not as robust to network changes

Problem

- Can we formulate a P2P network structure that is better adapted for searching ?

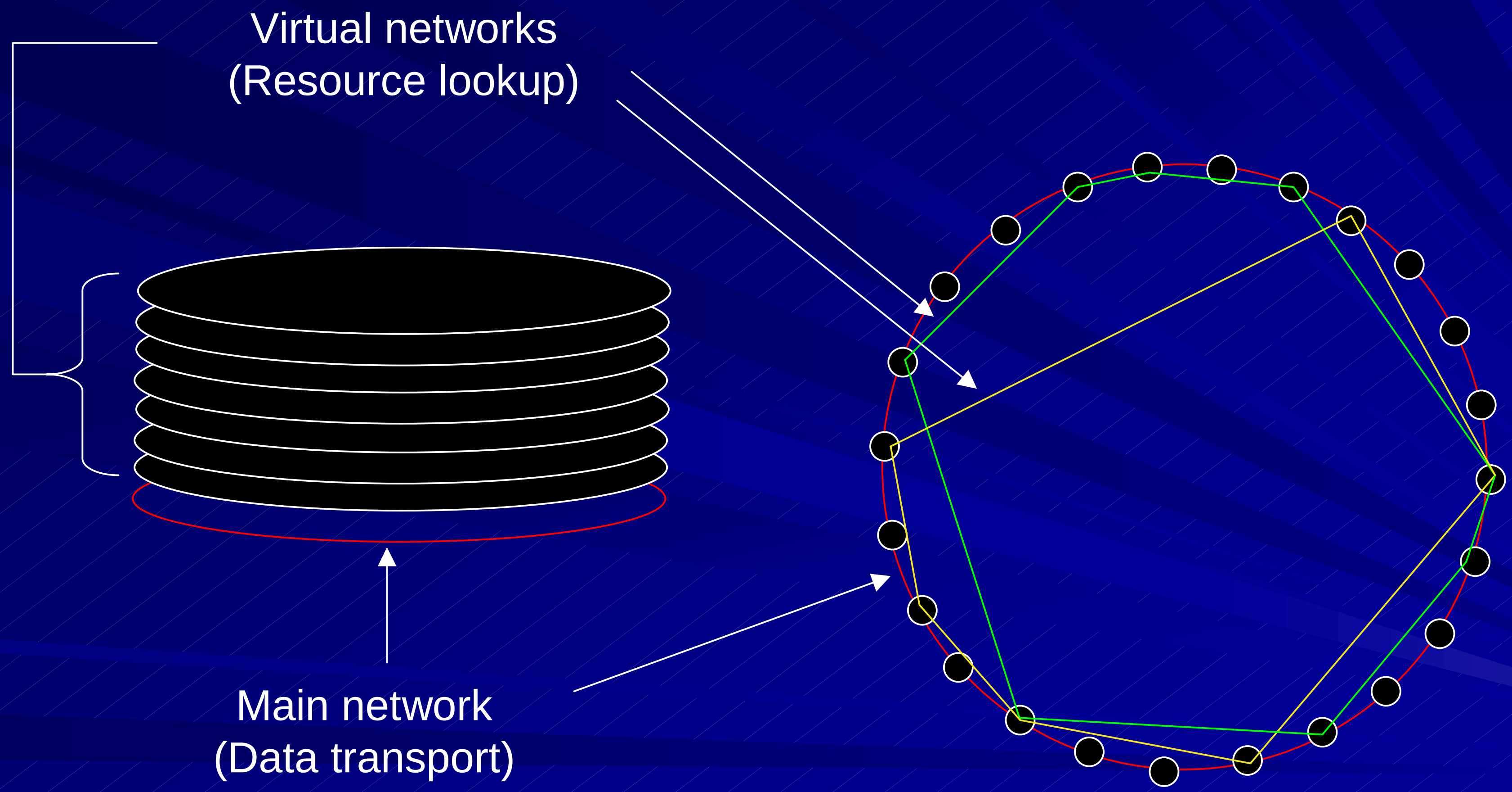
Desirable properties

- Efficient
  - No duplication from loop backs or alternate routes
  - Provide directed searches
- Robust
  - Minimally affected by network changes
- Adaptable
  - Restructures according to on-going activity

Stacked Virtual Search Rings

- One main network responsible for data lookup / transfer given resource ID
- On top of main network, construct stack of virtual ring networks that connect "similar" nodes based on some criteria for similarity
- Given a search request, find virtual network which most closely matches request and traverse it to find resource IDs

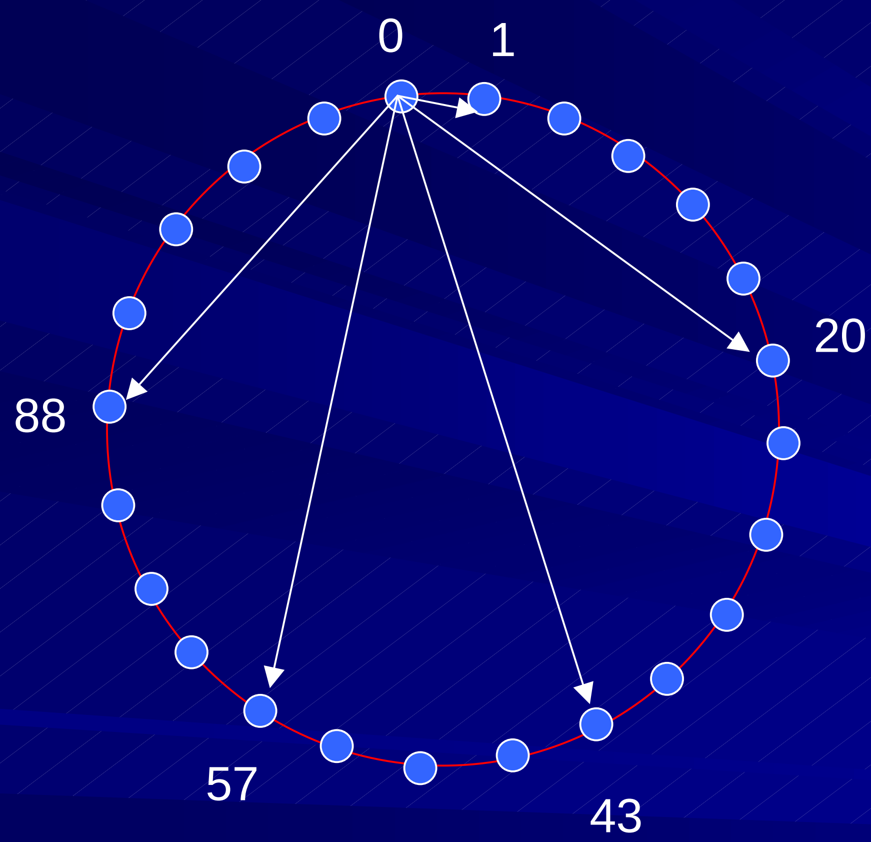
Stacked Virtual Search Rings



Observations

- Generalized search structure model
- BFS with TTL method implemented by having a single VN connecting all nodes
- Super-nodes method implemented by having single VN linking all super-nodes

Elimination of duplication



- While forwarding request, indicate stop address
- Request will not be forwarded beyond stop address
- e.g. Request forwarded to 1 will not go beyond 19
- Thus, no duplication

Directed searches

- If the criteria for similarity of a VN closely matches the search request, then all the nodes on that VN have a higher probability of finding a match than the nodes not on the VN ➔ more directed search
- Example: If criteria for similarity is “Have a file starting with sequence ‘ab’”, and the search request is for ‘abc*’

Advantages

- Ring structure of networks prevents unnecessary duplication
- Traversing VNs allows for directed searches
- Network search structure adapts to on-going requests through restructure of VNs
- VN abstraction allows use of any underlying data transport network

New problem

- What criteria should be used to link nodes in a VN ?

Proposed solution

- Assign each VN a unique ID $x_1x_2x_3x_4$, where each x_i is an alphanumeric char
- Define surrounding nodes as all nodes within d hops from current node
- A node is elected to a VN $x_1x_2x_3x_4$ by its surrounding nodes when a search request is received via pure flooding that matches at least 1 file on one of the surrounding nodes

Proposed solution

- All surrounding nodes send the list of files they have that match the sequence $x_1x_2x_3x_4$ to the elected node
- The number of VNs that a node may be part of is limited to a reasonable quantity
- No keep alive messages are sent for the VN. The structure is verified only when a node is added or removed from a VN.

Operation

- The client node looks up its table to find matching VNs
- If none found, it locates VNs that match its search request (S) by doing a BFS with TTL
- A VN matches S if (*S* matches $x_1x_2x_3x_4$) || (*$x_1x_2x_3x_4$* matches S)
- The client node forwards the request to a single node in each of the VNs it found
- Within the VNs, the nodes do a BFS with TTL to locate matching resources

Remarks

- VNs are constructed based on search requests and content available ➔ adapts
- Localized “super-nodes” are elected ➔ faster searching and load balancing
- Any node may be elected ➔ all nodes participate in searches
- Surrounding nodes between neighboring “super-nodes” may overlap making system more robust to “super-nodes” leaving

Potential problems

- If a node is part of 2 VNs which both match S, then the node will get the same request twice
- Virtual networks may get fragmented due to lack of keep-alive messages

Future work

- Simulations need to be performed to verify and quantify reduction in search overhead
- Implementation details need to be analyzed and discussed in more depth

Conclusion

- Explored present solutions available
- Described SVSR method for structuring P2P networks to optimize searches
- Proposed a criteria for constructing VNs