On the Discovery of Brokers in Distributed Messaging Infrastructures

Authors

Shrideep Pallickara, Harshawardhan Gadgil, Geoffrey Fox
Community Grids Lab
Indiana University

Presented by

Harshawardhan Gadgil
hgadgil@cs.indiana.edu

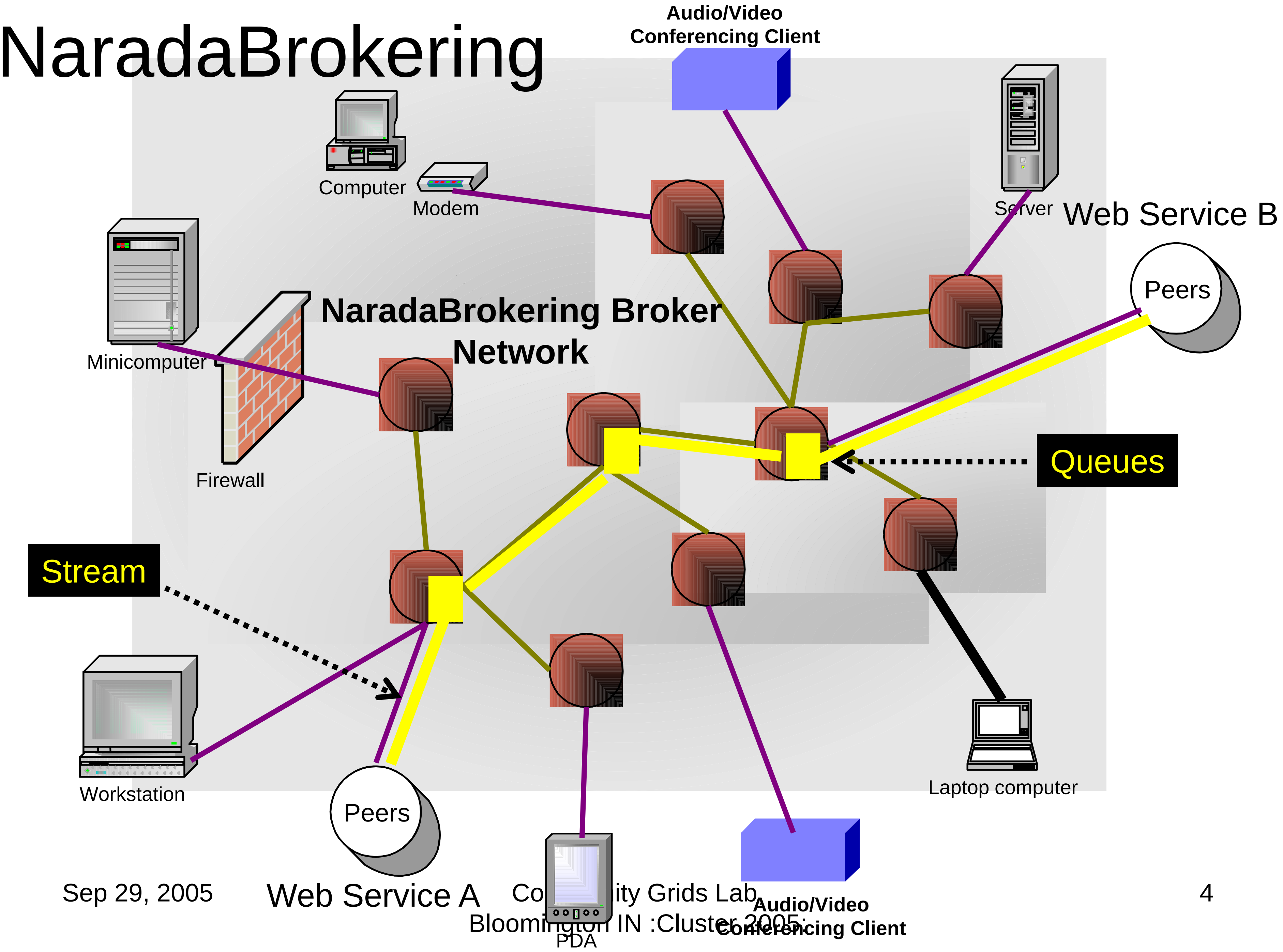
Talk Outline

- Overview of Distributed Messaging Architecture (NaradaBrokering)
- Motivation / Requirements
- Related Work & Our approach
- Some results
- Discussion of results

NaradaBrokering

- Distributed messaging middleware based on a network of cooperating broker nodes
  - Cluster based architecture allows system to scale in size
- Originally designed to provide uniform software multicast to support real-time collaboration linked to publish-subscribe for asynchronous systems.
- Project Website: **`http://www.naradabrokering.org`**

NaradaBrokering



NaradaBrokering Broker Network

NaradaBrokering Core Features

- Multiple protocol transport support
  - Transport protocols supported include TCP, Parallel TCP streams, UDP, Multicast, SSL, HTTP and HTTPS.
  - Communications through authenticating proxies/firewalls & NATs. Network QoS based Routing
  - Allows Highest performance transport
- Subscription Formats
  - Subscription can be Strings, Integers, **XPath** queries, **Regular Expressions**, **SQL** and tag=value pairs.
- Reliable Delivery
  - **Robust** and **exactly-once delivery** in presence of failures
- Ordered Delivery
  - **Producer Order** and **Total Order** over a message type. **Time Ordered** delivery using Grid-wide **NTP based absolute time**
- Recovery & Replay
  - **Recovery from failures** and disconnects. **Replay** of events/messages at any time. **Buffering** services.

NaradaBrokering Core Features

- Security
  - **Message-level** **WS-Security** compatible **security**
- Message Payload Options
  - **Compression** and **Decompression** of payloads
  - **Fragmentation** and **Coalescing** of payloads
- Message Compliance
  - Java Message Service (**JMS**) 1.0.2b compliant
  - Support for routing P2P **JXTA** interactions.
- Grid Feature Support
  - NaradaBrokering enhanced **Grid-FTP**. Bridge to **Globus** **GT3**.
- Web Service Support
  - Implementations of **WS-ReliableMessaging**, **WS-Reliability** **and** **WS-Eventing.**

Discovering Brokers

- Motivation
  - Peer-to-peer systems are very dynamic
  - Middleware manages scalability and availability to maximum extent ! HOWEVER...
  - Client's responsibility to discover the most appropriate broker that would maximize the its ability to leverage the services provided
  - Accessing the same broker over and over (statically configured) may lead to poor bandwidth utilizations and performance degradation

Desiderata

- The Discovery process must work on current state of broker network
  - Thus newly added brokers would be automatically and quickly assimilated
- Discovery process must be independent of failures within the brokering system
- Should result in better utilization of network and networked resources
  - Find the nearest (network distance) broker from the set of available brokers

Existing approaches

- IDMaps
  - Uses specialized nodes (tracers) that maintain the topology map of the network

    Shortest distance between A and B

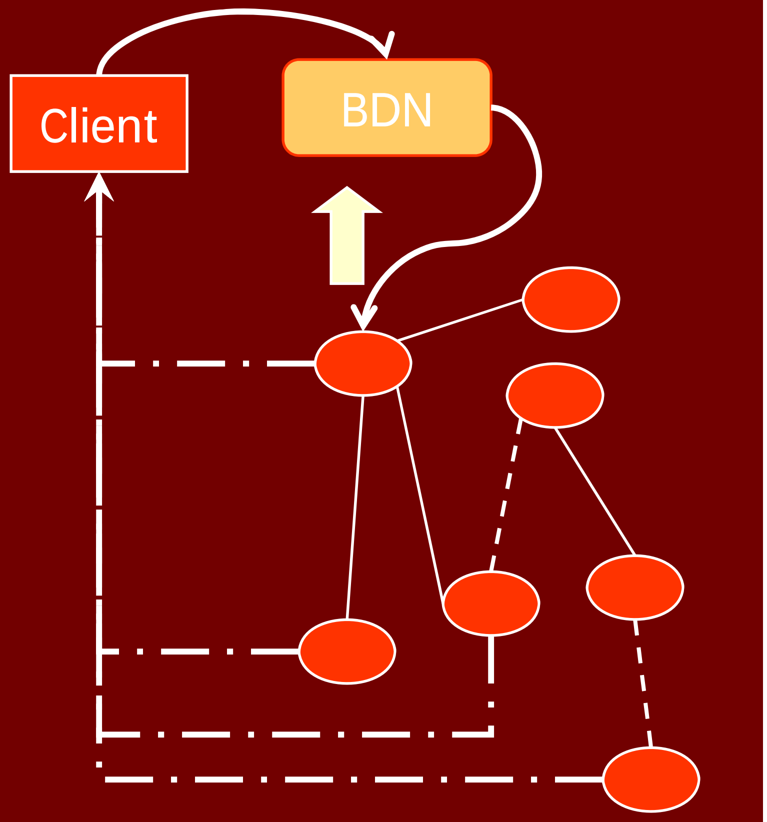
    $= D_{\boldsymbol{A-T1}} + D_{\boldsymbol{B-T2}} + SD_{\boldsymbol{T1-T2}}$
  - More the number of traces, better the accuracy of prediction. However requires internet-wide deployment of tracers
- JXTA uses rendezvous peers to match peers with matching constraints.
  - Requires knowledge of existence of these peers and means to connect to them

Existing approaches (contd.)

- Tiers approach uses hierarchical grouping of peers to improve scalability of the system
- Distributed Binning and Beaconing requires deployment of landmark entities to serve as reference points for proximity tests
- Global Network Positioning uses a distance function over a set of coordinates that characterize the position of the entity in the network to compute the nearest distance
  - The approach presented in this paper uses only UDP Ping to compute average Round-Trip time to calculate proximity

Our approach

- Broker Discovery Node (BDN)
  - Registry of existing brokers
  - Forwards **`BrokerDiscoveryRequest`** to registered brokers
  - As soon as a discovery request arrives, it is propagated to all connected brokers over a special topic
- Brokers matching requested criteria respond using UDP
  - Why UDP ? Unreliable, hence response OR lack of one is a good measure of availability of broker



Our approach

- Avoiding Flooding:
  - Each discovery request has a UUID. Broker keeps track of (say) 1000 UUIDs. If a request comes with a UUID already seen, then request is dropped and not propagated.
- The client constructs a Broker Target Set from a set of weighed metrics
  - Number of links, Total / Available memory, Response time
  - NOTE: Broker Target Set is very small (usually should be 3 – 5 OR less)
- The client then re-pings each broker from the target set to determine the nearest broker (OR uses some other criteria to determine the broker to connect to).

Advantages of our approach

- Not all brokers in the brokering network need to be registered with the BDN
  - In fact if only one broker is registered, it suffices.
  - What happens if broker network is partitioned ?
    - See next point...
- If BDN fails (OR there is a Broker network partition), discovery request may be propagated using multicast (if network configuration allows it)
- No deployment of special entities required for proximity analysis

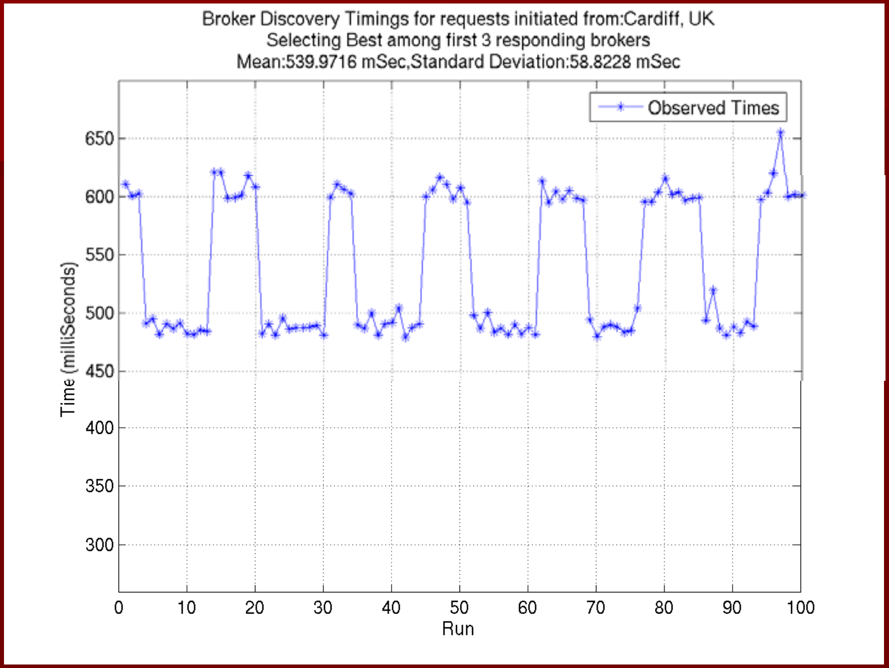
Advantages of our approach

- New brokers in the system are automatically incorporated to the discovery process
  - Since Broker Discovery response includes usage metrics, newly added brokers would be preferentially selected
- Private BDNs can be easily setup
- Approach ensures that the client connects to the nearest available broker if the client presents the right credentials

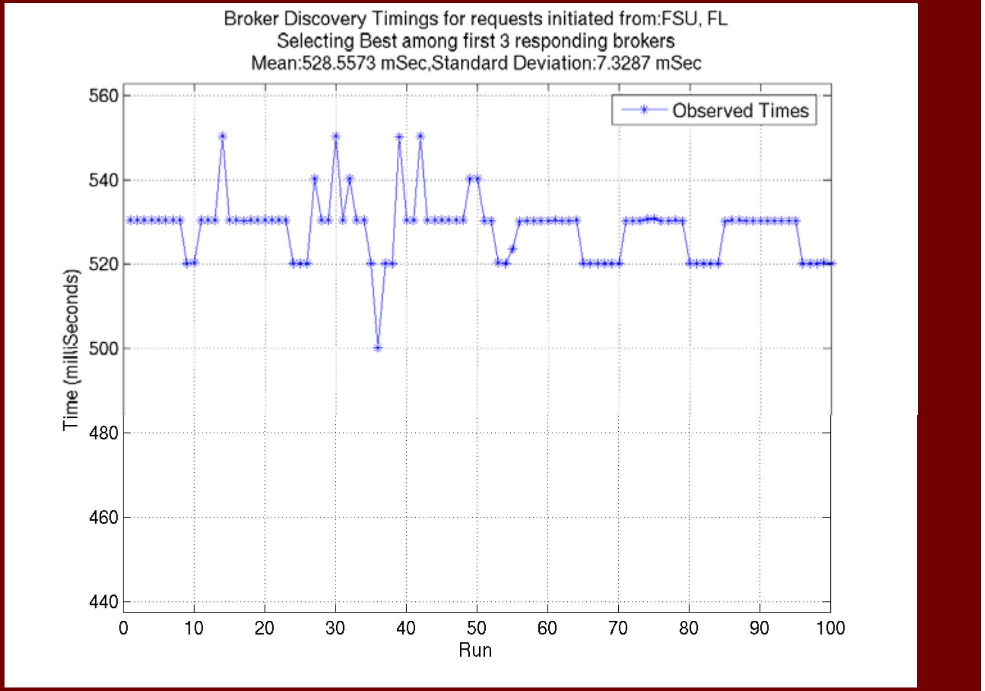
Some Results

- Distributed brokers in FSU (Tallahassee), NCSA, UMN (Twin cities), Cardiff (UK) and IU / IUPUI (Indianapolis)
  - High resolution timing (microsecond)
- Different topologies
  - Unconnected (None of the brokers connected to each other, All registered with BDN)
  - Linear (One broker registers itself with BDN, rest connected to this broker in a linear fashion)
  - Star (One broker registers itself with BDN, all others connected to this broker directly)
- Refer paper for complete set of results

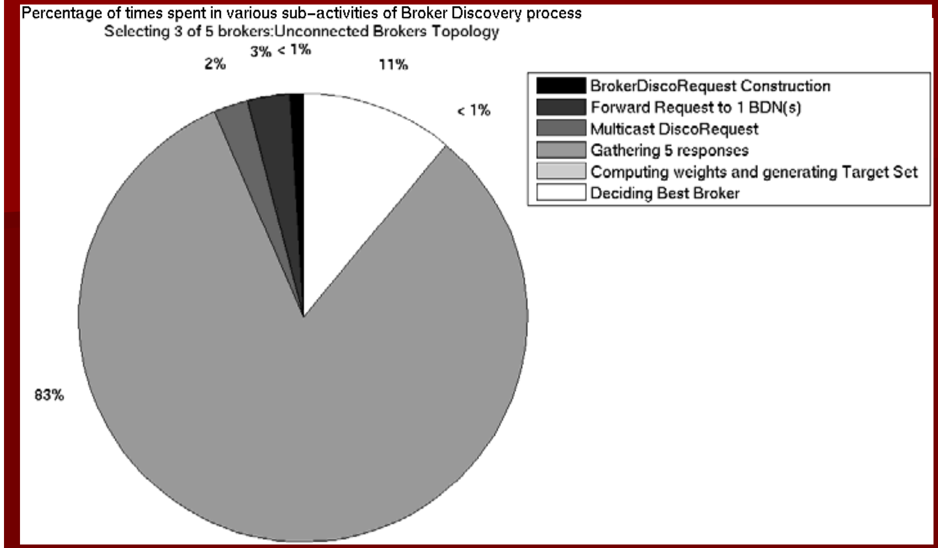
Results

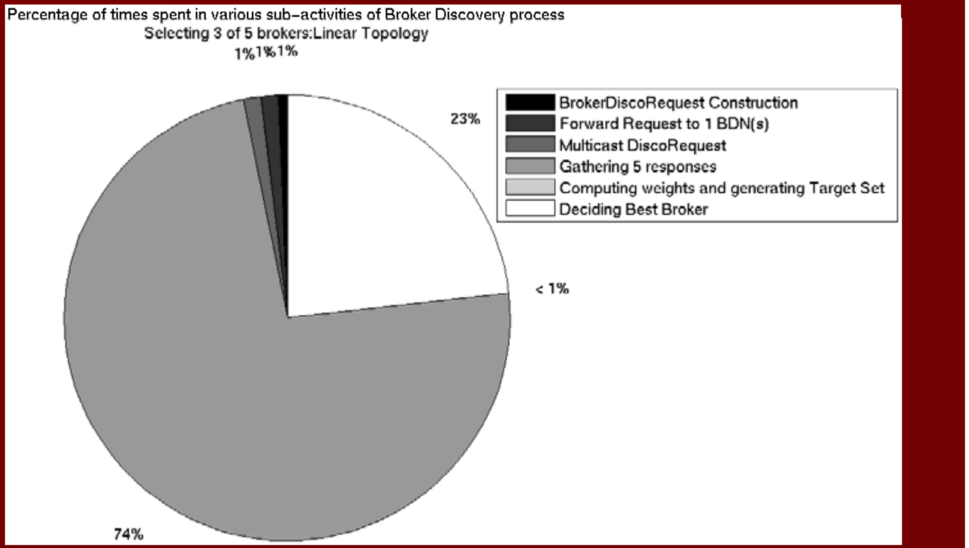


- Shown (Graphs for Discovery times when discovery is initiated from Cardiff, UK and FSU, FL)
- Average time to discover nearest broker
  - Approximately 350 – 550 mSec

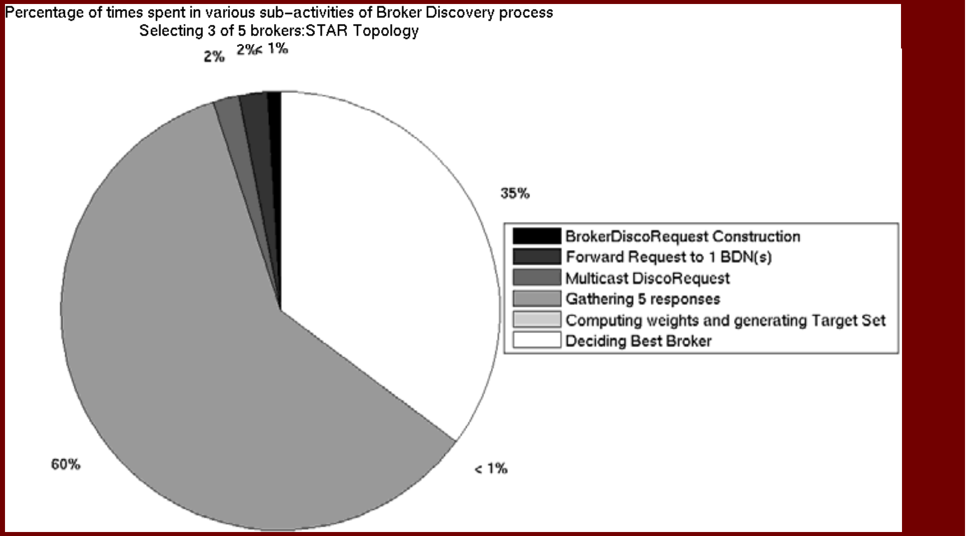


Results (contd.)

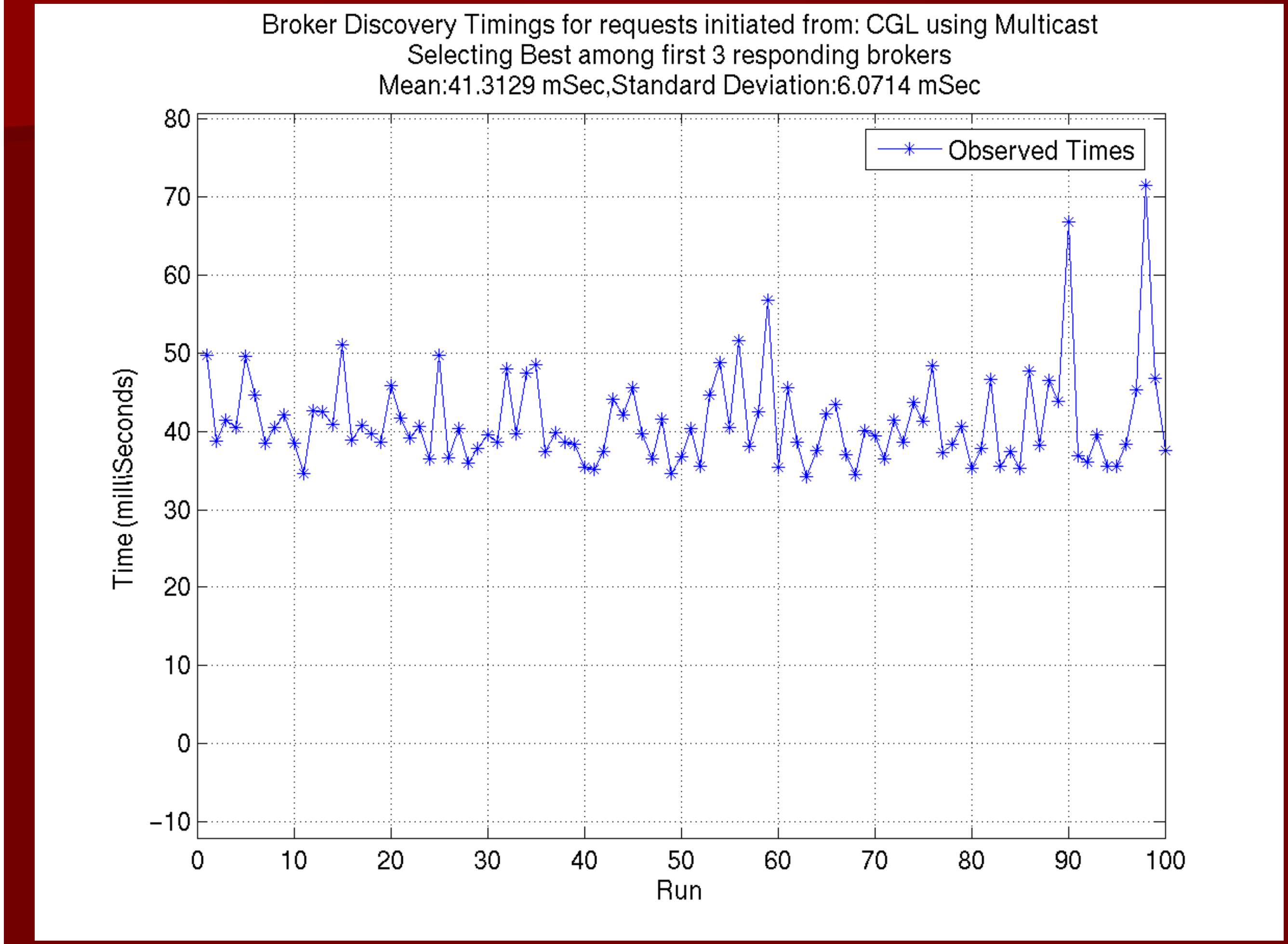

Percentage of times spent in various sub–activities of Broker Discovery process
Selecting 3 of 5 brokers:Unconnected Brokers Topology


Percentage of times spent in various sub–activities of Broker Discovery process
Selecting 3 of 5 brokers:Linear Topology

- Maximum time spent (60% – 80%) in gathering initial response to construct the BrokerTargetSet
  - Note this depends on how fast the discovery request propagates through the network which is dependent on the broker topology


Percentage of times spent in various sub–activities of Broker Discovery process
Selecting 3 of 5 brokers:STAR Topology

Results (contd.)



- Multicast whenever available (usually in local networks only) takes approximately abt. 40 mSec to find the nearest broker

Discussion

- Discovery process depends mainly on the network bandwidth.
- Maximum time spent in waiting for initial responses from brokers.
- Higher timeout
  - More time spent in overall discovery
  - BUT, more results gathered, possibly more accurate target set construction
  - NOTE: IF only few brokers exist OR only few brokers decide to respond then unnecessary waste of time
- Lower timeout
  - Less time spent in overall discovery
  - BUT, One risks gathering lesser number of responses from brokers
- Multicast works under the assumption that at-least 1 broker is reachable at the configured **`address:port`**

Discussion (contd.)

- Current scheme may be augmented with Security by encrypting Discovery Request/Response
- Requests from authorized clients only would be honored
- Not yet implemented but we tested to find the cost associated with such a scheme
  - Approx 6 mSec to validate a client certificate
  - Approx 25 mSec to Encrypt / Decrypt a Broker Discovery Request

Conclusion...

- Presented an architecture for discovering existing brokers
- Presented results in WAN (Wide area network) settings
- Security and private BDNs can easily be incorporated in our scheme

Acknowledgements

- Dr. Gordon Erlebacher (FSU)
- Dr. Mary Thomas (SDSU)
- Ben Kadlec (UMN)
- Cardiff, NCSA

Questions / Comments

- Any Questions / Comments ?

THANKS

for attending the presentation