NaradaBrokering and GlobalMMCS

22 July 2004

Grid Summer School

Geoffrey Fox

Community Grids Lab

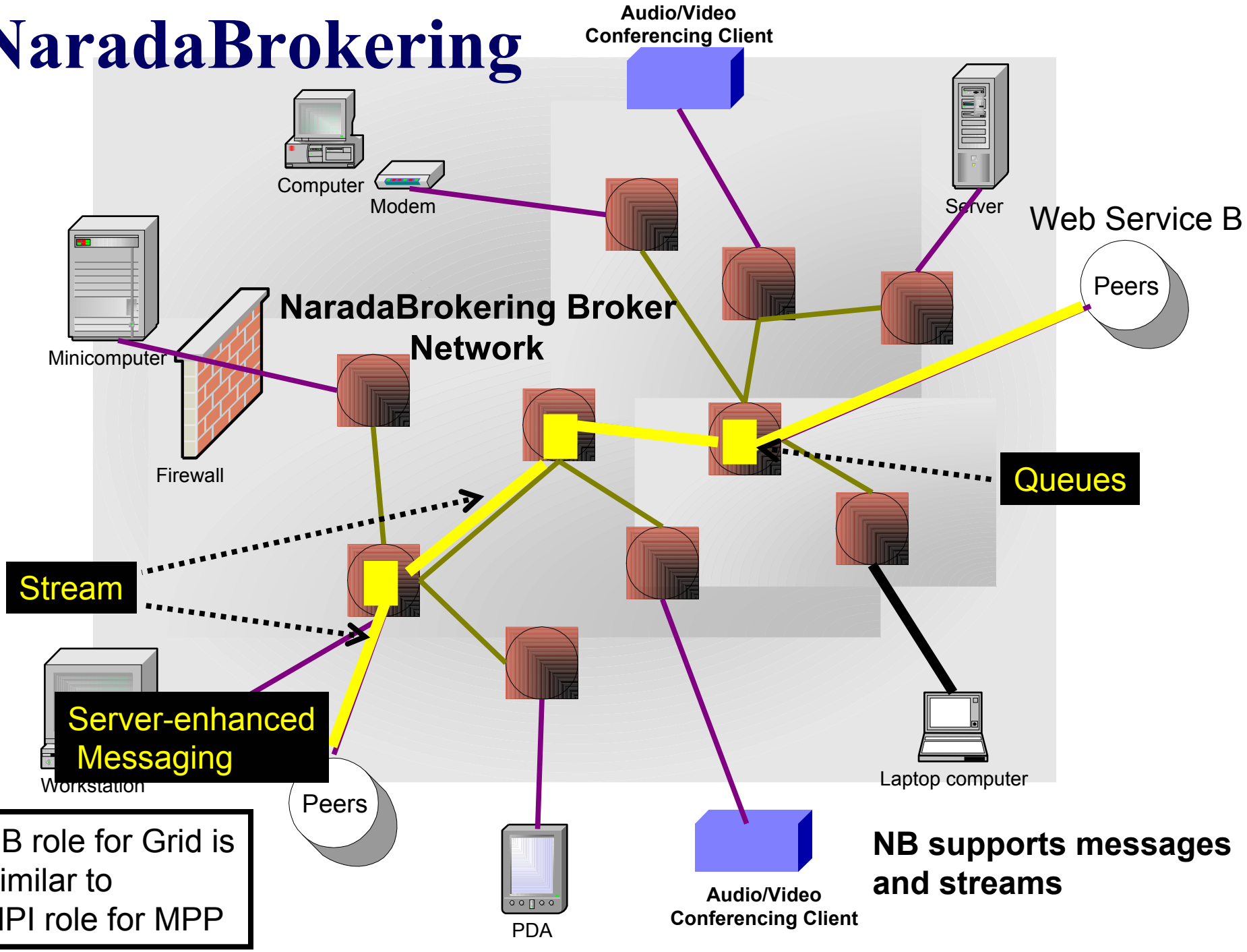
Indiana University

gcf@indiana.edu

NaradaBrokering can Support

- Grid Messaging reliably in spirit of WS-ReliableMessaging
- Virtualize inter-service communication
- Federate different Grids Together
- Scalable pervasive audio-video conferencing – “Video over IP”
- General collaborative Applications and Web services including e-learning, e-Sports and Internet multiplayer gaming
- Build next generation clients interacting with messages not method-based user interrupts
- Unify peer-to-peer networks and Grids
- Handle streams as in “media or sensor Grids”
- Handle events as in WS-Notification

NaradaBrokering



NB role for Grid is
Similar to
MPI role for MPP

NB supports messages and streams

"GridMPI" v. NaradaBrokering

- In parallel computing, MPI and PVM provided "all the features one needed' for inter-node messaging
- NB aims to play same role for the Grid but the requirements and constraints are very different
  - NB is not MPI ported to a Grid/Globus environment
- Typically MPI aiming at microsecond latency but for Grid, time scales are different
  - 100 millisecond quite normal network latency
  - 30 millisecond typical packet time sensitivity (this is one audio or video frame) but even here can buffer 10-100 frames on client (conferencing to streaming)
  - 1 millisecond is time for a Java server to "think"
- Jitter in latency (transit time through broker) due to routing, processing (in NB) or packet loss recovery is important property
- Grids need and can use software supported message functions and trade-offs between hardware and software routing different from parallel computing

NaradaBrokering

- 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.
- Now has several core functions
  - Reliable order-preserving "Optimized" Message transport (based on performance measurement) in heterogeneous multi-link fashion with TCP, UDP, SSL, HTTP, and will add GridFTP
  - General publish-subscribe including JMS & JXTA and support for RTP-based audio/video conferencing
  - Distributed XML event selection using XPATH metaphor
  - QoS, Security profiles for sent and received messages
  - Interface with reliable storage for persistent events

Laudable Features of NaradaBrokering

- Is open source http://www.naradabrokering.org
- Has client "plug-in" as well as standalone brokers
- Will have a discovery service to find nearest brokers
- Can communicate in firewall environments if you can launch browser and view sites (e.g. google.com)
- Supports uniform time across a distributed network
- Supports JXTA, JMS (Java Message Service) and more powerful native mode
- Transit time < 1 millisecond per broker
- Will have setup and broker network administration module

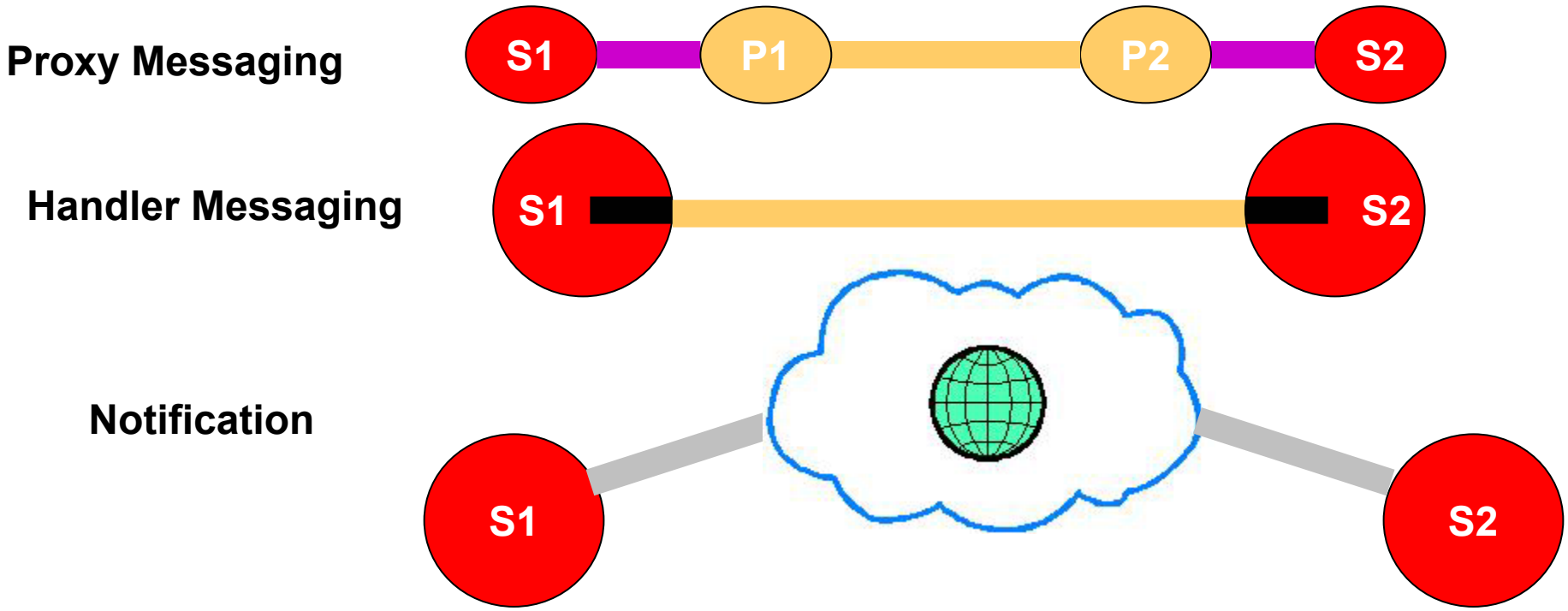
NaradaBrokering Naturally Supports

- **Filtering** of events to support different client requirements (e.g,. PDA versus desktop, slow lines, different A/V codecs)
- **Virtualization** of addressing, routing, interfaces
- **Federation** and **Mediation** of multiple instances of Grid services as illustrated by
  - Composition of **Gridlets** into full Grids (Gridlets are single computers in P2P case)
  - **JXTA** with peer-group forming a Gridlet
- **Monitoring** of messages for Service management and general autonomic functions
- **Fault tolerant** data **transport**
- **Virtual Private Grid** with fine-grain **Security** model

Current NaradaBrokering Features

Multiple transport support In publish-subscribe Paradigm with different Protocols on each link	Transport protocols supported include TCP, Parallel TCP streams, UDP, Multicast, SSL, HTTP and HTTPS. Communications through authenticating proxies/firewalls & NATs. Network QoS based Routing
Subscription Formats	Subscription can be Strings, Integers, **XPath** queries, **Regular Expressions**, **SQL** and tag=value pairs.
Reliable delivery	**Robust** and **exactly-once delivery** of messages 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 and Replay	**Recovery from failures** and disconnects. **Replay** of events/messages at any time.
Security	**Message-level WS-Security** compatible **security**
Message Payload options	Compression and Decompression of payloads Fragmentation a nd Coalescing of payloads
Messaging Related Compliance	Java Message Service (**JMS**) 1.0.2b compliant Support for routing P2P **JXTA** interactions.
Grid Application Support	NaradaBrokering enhanced **Grid-FTP**. Bridge to the **Globus TK3**.
Web Service reliability	Prototype implementation of **WS-ReliableMessaging**

NaradaBrokering Service Integration



S? **Service**

P? **Proxy**

Any Transport

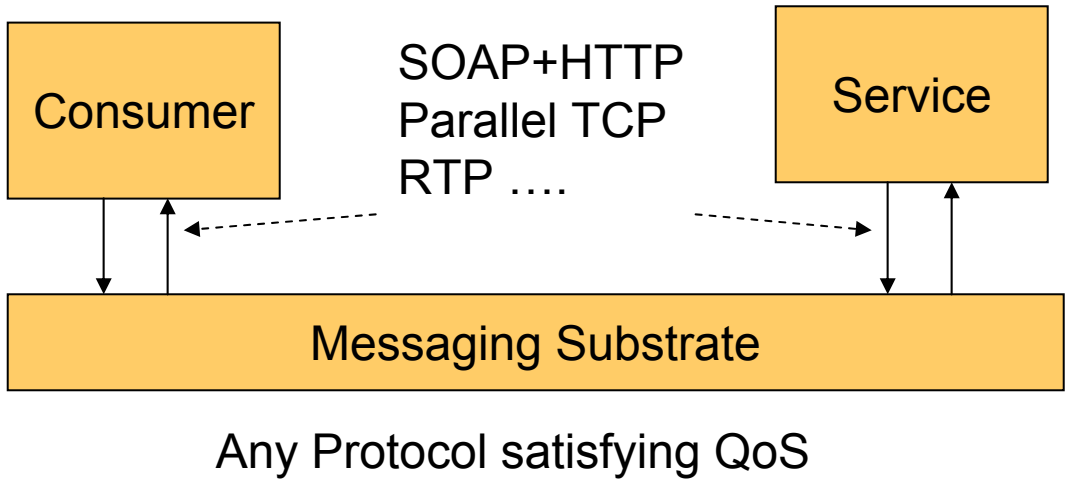
NB Transport

Standard SOAP Transport

Internal to Service: SOAP Handlers/Extensions/Plug-ins Java (JAX-RPC) .NET Indigo and special cases: PDA's gSOAP, Axis C++

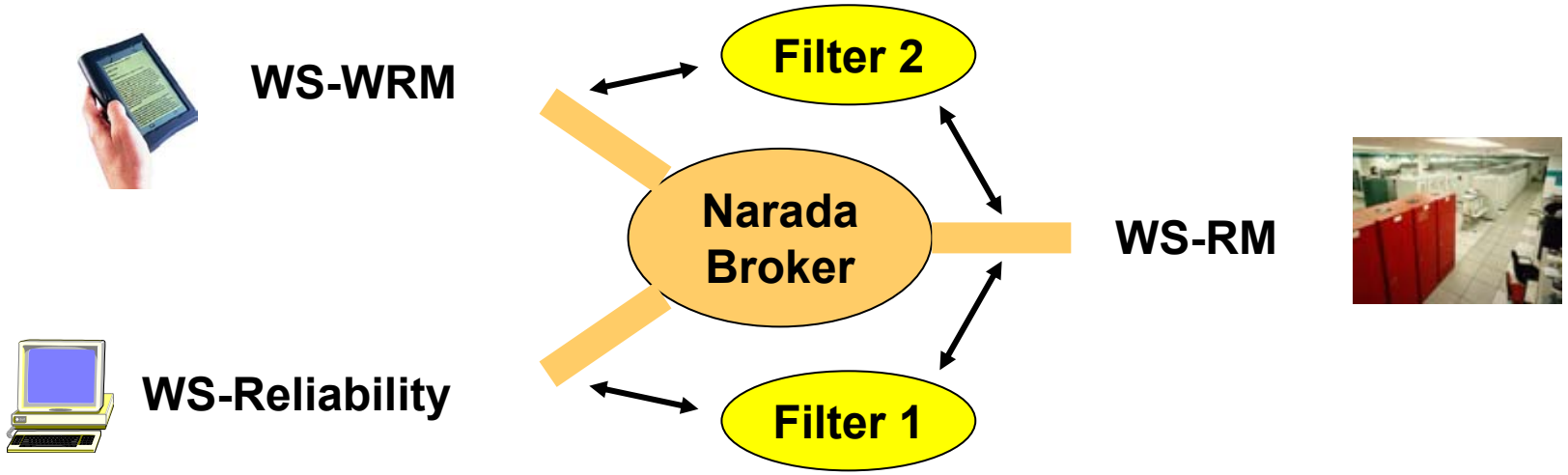
NB in the Transport Layer I

- Transport with optimizations for features such as performance, reliability, security using proxy or handler model
- Prototype of WS-RM (Reliable Messaging) using proxy
- GridFTP with NaradaBrokering transport (Parallel TCP plus WSRM)



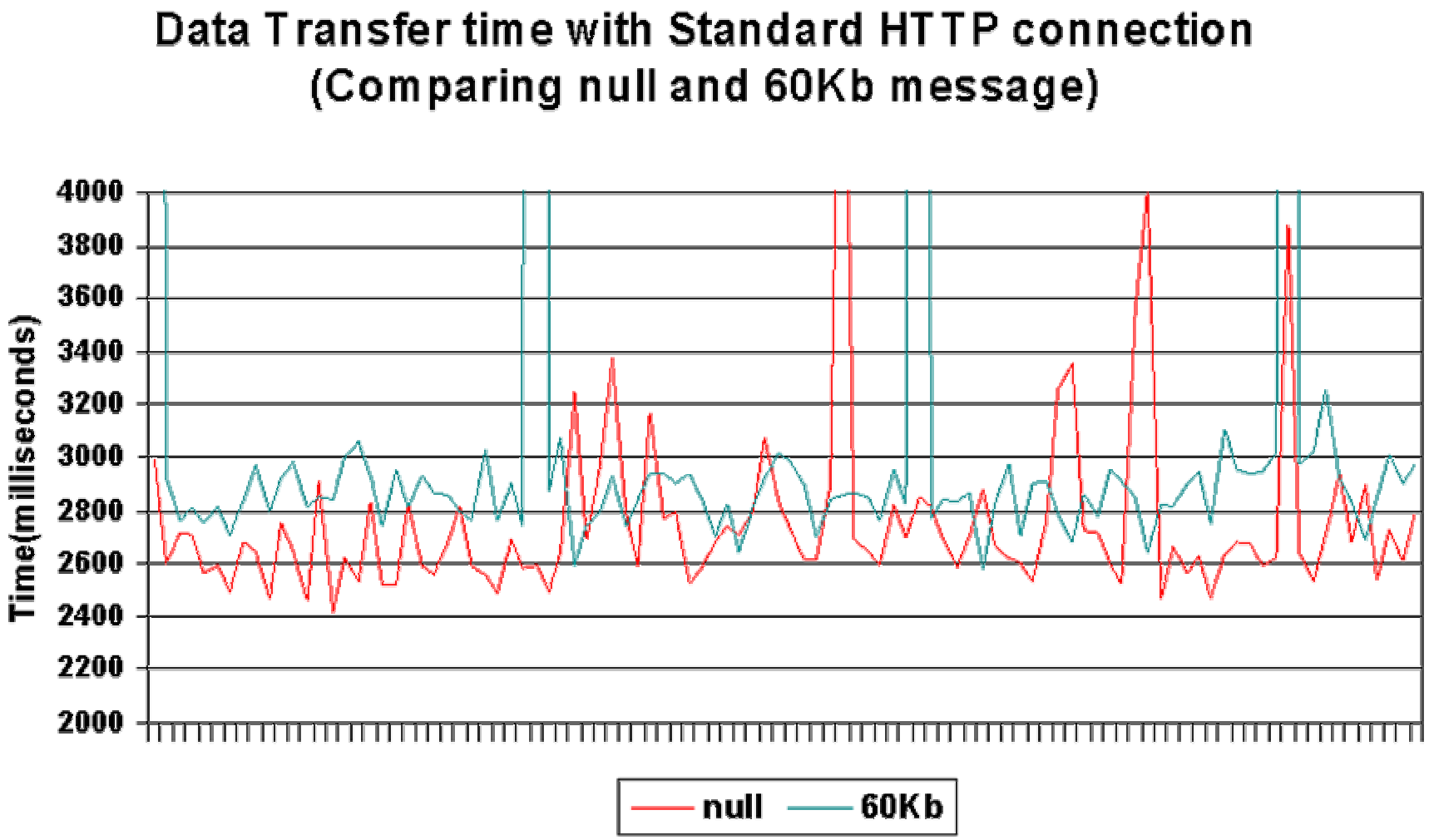
NB in the Transport Layer II

- Some PDA's have very asymmetric latency for Grid ⇔ PDA communication – we have designed a modified WSRM – WS-WRM wireless reliable messaging with different ack/nack choice
- Plan to support and federate WS-RM, WS-Reliability, WS-WRM



PDA Latency Measurement

- These show high PDA latency

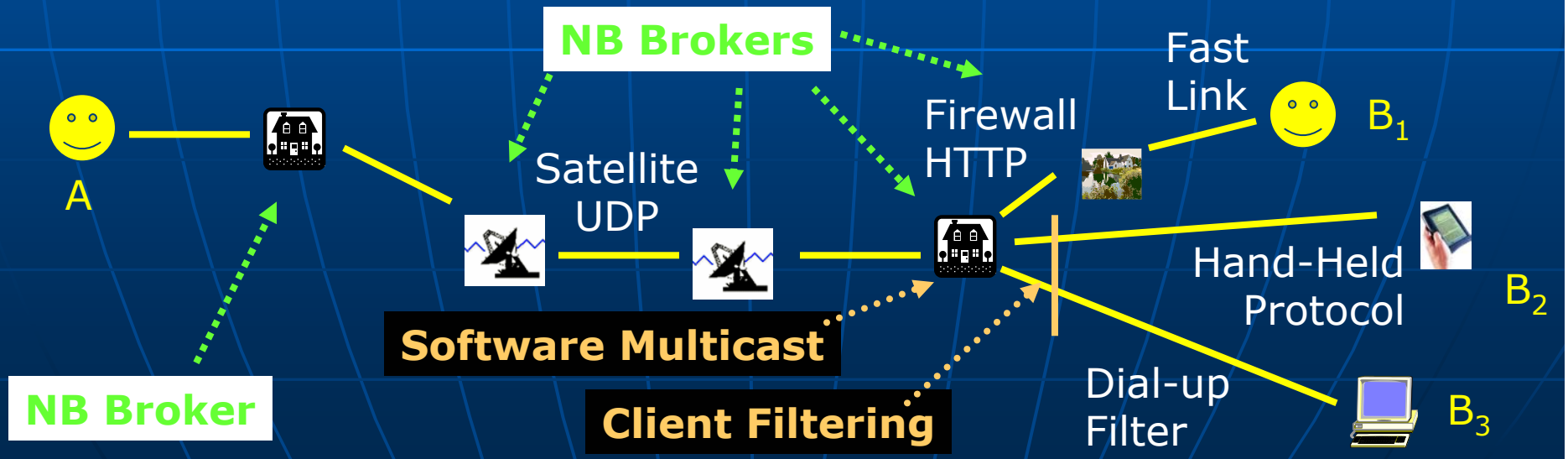


NB in the Transport Layer III

- We will add other higher performance protocols to NB transport options such as those based on UDP or modified TCP/IP
- We could support Virtual Private Network VPN to improve security (Virtual Private Grid)
  - more choice on firewall/NAT tunneling
- Currently NB has a rich set of firewall penetration options but these are not yet fully packaged with correct strategy to use

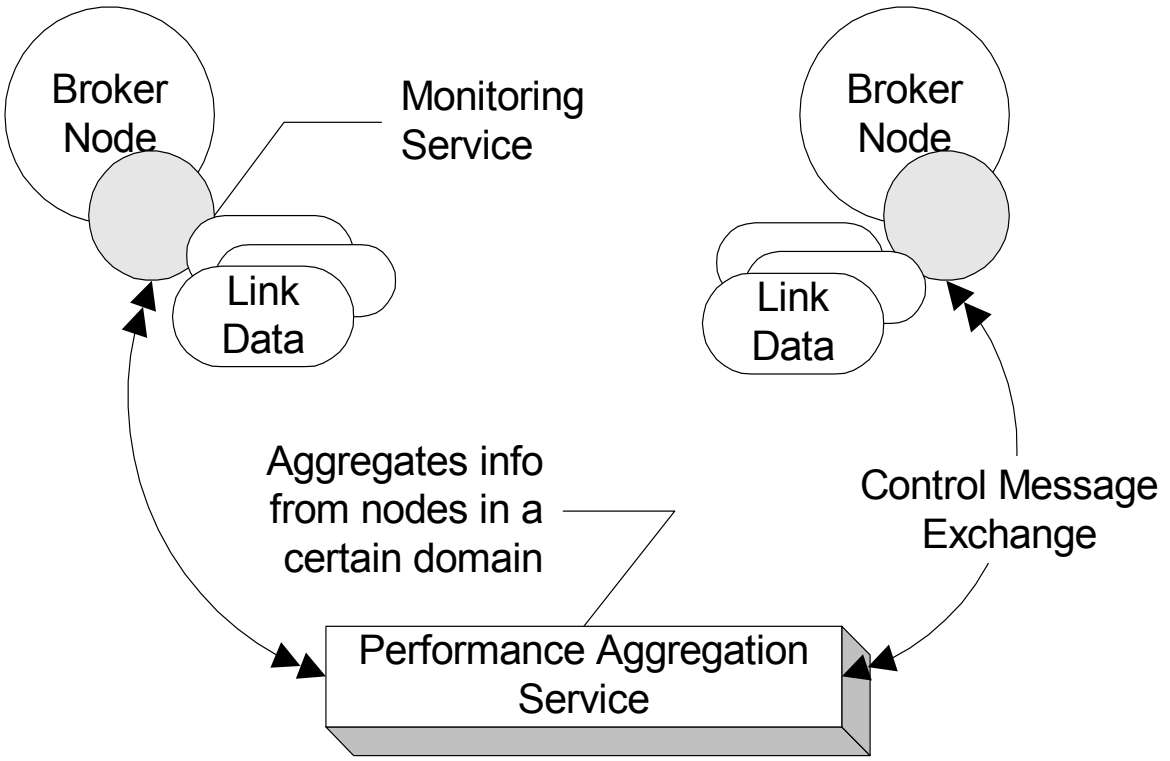
Virtualizing Communication

- **Communication** specified in terms of **user goal** and **Quality of Service** – not in choice of port number and protocol
- **Protocols** have become **overloaded** e.g. MUST use UDP for A/V latency requirements but CAN't use UDP as firewall will not support ……
- A given communication can involve **multiple transport protocols** and **multiple destinations** – the latter possibly determined dynamically



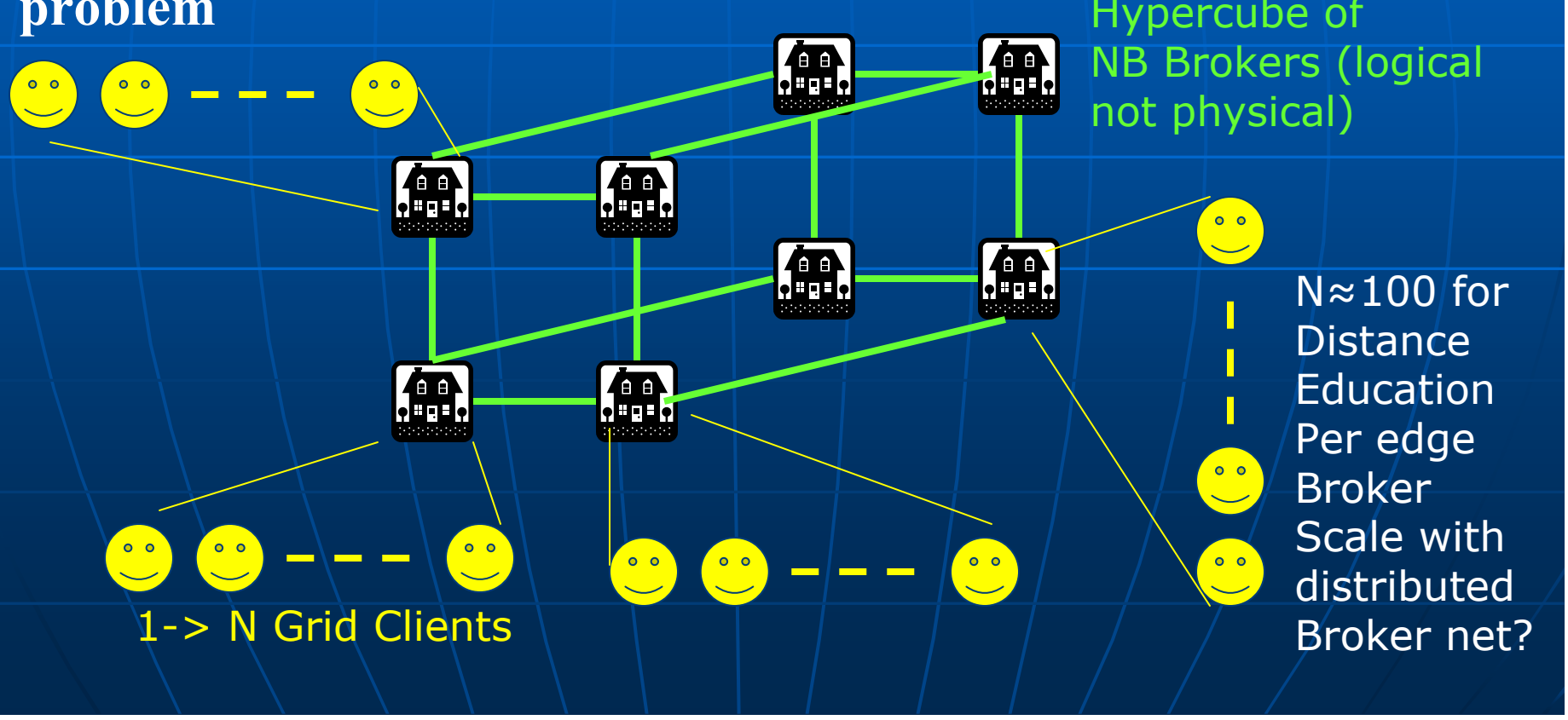
Performance Monitoring

- **Every broker** incorporates a **Monitoring service** that **monitors links** originating from the node.
- Every link measures and exposes a set of **metrics**
  - Average delays, jitters, loss rates, throughput.
- Individual links can **disable** measurements for individual or the entire set of metrics.
- Measurement intervals can also be varied
- Monitoring Service, returns measured metrics to **Performance Aggregator**.



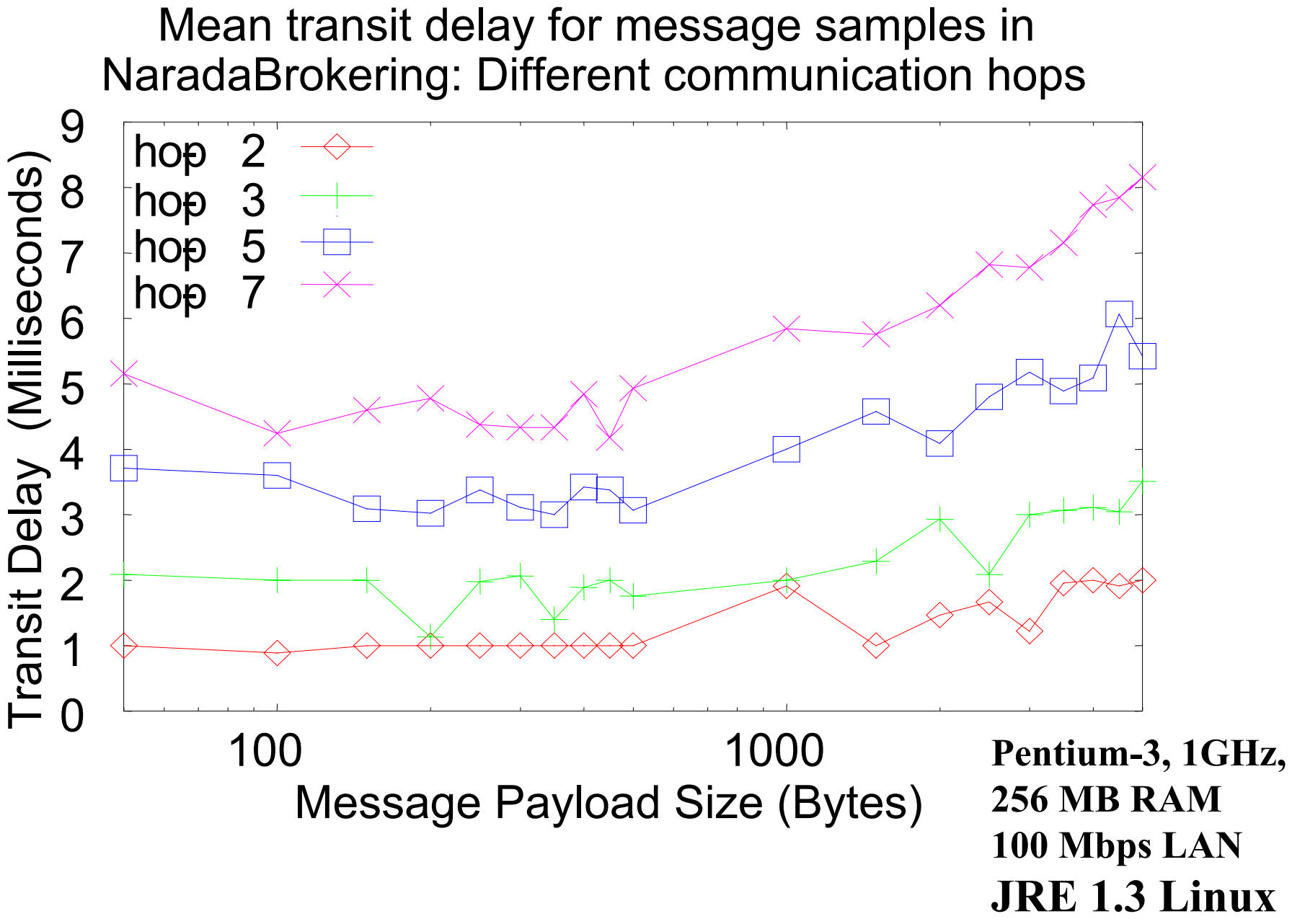
Architecture of Message Layer

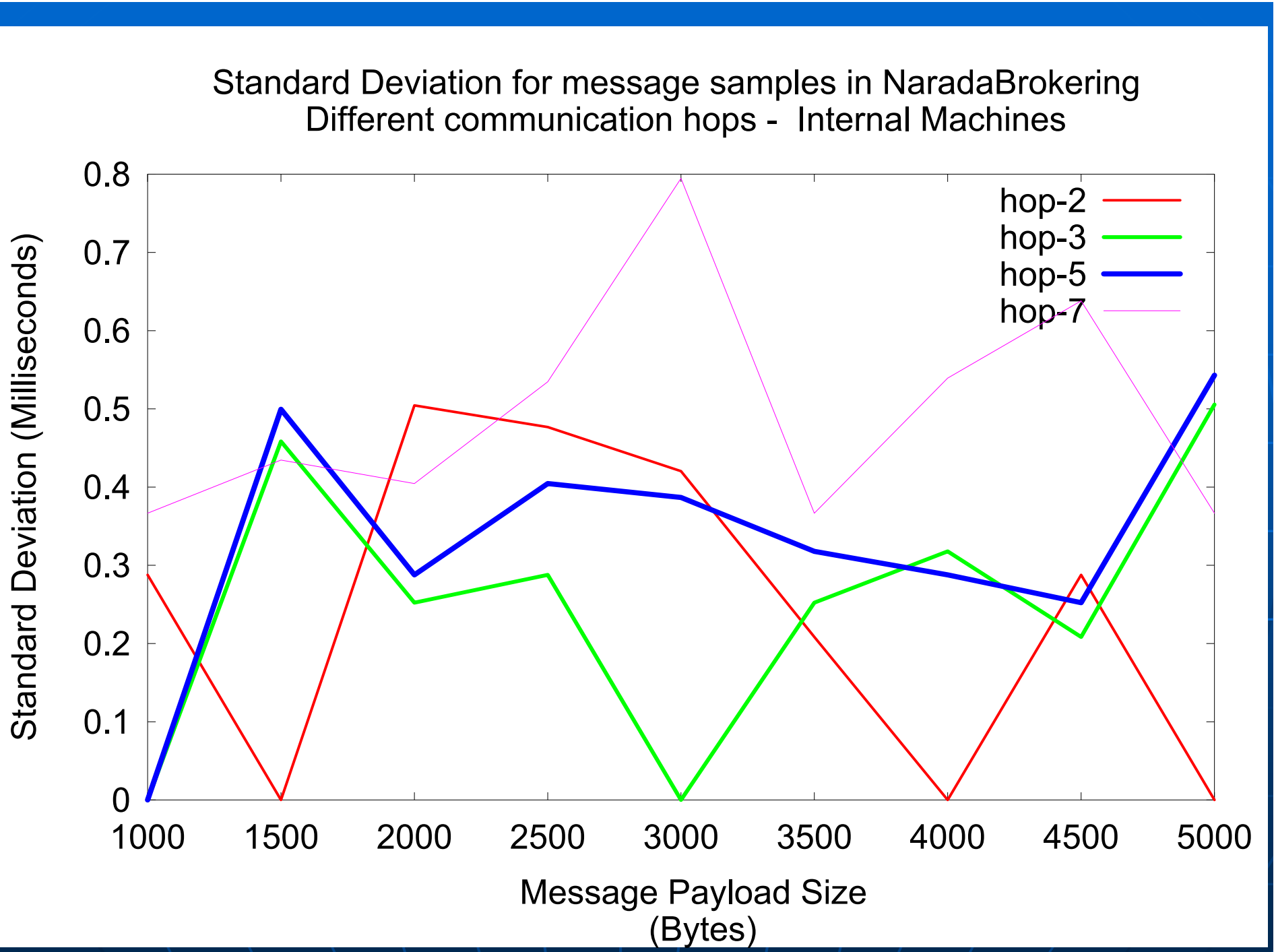
- Need to optimize not only routing of particular messages but classic publish/subscribe problem of integrating different requests with related topics (subscribe to sports/basketball/lakers and sports)
- Related to Akamai, AOL … caching and Server optimization problem

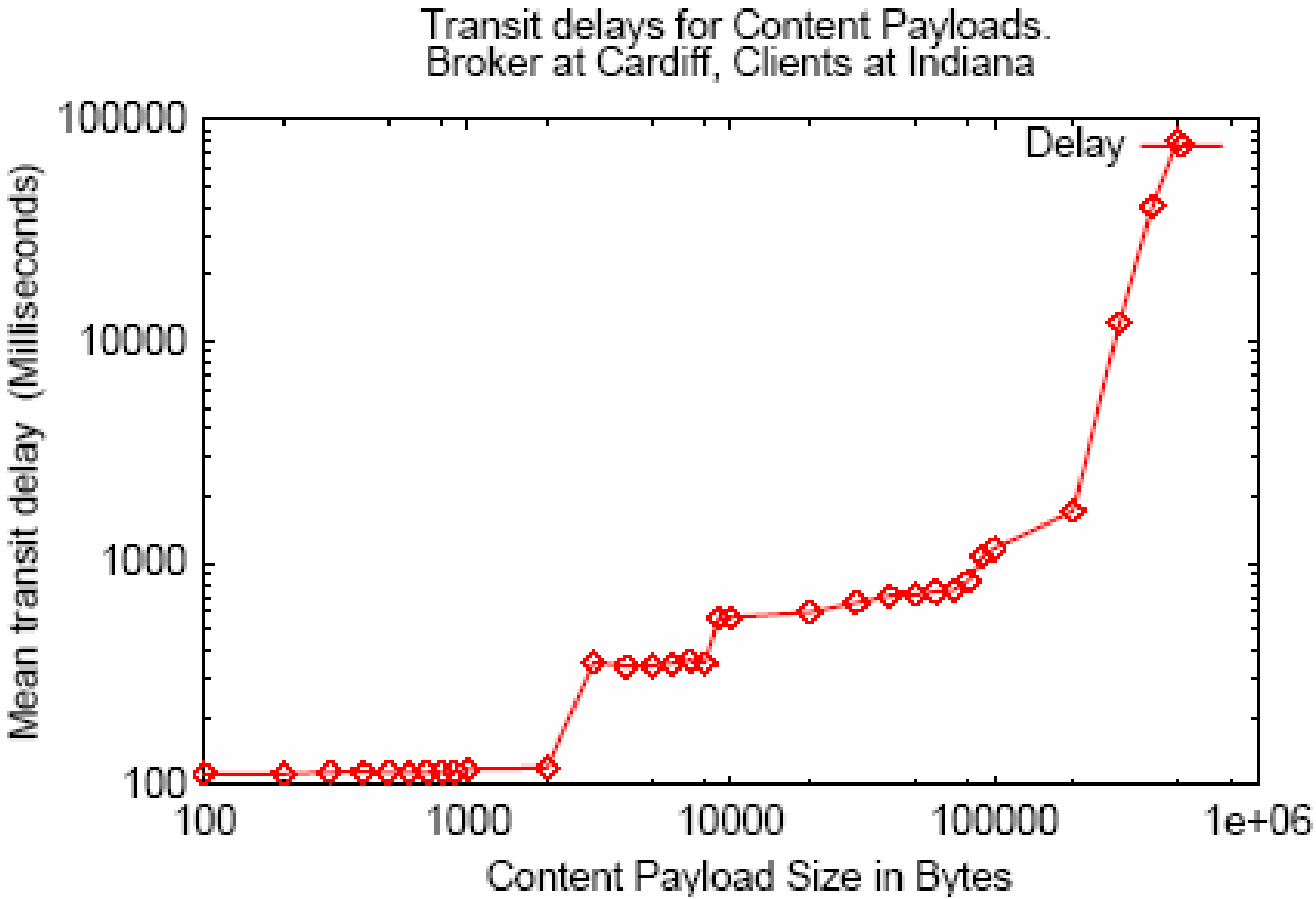


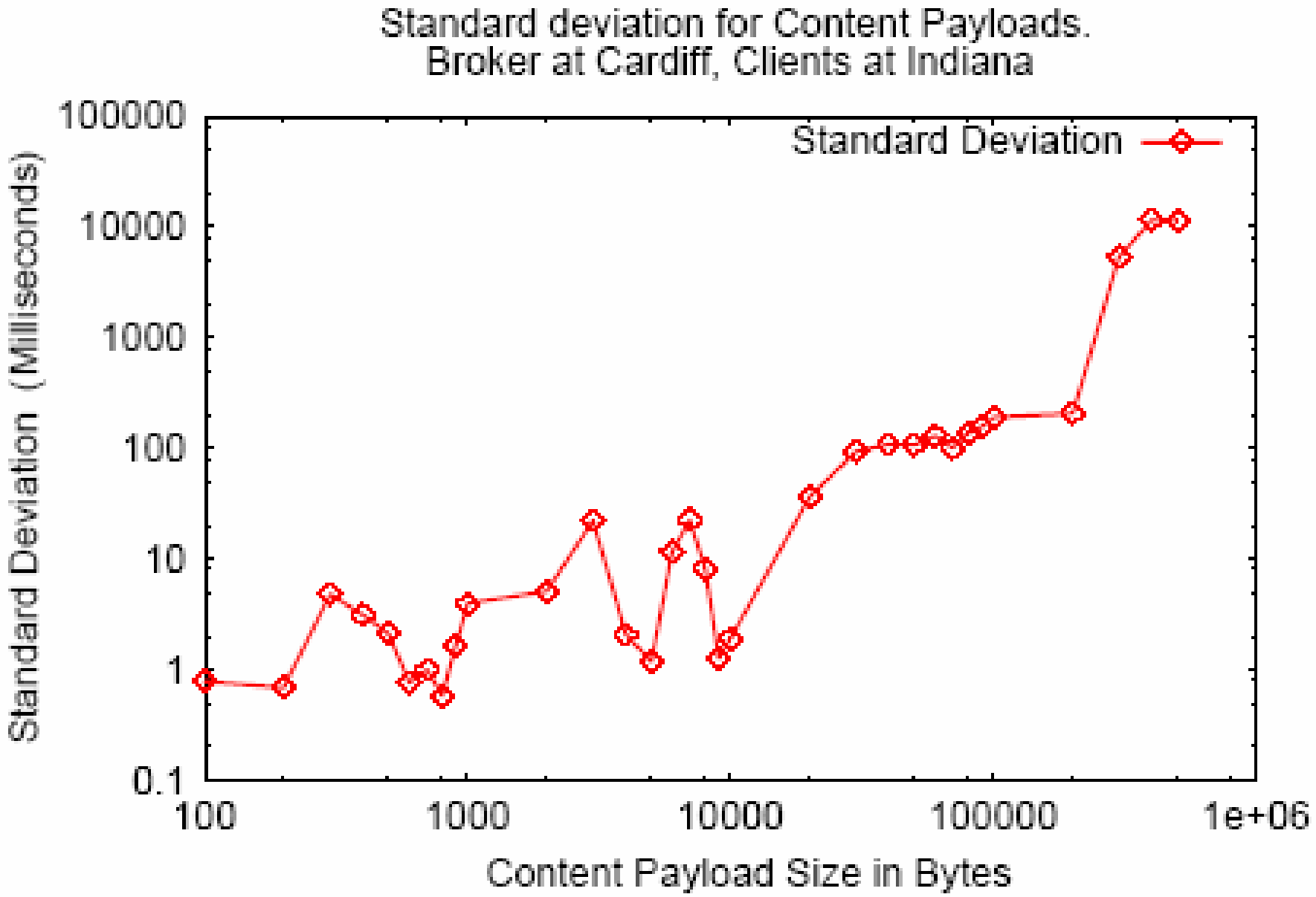
NaradaBrokering Communication

- Applications interface to NaradaBrokering through UserChannels which NB constructs as a set of links between NB Brokers acting as "waystations" which may need to be dynamically instantiated
- UserChannels have publish/subscribe semantics with XML topics
- Links implement a single conventional "data" protocol.
  - Interface to add new transport protocols within the Framework
  - Administrative channel negotiates the best available communication protocol for each link
- Different links can have different underlying transport implementations
  - Implementations in the current release include support for TCP,UDP, Multicast, SSL, RTP and HTTP.
  - GridFTP most interesting new protocol
  - Supports communication through proxies and firewalls such as iPlanet, Netscape, Apache, Microsoft ISA and Checkpoint.

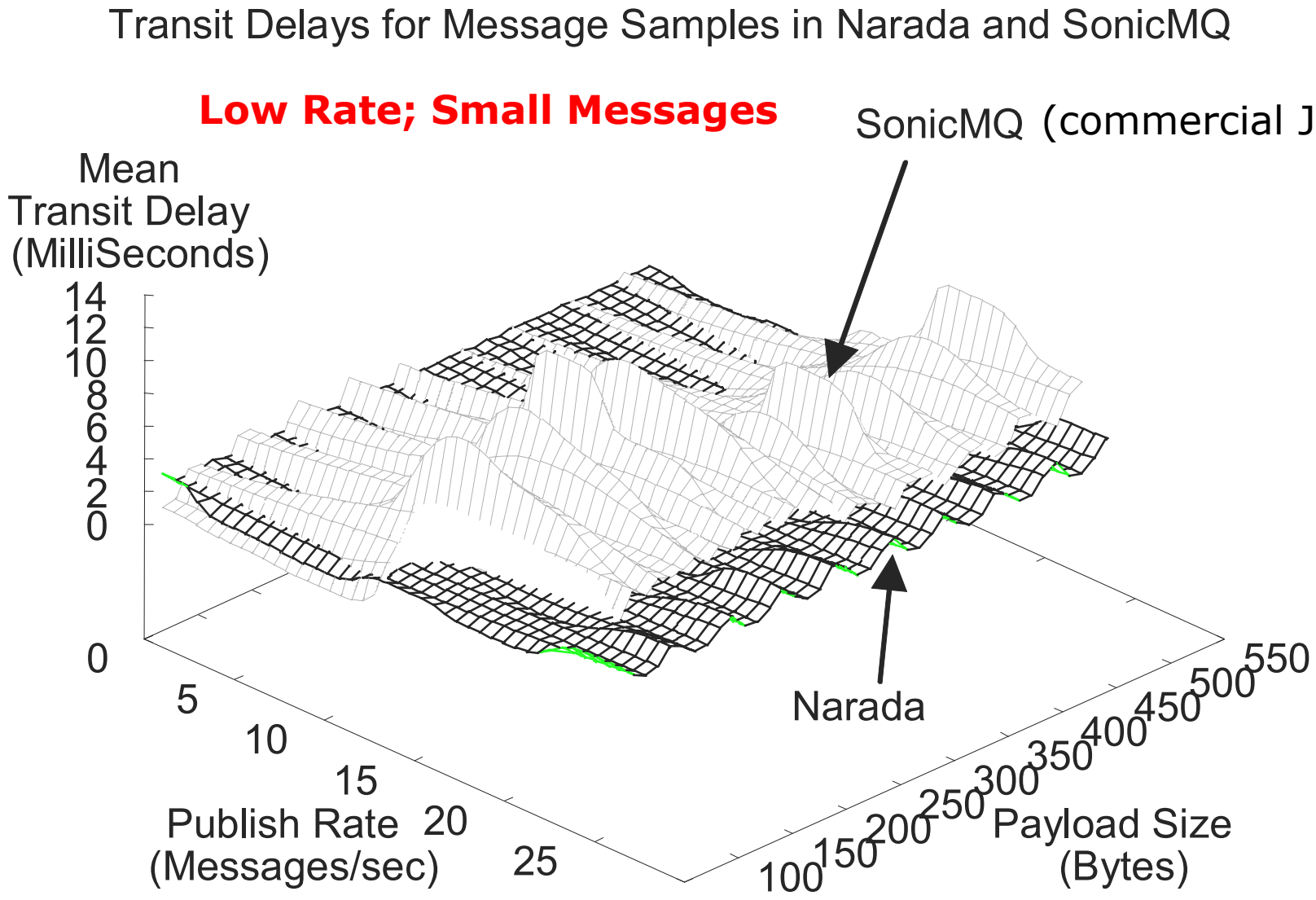

Mean transit delay for message samples in NaradaBrokering: Different communication hops
Transit Delay (Milliseconds)
0
1
2
3
4
5
6
7
8
9
hop 2
hop 3
hop 5
hop 7
100
1000
Message Payload Size (Bytes)
Pentium-3, 1GHz,
256 MB RAM
100 Mbps LAN
JRE 1.3 Linux


Standard Deviation for message samples in NaradaBrokering
Different communication hops - Internal Machines
Standard Deviation (Milliseconds)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1000
1500
2000
2500
3000
3500
4000
4500
5000
Message Payload Size
(Bytes)
hop-2
hop-3
hop-5
hop-7


Transit delays for Content Payloads.
Broker at Cardiff, Clients at Indiana
Mean transit delay (Milliseconds)
100000
10000
1000
100
Delay
100
1000
10000
100000
1e+06
Content Payload Size in Bytes

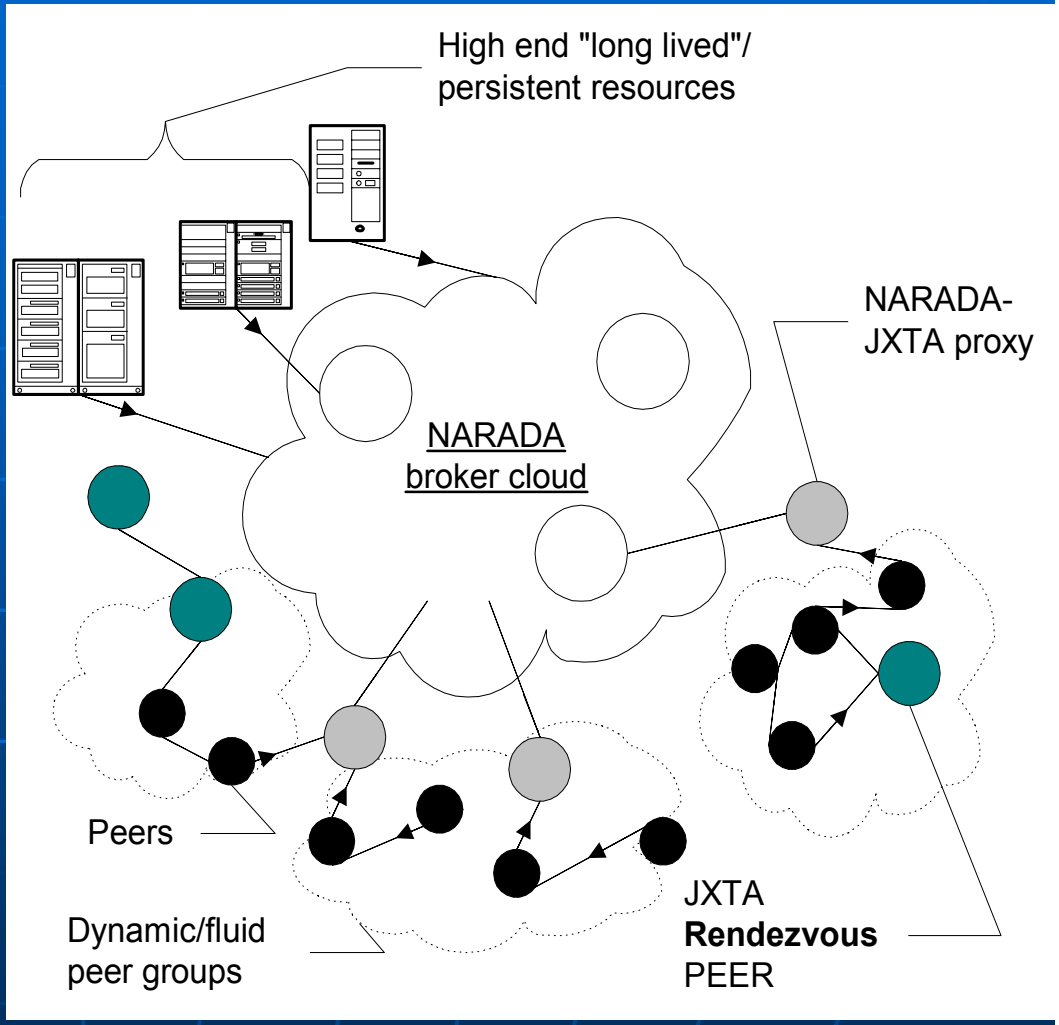

Standard deviation for Content Payloads.
Broker at Cardiff, Clients at Indiana
Standard Deviation
Standard Deviation (Milliseconds)
100000
10000
1000
100
10
1
0.1
100
1000
10000
100000
1e+06
Content Payload Size in Bytes

NaradaBrokering and JMS (Java Message Service)



NaradaBrokering and JXTA Federation

- Based on **hybrid proxy** that acts as both **Rendezvous** peer (JXTA routers) and NaradaBrokering end-point.
- **No changes to JXTA core** or constraints on interactions
  - Change made to Rendezvous layer
- **Peers are not aware** that they interact with a Narada-JXTA proxy or Rendezvous peer.



- NB provides JXTA guaranteed **long distance delivery**
- NB **federates** multiple JXTA Peer Groups

End-point Services in Native NaradaBrokering

- Allows you to create Consumers (subscribers) of events (an event is a time stamped message where time stamp can be empty!)
- Allows you to create Producers of events (publishers)
- Allows you to discover brokers and initialize communications with the broker.
- Services available at the client side will perform
  - Compression of payloads
  - Computation of Message digests for Integrity
  - Secure encryption of payload based on the specified keys
  - Fragmentation of large payloads into smaller packets
  - Redundancy service which maintains active (alternate) connections to multiple brokers.

Event Consumer Capabilities

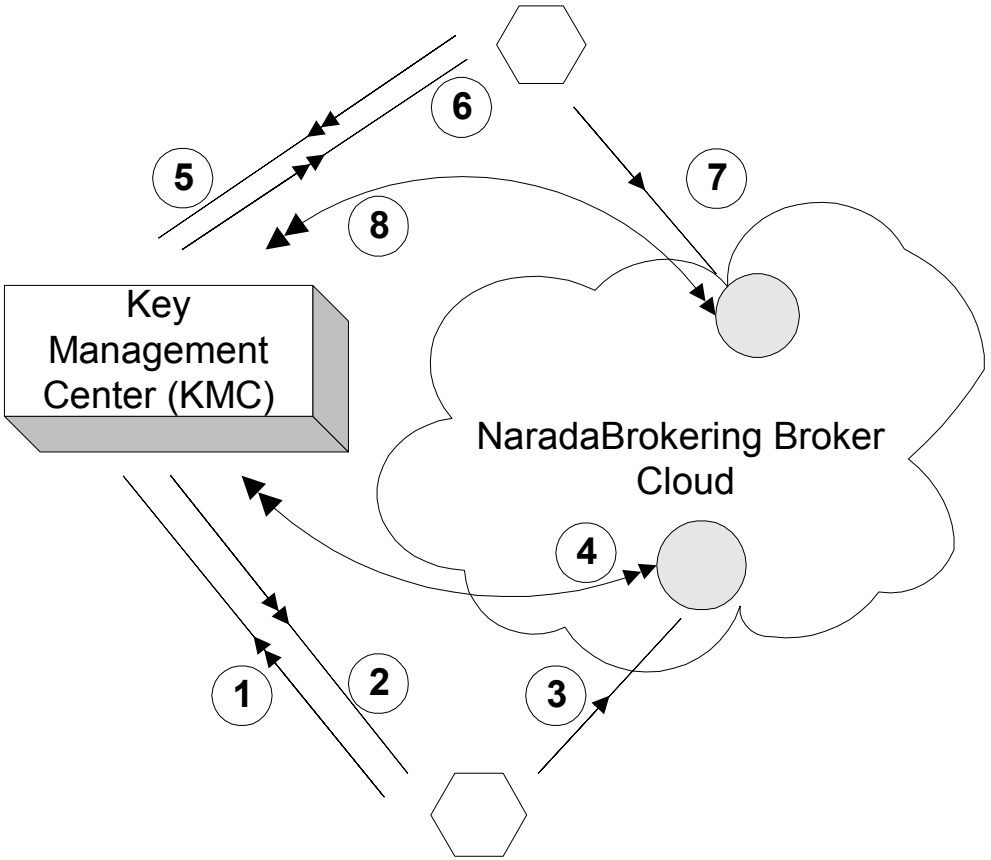
- Allow you to subscribe to events that conform to a certain template.
  - The specified subscription profile could topic-based strings, XPath queries, <tag=value> pairs or integer topics.
- Event Consumers can also create Consumer constraints to specify various properties regarding the delivery of events.
- Consumer constraints are different from subscriptions.
  - Subscriptions (or Profiles) are evaluated in a distributed fashion by the broker network,
  - Consumer constraints are QoS related and are managed by the QoS services running on the end-point.
- Consumer constraints can specify
  - Reliable Delivery of events
  - Ordered (Publisher, causal and time ordered) delivery of events
  - Exactly once delivery of events
  - Delivery after un-compression of compressed payload
  - Delivery after decrypting encrypted payload

Event Producer Capabilities

- Facilitate the generation of events in correct format (next slide)
- Facilitate the publishing of events to brokers
- Allow the creation of Publisher constraints which facilitate specification of properties that need to be satisfied by published events
- Among the constraints that can be specified include
  - Method of Securing message payloads
  - Computing message digests
  - Compressing message payloads
  - Fragmenting large payloads

Native NaradaBrokering Event

- The event comprises of
  - Event headers
  - Content Synopsis (for selection as in JMS properties WITHOUT reading body)
  - Content Payload
  - Dissemination Traces (generated on the fly as event traverses broker network)
- This is different from structure of JMS or JXTA events
- This NBEvent structure supports the extra capabilities discussed earlier
- The event headers specify information regarding
  - Security and Integrity of encapsulated payload
  - Fragmentation of events
  - Compression of payloads
  - Correlation identifiers (to define ordering between different streams as is needed in some collaboration applications)
  - Priority
  - Application Type
  - Event Identifiers



1 — Request permission to publish

2 — Respond back with topic key if authorized to publish

3 — Encrypt message with topic key
Compute Message Digest(MD)
Sign MD and message ID
Publish Message

4 — Verify Signature & Permissions
Check integrity by verifying MD
Check ID for replay attacks

5 — Request permission to subscribe

6 — Respond back with topic key if authorized to subscribe

7 — Create subscription request
Compute Message Digest
Sign MD and message ID
Issue Subscription request Message

8 — Verify Signature
Verify Permissions for Subscribing
Check integrity by verifying MD
Check ID for replay attacks

Broker Node

Entity (Publisher or Subscriber)

SSL encrypted communications

Based on Message Level Security

Messages organized into topics

Each topic has a separate key; Topics can be organized into sessions

Functionality I	WebSphere MQ (formerly MQSeries)	Pastry	NaradaBrokering
Maximum number of nodes hosting the messaging infrastructure	Medium (MQ is based on the point-to-point model. There is a limit on the effectiveness of this mode in large configurations).	Very large	Very large
JMS Compliant	Yes	No	Yes
Guaranteed Messaging (Robust)	Yes	Yes	Yes
Support for routing P2P Interactions	No	Yes	JXTA and later Gnutella
Support for Audio/Video Conferencing & raw RTP clients	No	No	Yes
Communication through proxies and firewalls	Yes	No	Yes
Support for XPath queries/ subscriptions	No	Yes	Yes
end-to-end Security	Yes	No	Yes
Network Performance Monitoring	No	No	Yes

Functionality II	WebSphere MQ (formerly MQSeries)	Pastry	NaradaBrokering
Workflow Support	Yes	No	No
Support for P2P distributed caching	No	Yes (Squirrel)	No
Platforms or Hosting Environments	35 different OS/ platforms supported. Also supports the Java Platform.	Supported on platforms which support C# (Microsoft) or Java (Rice).	Platforms supporting Java 1.4 (tunneling C++)
Maturity of Software	Extremely mature, with very robust diagnostic information	Fair	Fair with some "production" testing
Transport Protocols Supported	TCP, HTTP, Multicast, SSL, SNA etc.	TCP, UDP	TCP (Blocking and non-blocking), Parallel TCP, UDP, Multicast, HTTP, SSL, RTP, (GridFTP)
Multiple transport protocols over multiple hops.	Yes	No	Yes
Broker Network Design Interface	No	No	In Progress

WS-Reliability & WS-RM

- There are two rival reliable messaging specifications for Web Services that provide reliable delivery between two endpoints.
- Both the specifications use positive acknowledgements to ensure reliable delivery. WSRM recently has incorporated support for NACKS.
- Both specifications include support for faults
- WS-Reliability is a SOAP based protocol
- WS-ReliableMessaging provides an XML schema for reliable messaging.
  - Includes a SOAP binding.

NaradaBrokering & Reliable Delivery specifications

- We can provide support for both these specifications
  - In NaradaBrokering we provide reliable delivery from multiple points to multiple points
- We have identified issues that will allow federation between these specifications
  - Sequence numbering, fault mappings, numbering rollovers, quality of service guarantees
- Federation would allow
  - WSRM sender & WS-Reliability receiver
  - WS-Reliability sender & WSRM receiver

NaradaBrokering, WS-Notification & JMS

- NaradaBrokering is JMS compliant
- Topics in NaradaBrokering could be based on XML, String(as in JMS), Plain text, Integers, and (tag=value) tuples.
  - Subscriptions could be XPath queries, SQL queries, Regular expressions, Strings and integers
- Almost all the primitives needed in WS-Notification are available in NaradaBrokering
  - Exception: Entities never communicate directly with each other, as proposed in WS-Notification.
  - We are either allow such direct communication or mimic in NB – no performance overhead!
- We are currently building a prototype implementation of WS-Notification
- Need to relate WS-Notification with WS-Eventing and WS-Events

NaradaBrokering and NTP

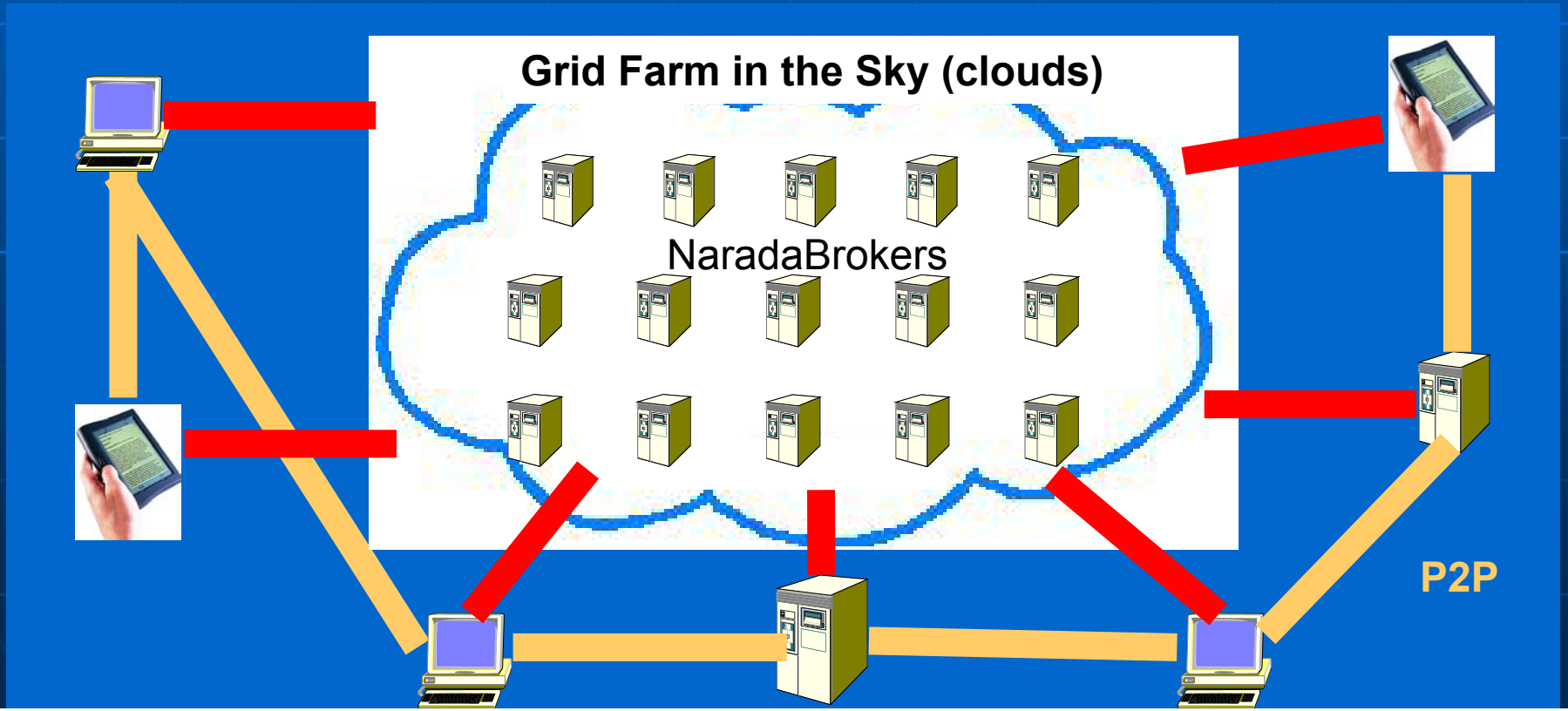
- NaradaBrokering includes an implementation of the Network Time Protocol (NTP)
- All entities within the system use NTP to communicate with atomic time servers maintained by organizations like NIST and USNO to compute offsets
  - Offset is the computed difference between global time and the local time.
  - The offset is computed based on the time returned from multiple atomic time servers.
    - The NTP algorithms weighs results from individual time clocks based on the distance of the atomic server from the entity.
- This ensures that all entities are within 1 millisecond of each other.
- The timestamps account for clock drifts that take place on machines
  - Time returned is based on software clocks which can slow down with increased computing load on the machine.

Higher Level NB Capabilities

- Could presumably have a Perl Interface for WSRF:Lite
- Could federate between WS-Notification, JMS (which it already supports) and WS-Eventing (Microsoft)
- Using filters, it can be used to
  - Reduce image size for PDA
  - Convert Access Grid A/V to a form suitable for some PDA's (RealNetworks, Windows Media)
  - Move XML between SOAP header and body to federate between WS-RF and WS-I
- Could supply WS-Security if used in handler mode
- Could support replicated Subscriber (Fault-tolerance/ Performance) Services
  - NaradaBrokering will choose between several subscribing replicated services

P2P and NaradaBrokering I

- **Server/Broker-free** version to support **"immediate deployment" of NB-based Community Grids** using **P2P** versions of Grid applications
- **Initially:** Use a **broker free** version of NB and **file-based Web services**
  - If successful, add **brokers in the Grid sky** to achieve better **performance** (if broker has better network link than clients)
- Service providers and supercomputer/national grid centers could sell such **Grid Farm** services



P2P and NaradaBrokering II

- Add DHT (Distributed Hash Table) approach (used in latest JXTA) in NB. NB nodes will have Ids that can determine where a specific content would be stored.
  - Provides scalable location of content
- 3 models of Information Systems
  - DHT for stable large volume distributed information
  - Fault Tolerant Metadata Catalog – subscribe multiple instances of metadata service to the MetadataCatalog topic – publish queries to this replicated subscriber topic
  - Flooding if all else has failed
- GridTorrent: Merge NB-enhanced GridFTP and P2P BitTorrent http://bitconjurer.org/BitTorrent/ to provide WSRM fault tolerant Parallel TCP P2P or Grid file transfer
  - BitTorrent supports fragmented distributed files which are natural WSRM and NB architecture
  - Don't really want GridFTP server; prefer to use fault tolerant GridTorrent metadata service (as above)

Fault Tolerant P2P e-Science Grid

- NaradaBrokering could provide several features of value to say particle physics Grid
- Supports Web Service standards: WS-RM, WS-Security, WS-Notification etc.
- Provide fault tolerant NaradaBroker network
- DHT provides worldwide scalable base information system
- Replicate all important services: RGMA, SRB, GRAM ..
- Associate each service with a topic
  - Replicated services subscribe to topic
  - Network and load QoS based choice of service
- GridTorrent file transfer automatically provides distributed fault tolerant caching – a better RLS

NaradaBrokering Futures

- Support for WS-Notification, WS-Eventing and federation between these schemes.
- Production release of discovery of “nearest” brokers and automated setup of distributed broker networks.
- Support for WS-Reliability, and federation between WSRM and WS-Reliability
- Support for ad-hoc networks
- Replicated resource management and redundancy.

Collaboration and Web Services

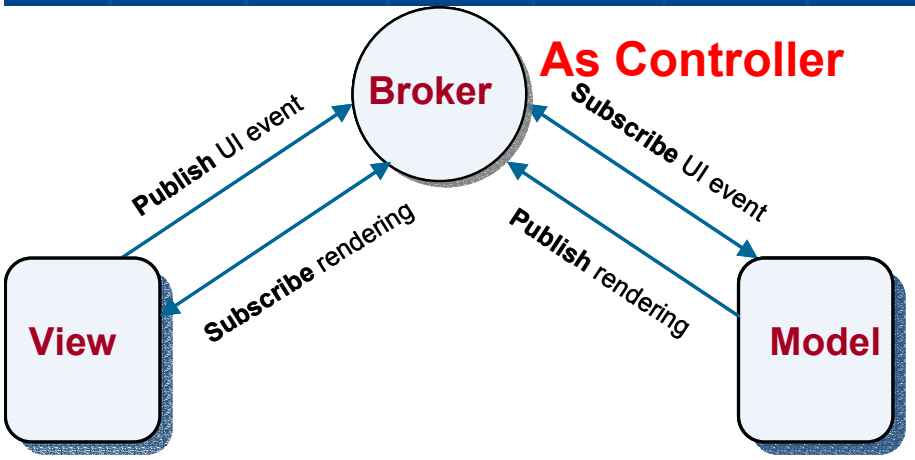
- Collaboration has
  a) Mechanism to set up members (people, devices) of a "collaborative sessions"
  b) Shared generic tools such as text chat, white boards, audio-video conferencing
  c) Shared applications such as Web Pages, PowerPoint, Visualization, maps, (medical) instruments ….
- b) and c) are "just shared objects" where objects could be Web Services but rarely are at moment
  - We can port objects to Web Services and build a general approach for making Web services collaborative
- a) is a "Service" which is set up in many different ways (H323 SIP JXTA are standards supported by multiple implementations) – we should make it a WS

Shared Event Collaboration

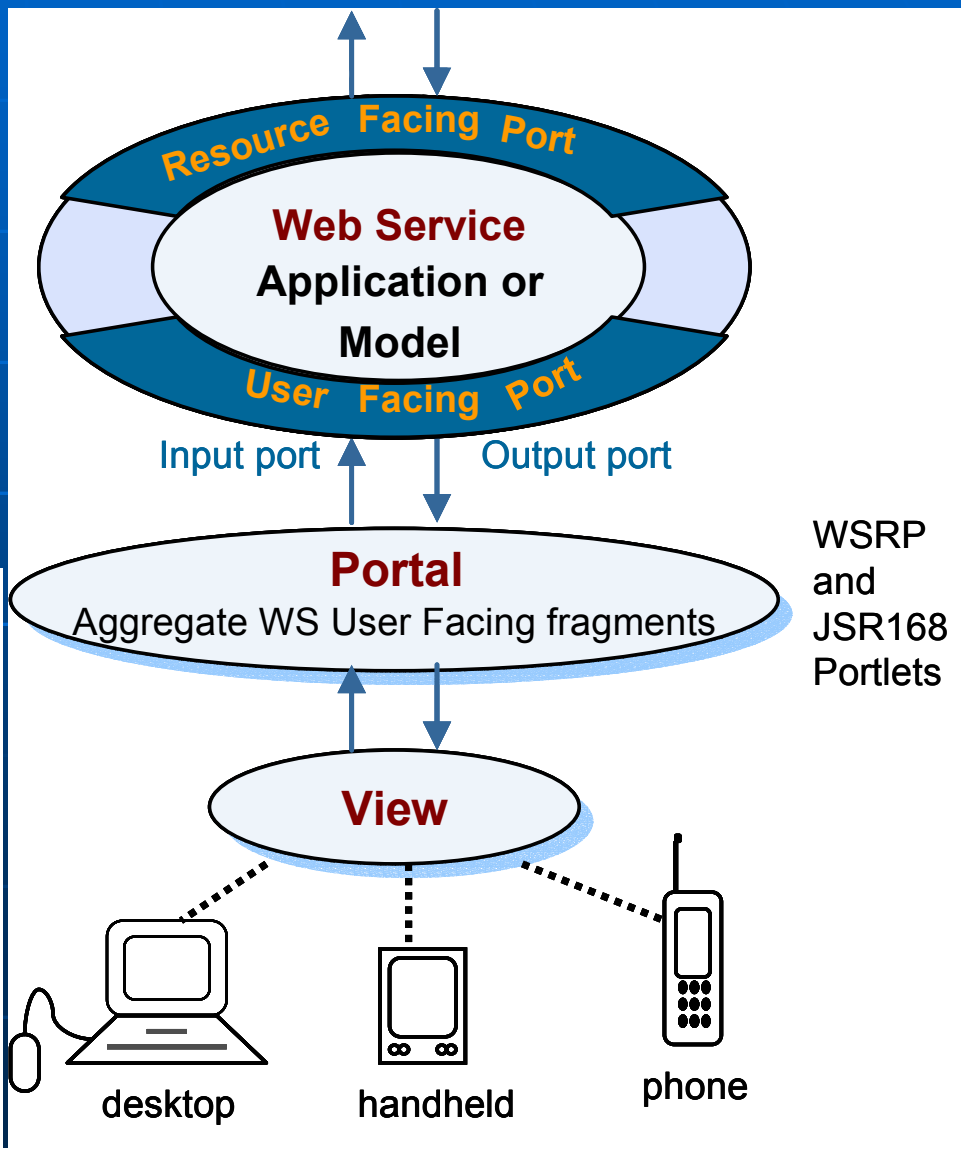
- All collaboration is about sharing events defining state changes
  - Audio/Video conferencing shares events specifying in compressed form audio or video
  - Shared display shares events corresponding to change in pixels of a frame buffer
  - Instant Messengers share updates to text message streams
  - Microsoft events for shared PowerPoint (file replicated between clients) as in Access Grid
- Finite State Change NOT Finite State Machine architecture
- Using Web services allows one to expose update events of all kinds as message streams
- Need publish/subscribe approach to share messages (NB) plus
- System to control "session" – who is collaborating and rules
  - XGSP is XML protocol for controlling collaboration building on H323 and SIP

Web Services and M-MVC

- Web Services are naturally M-MVC – Message based Model View Controller with
  - Model is Web Service
  - Controller is Portal and Messages (NaradaBrokering)
  - View is rendering

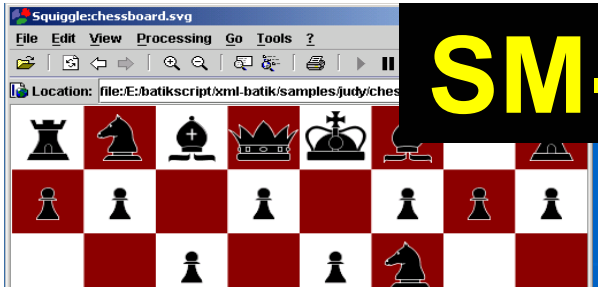


Explicit message-based Publish/Subscribe MVC model

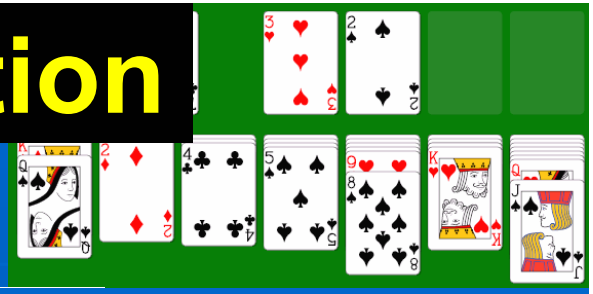


Desktop and Web Services with MMVC

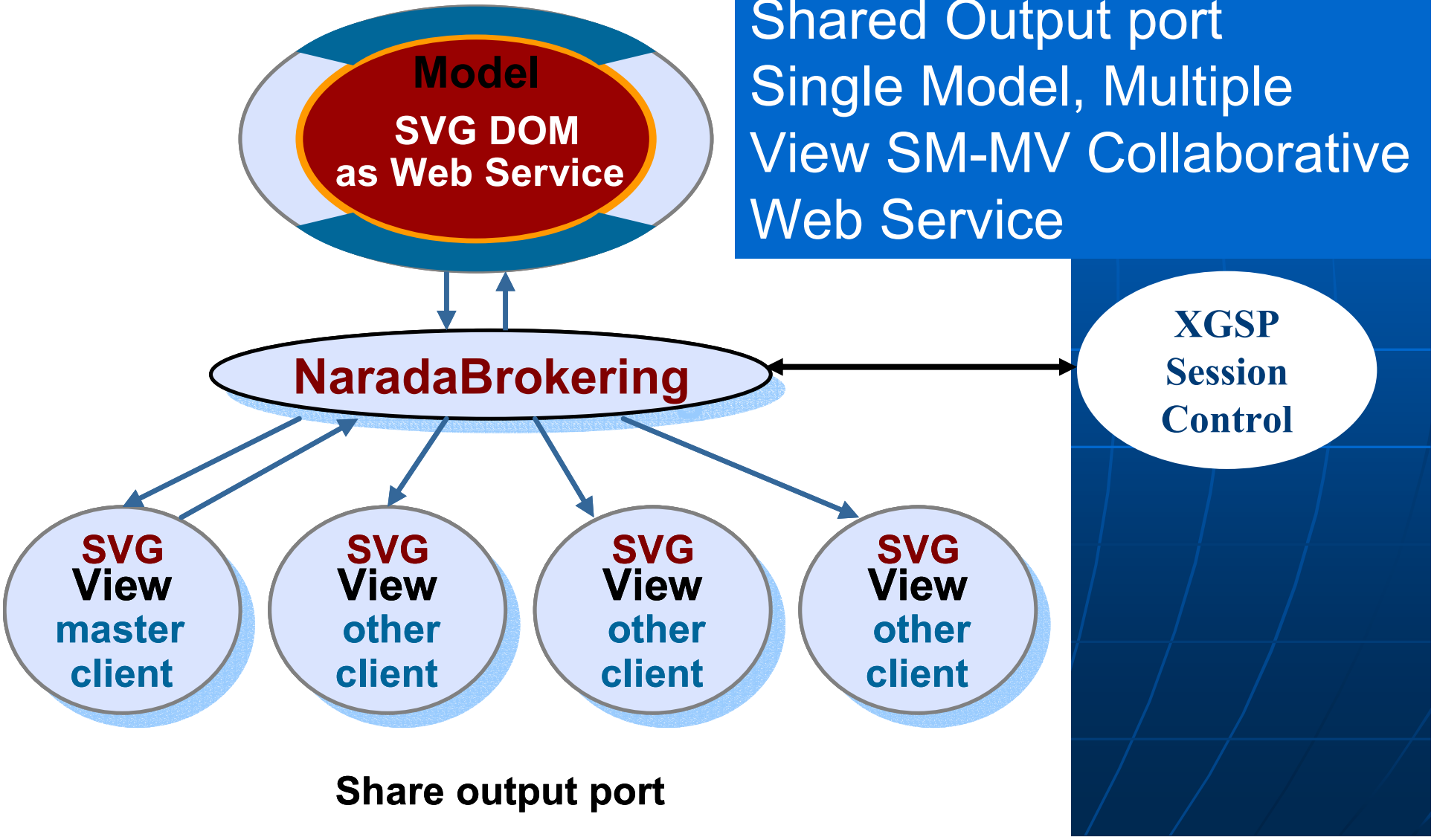
- Most desktop applications are in fact roughly MVC with controller formed by "system interrupts" with View and Model communicating by "post an event" and define a "listener" programming mode
- We propose to integrate desktop and Web Service approach by systematic use of MMVC and NaradaBrokering
- Allows easier porting to diverse clients and automatic collaboration
- Attractive for next generation of Linux desktop clients
- We have demonstrated for SVG Browser (Scalable Vector Graphics), OpenOffice and PowerPoint
- "Glob" programming style makes hard



SM-MV Collaboration

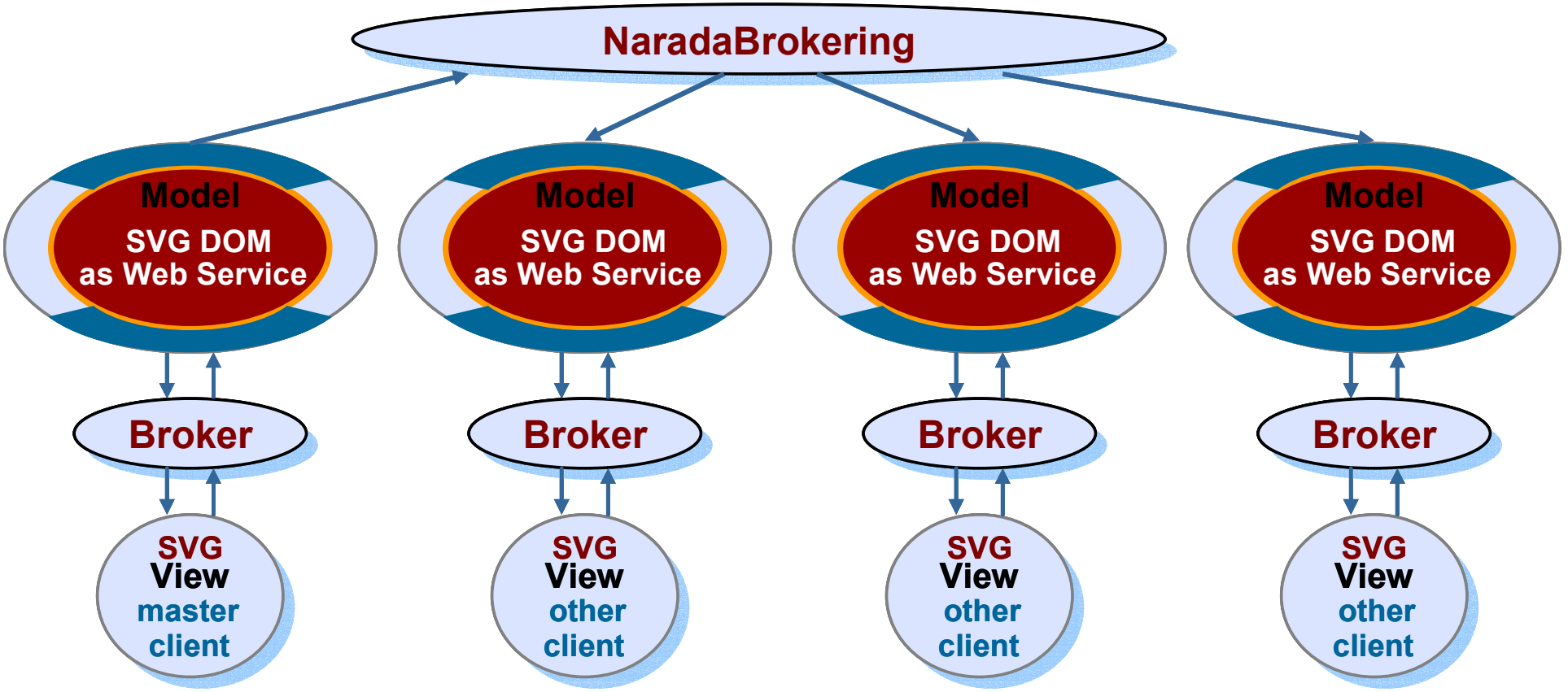


Shared Output port
Single Model, Multiple View SM-MV Collaborative Web Service



MM-MV Collaboration

Shared Input port
Multiple Model, Multiple View MM-MV
Collaborative Web Service



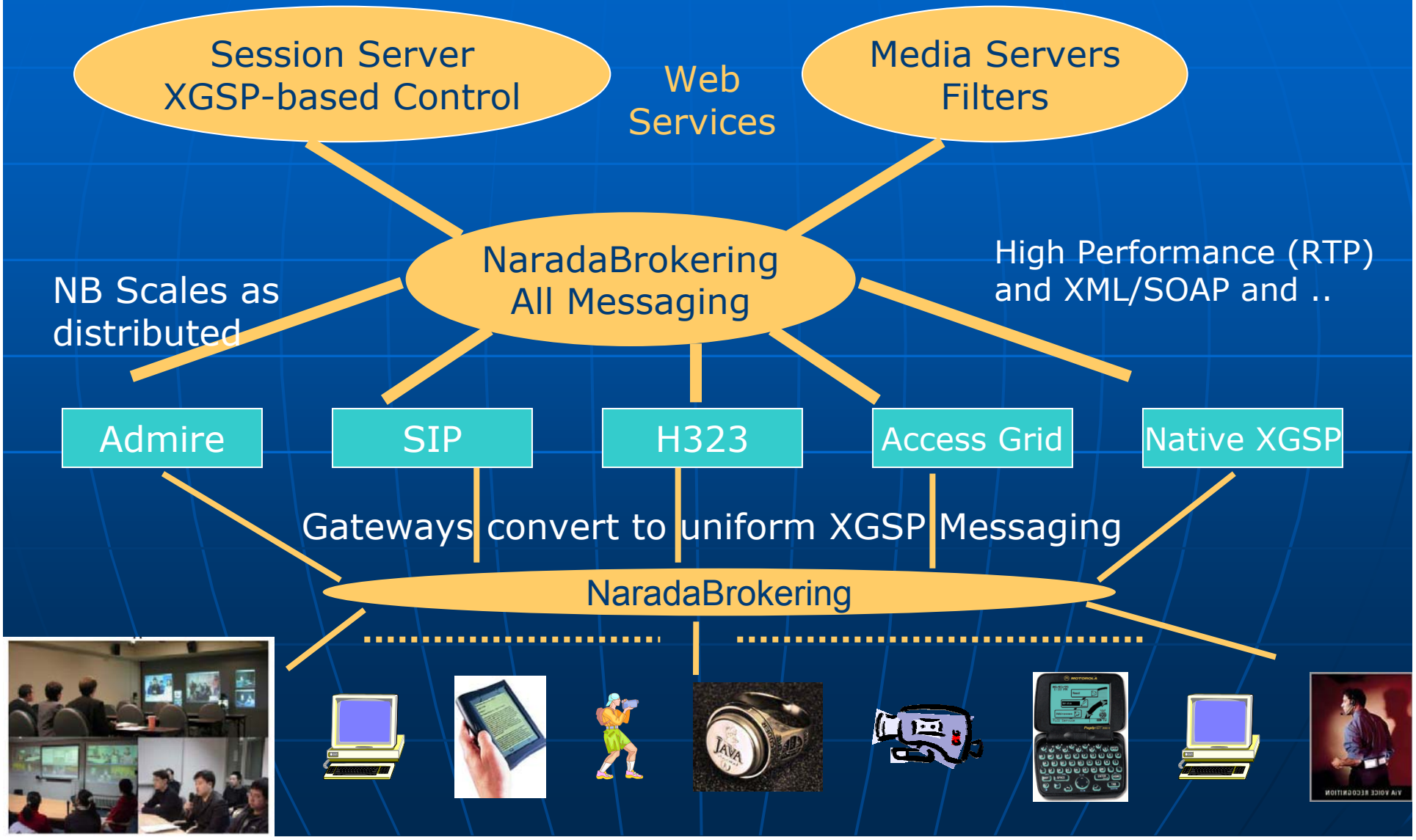
Share input port

Global-MMCS Community Grid

- We are building an open source protocol independent Web Service "MCU" which will scale to an arbitrary number of users and provide integrated thousands of simultaneous users collaboration services.
- The function of A/V media server is distributed using NaradaBrokering architecture.
  - Media Servers mix and convert A/V streams
- Open XGSP MCU based on the following open source projects
  - openh323 is basis of H323 Gateway
  - NIST SIP stack is basis of SIP Gateway
  - NaradaBrokering is open source messaging
  - Java Media Framework basis of Media Servers
  - Helix Community http://www.helixcommunity.org for Real Media
- http://www.globalmmcs.org open source "non advertised" release

XGSP Web Service MCU Architecture

Use Multiple Media servers to scale to many codecs and many versions of audio/video mixing

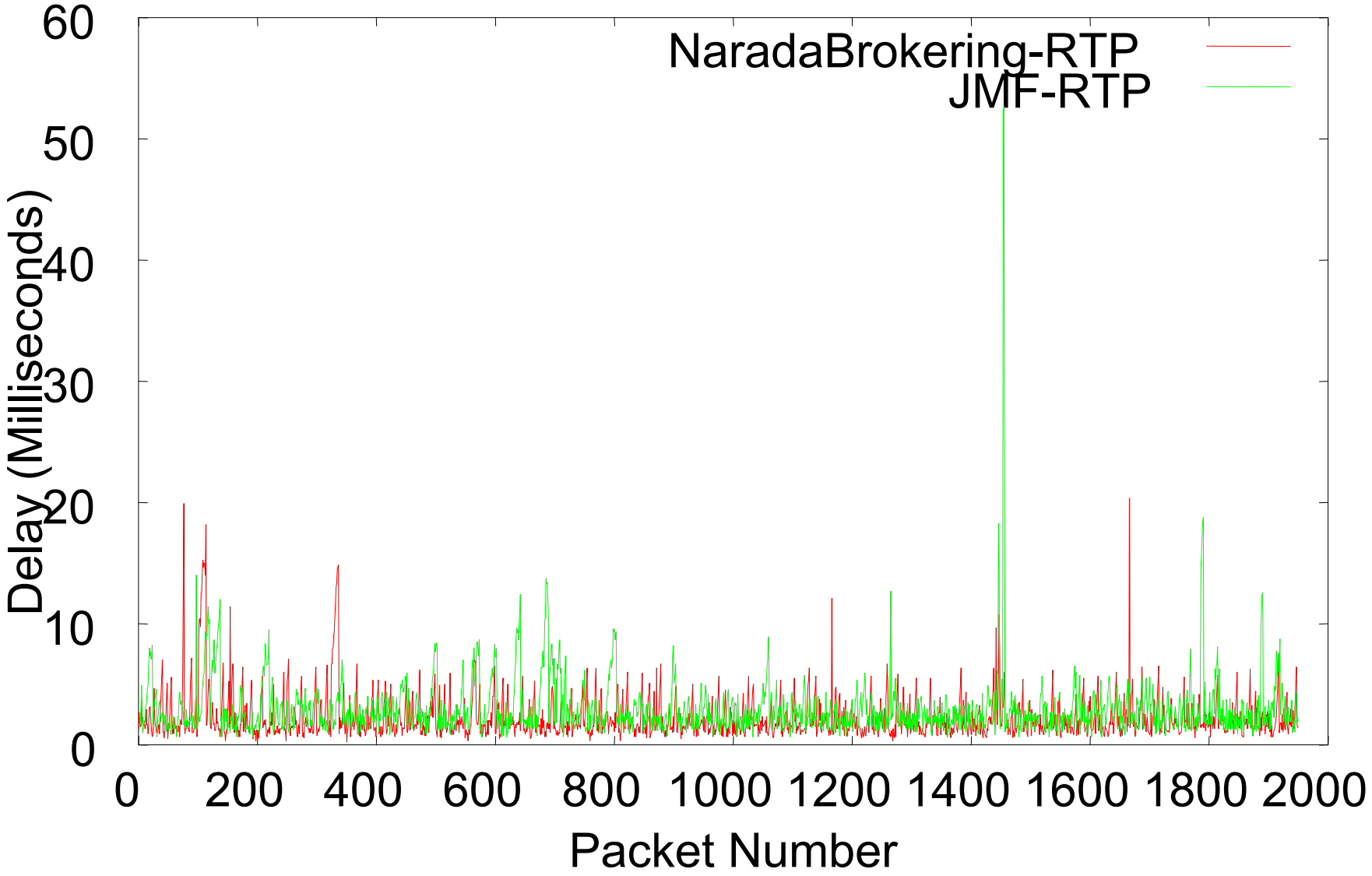


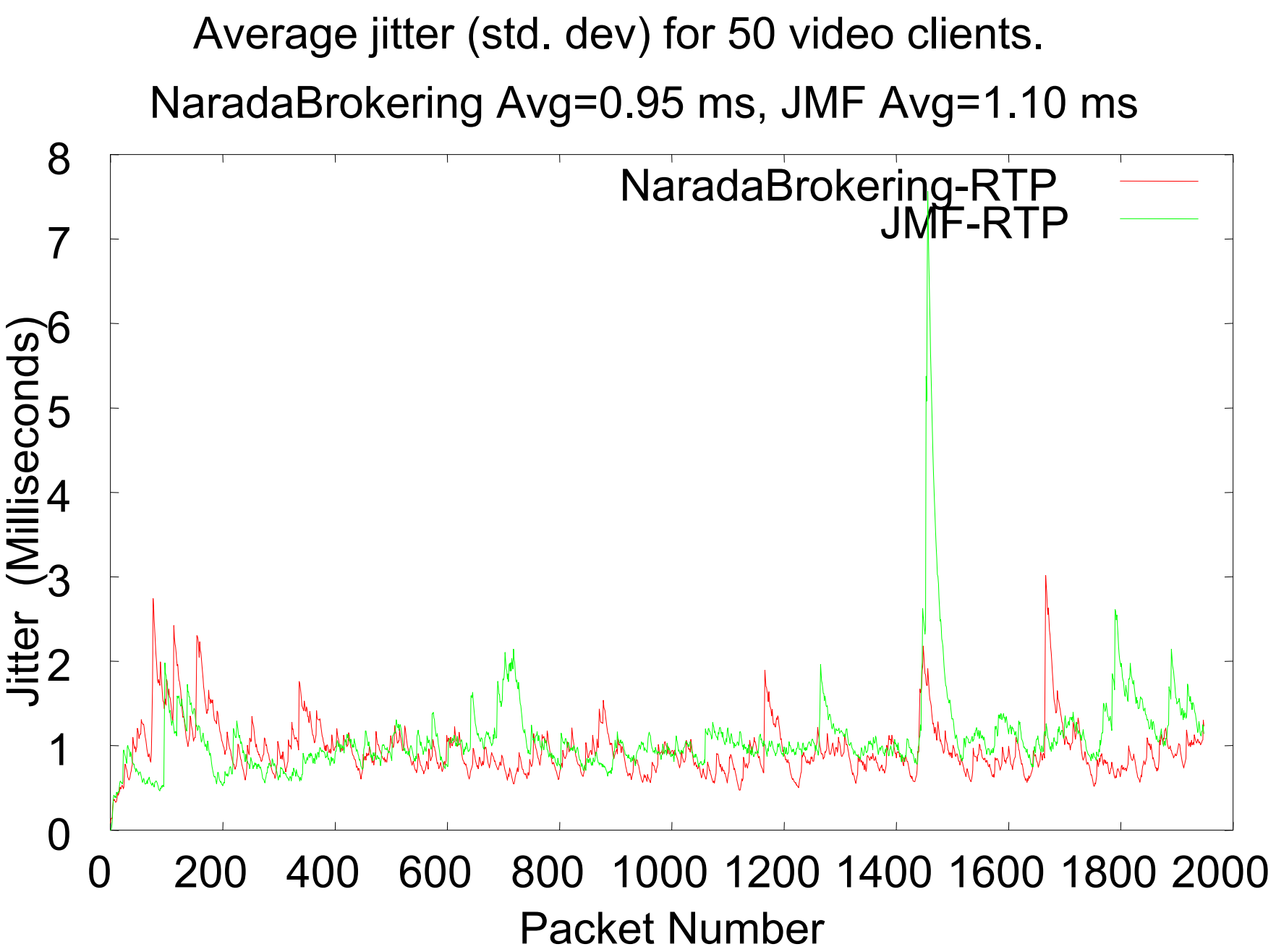
Break up into Web Services

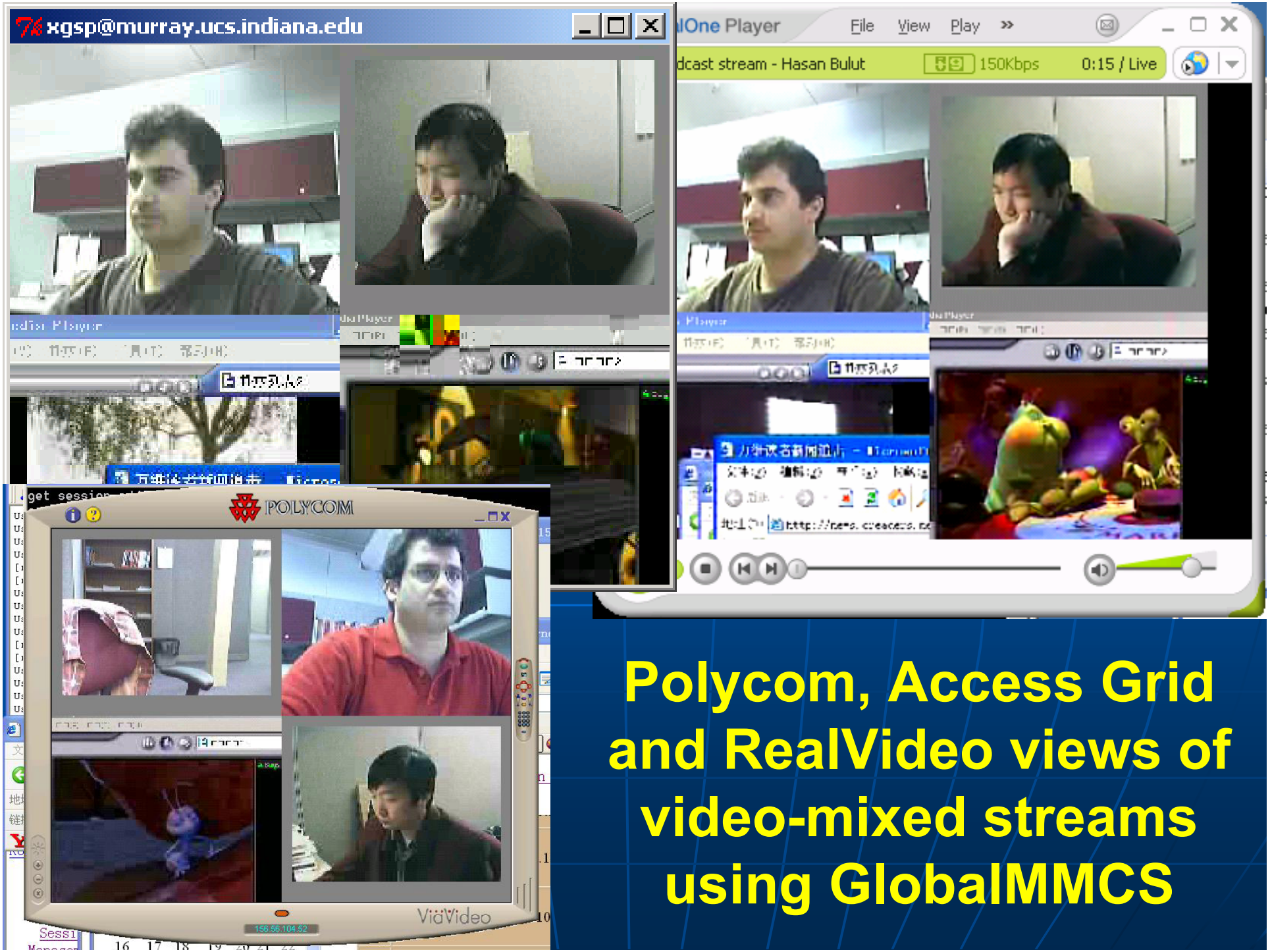
- Monolithic MCU becomes many different "Simple Services"
  - Session Control
  - Thumbnail "image" grabber
  - Audio Mixer
  - Video Mixer
  - Codec Conversion
  - Helix Real Streaming
  - PDA Conversion
  - H323/SIP Gateways
- As independent can replicate particular services as needed
  - Codec conversion might require 20 services for 20 streams spread over 5 machines
- 1000 simultaneous users could require:
  - 1 session controller, 1 audio mixer, 10 video mixers, 20 codec converters, 2 PDA converters and 20 NaradaBrokers
- Support with a stream optimized Grid Farm in the sky
  - Future billion way "Video over IP" serving 3G Phones and home media centers/TV's could require a lot of computing

GlobalMMCS and NaradaBrokering

- All communication – both control and "binary" codecs are handled by NaradaBrokering
- Control uses SOAP and codecs use RTP transport
- Each stream is regarded as a "topic" for NB
- Each RTP packet from this stream is regarded as an "event" for this topic
- Can use replay and persistency support in NB to support archiving and late clients
- Can build customized stream management to administer replay, and who gets what stream in what codec
- NaradaBrokering supports unicast and multicast
- Use firewall penetration and network monitoring services in NB to improve Q0S


Average delays per packet for 50 video-clients
NaradaBrokering Avg=2.23 ms, JMF Avg=3.08 ms
NaradaBrokering-RTP
JMF-RTP
Delay (Milliseconds)
Packet Number
0
10
20
30
40
50
60
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Average jitter (std. dev) for 50 video clients.
NaradaBrokering Avg=0.95 ms, JMF Avg=1.10 ms
NaradaBrokering-RTP
JMF-RTP
Jitter (Milliseconds)
Packet Number
0
1
2
3
4
5
6
7
8
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Polycom, Access Grid and RealVideo views of video-mixed streams using GlobalMMCS

Integration of PDA, Cell phone and Desktop Grid Access

NB Support for optimized PDA Communication

GlobalMMCS Futures

- Current "release" has very rudimentary session management
  - Should support adding members and applications to a collaborative session
  - Administration and master/non-master roles
- Collaborative PowerPoint, OpenOffice, SVG release waiting XGSP session manager
- Most interesting clients are Java applets supporting portlet model for modern portals
- Linux and Macintosh clients require higher performance JMF – Java Media Framework
- Need to support use of NB QoS features
- Add additional codecs like MPEG2 and MPEG4
- Improve video codec-based shared display
- Need scheduler of dynamic services sensitive to streaming bandwidth requirement as well as CPU use of codec conversion