

Open Architecture Publish-Subscribe Benchmarking

Bruce McCormick, EG&G
Leslie Madden, NSWCD



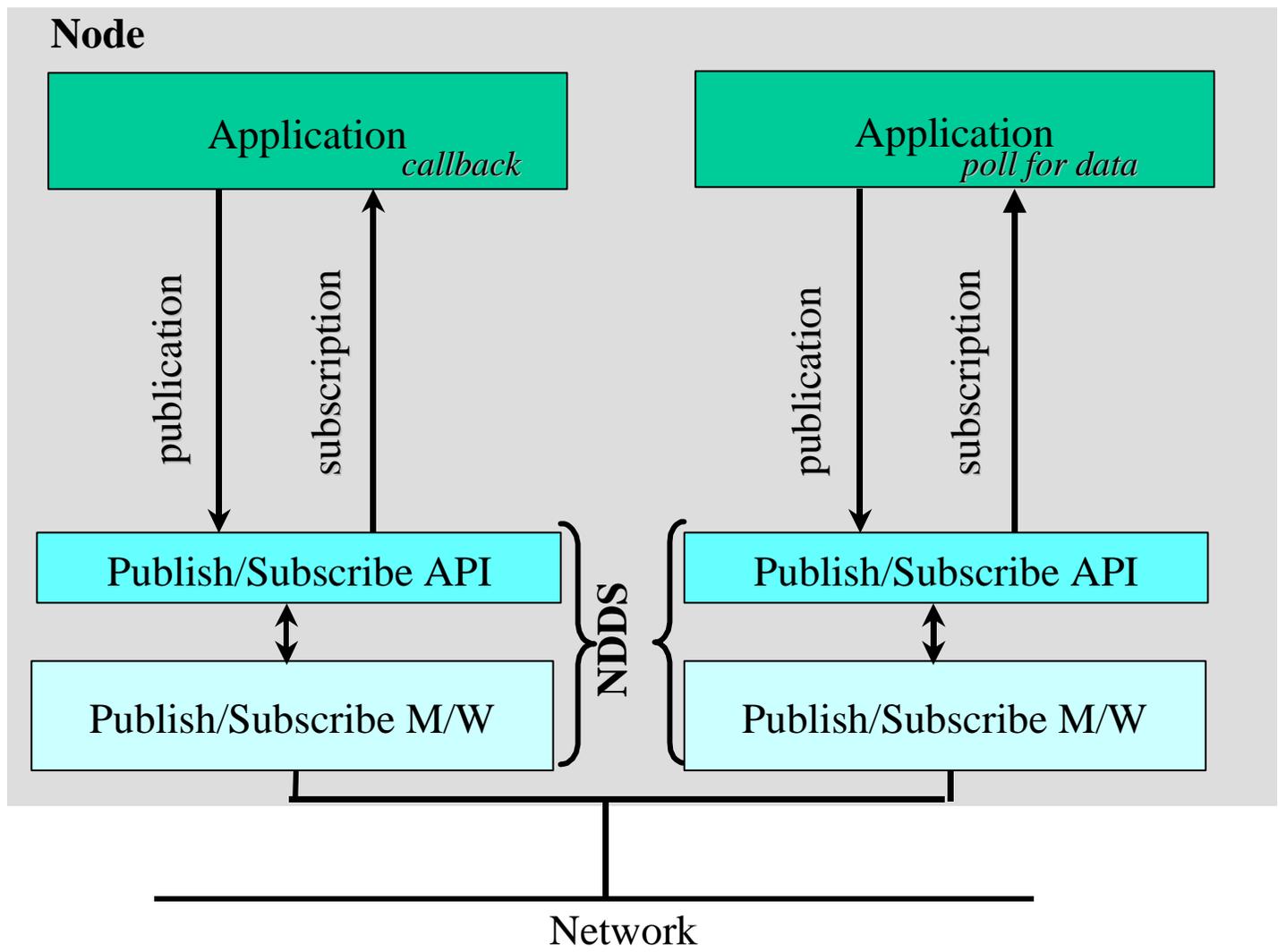


Objectives

- ◆ To gain familiarity with existing publish-subscribe middleware
- ◆ Understand ability of products to support bounded latencies and sustained throughput required by combat system applications
- ◆ Understand architectural differences in products that may affect ability of products to scale to large numbers of processors & processes
- ◆ Provide a baseline of comparison as products migrate toward DDS compliance



RTI NDDS 3.0M



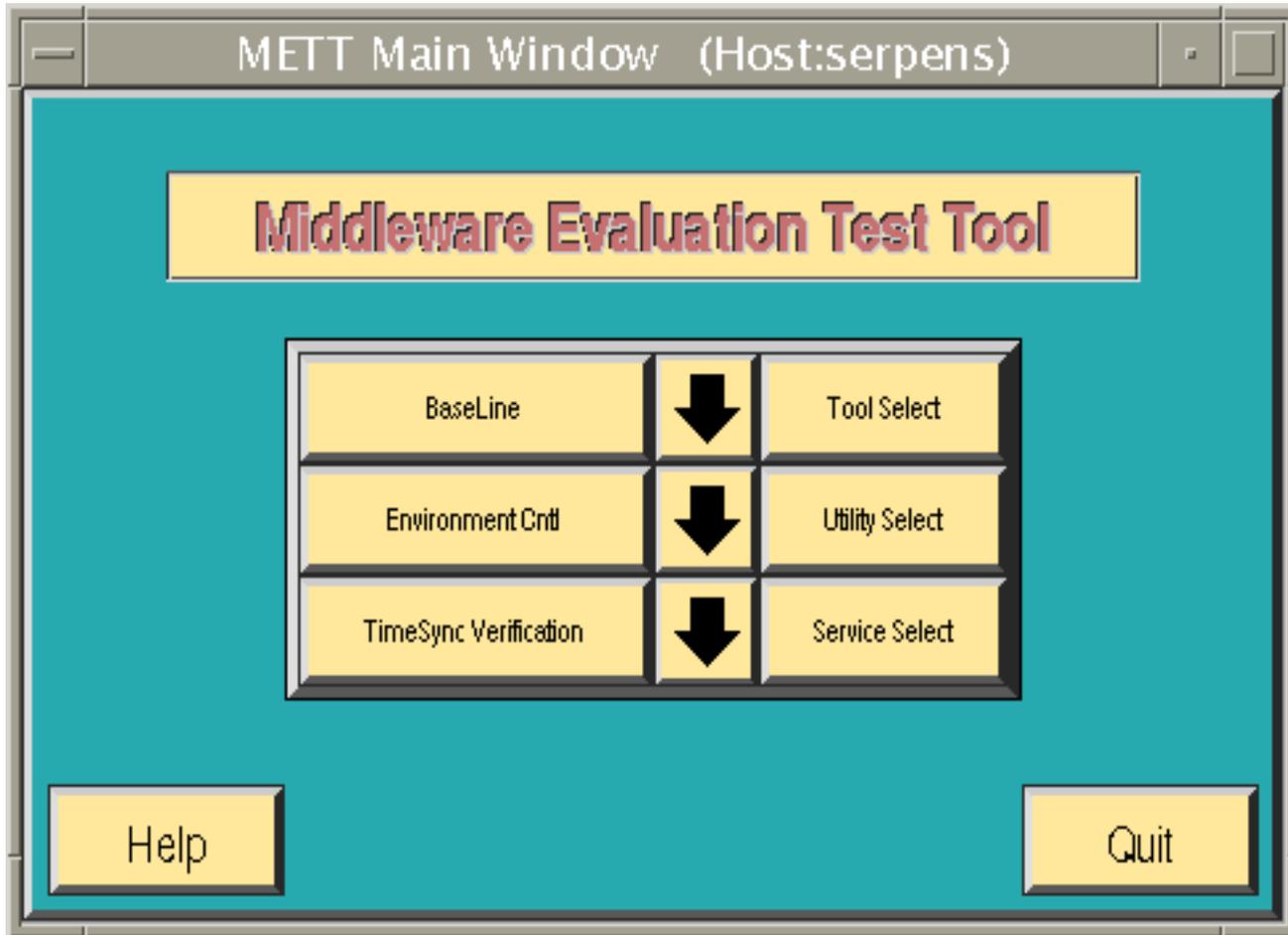


Architectural Features Affecting Results

SPLICE	NDDS
Stores data into in-memory data store for future access	Receives data into message queues
Implements reliable but not FIFO – most recent message issue (based on timestamp) overwrites previous issue	Implements Reliable/FIFO – all issues delivered. Improperly sized queues can lead to flow control behavior

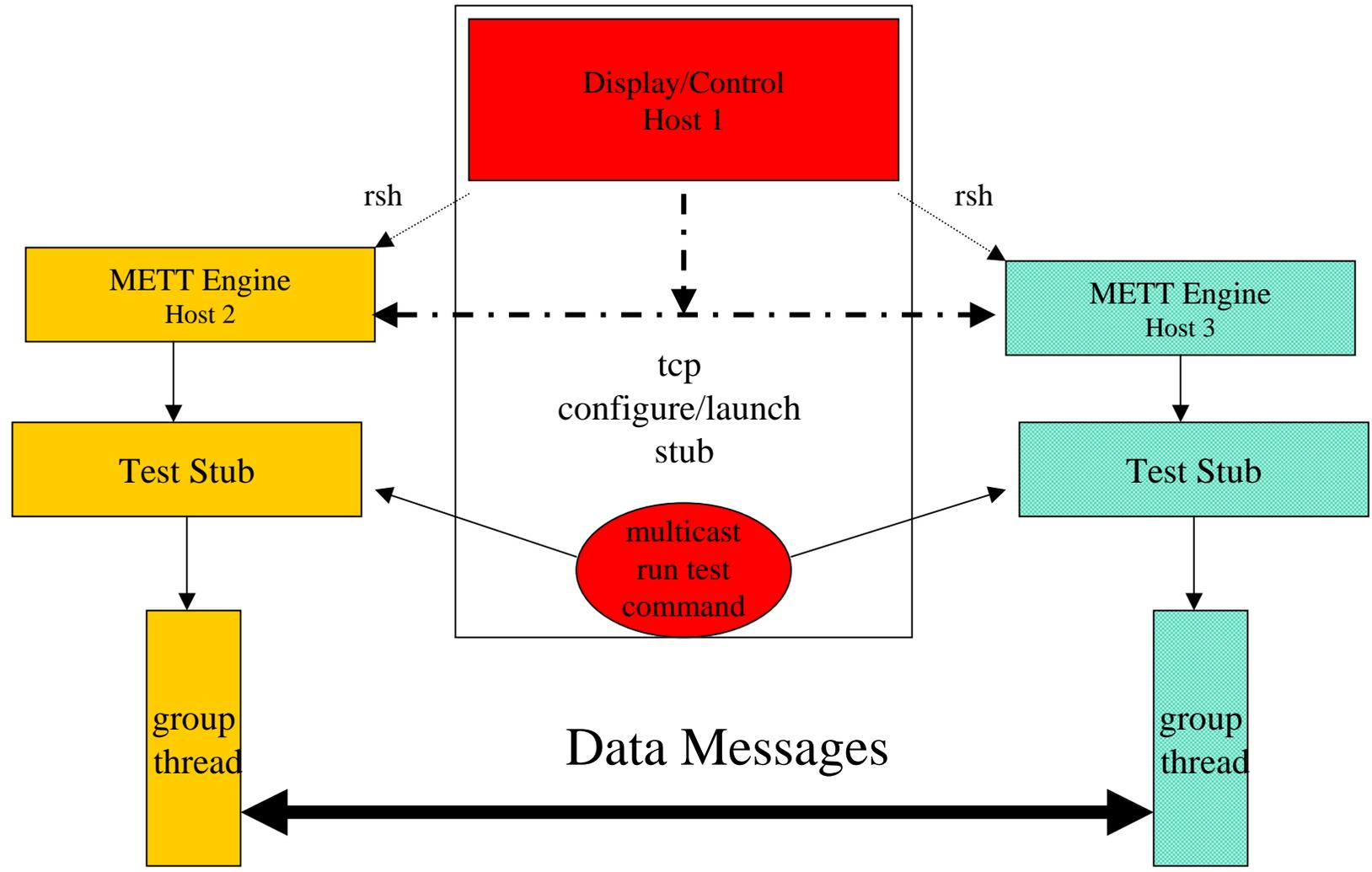


Mett Tool





Mett Design





Mett Stubs

Middleware Test Controller: (Host:serper)

Test Template ID: Middleware Tool: Empty Set

Group Name(s): Test Interface: rpens.nswc.navy.mil

Host Name: adara3.nswc.navy.mil Update Interval: 2

User Data 1: User Data 2:

User Data 3: User Data 4:

Application Type: Listen Only Transport Type: DEERING

Transmission Type: FIFO Transmit Method: Cast

Msg Pattern: Fixed Init MsgSz: 200 Max MsgSz: 200

Memory Model: Static Increment: 0 Period: 1

Smooth Const: 50 Msg Rate: 0 Msg Count: Msg Cnt: 100

File Select: No File File Fmts: Text File Path: /tmp

View Display: View Output: Full Detail:

Monitor CPU: Disabled Test Duration: One Shot

Interface Control: Normal Interface Name:

Activate On: Cnt: Disable On Msg: 10

Launch Run Done

Test Stub Controller: (Host:serpens)

Application Control

Quit Op Status Help

Update All Data Update Sel Data Clear Sel Browser

```

# ===== Configuration =====
# HostName: serpens
# Application Type: Send Only
# Middleware Type: Ensemble
# Msg Reception Method: Callback
# Transmission Type: FIFO
# Transport Type: DEERING
# Transmit Method: Cast
# Middleware Specific Arg1: N/A
# Middleware Specific Arg2: N/A
# Middleware Specific Arg3: N/A
# Middleware Specific Arg4: N/A
#
# File Format Type: Text
# Files Created: No File
# File Path: /tmp
# Output Detail: Partial
#

```

Browser Select

Program Cfg Program Stats Middleware Cfg

Msg Stats Msg Detail Error Log

Program Event Enable Interface Plot: 217355

Test Stub Window

Test Configuration Window



Test Environment

- ◆ All nodes in the same subnet
- ◆ Use multicast
- ◆ Privileged User
 - Mett sets high (Max – 5) RT thread and process priorities
 - Not bound to any CPU (SMP)

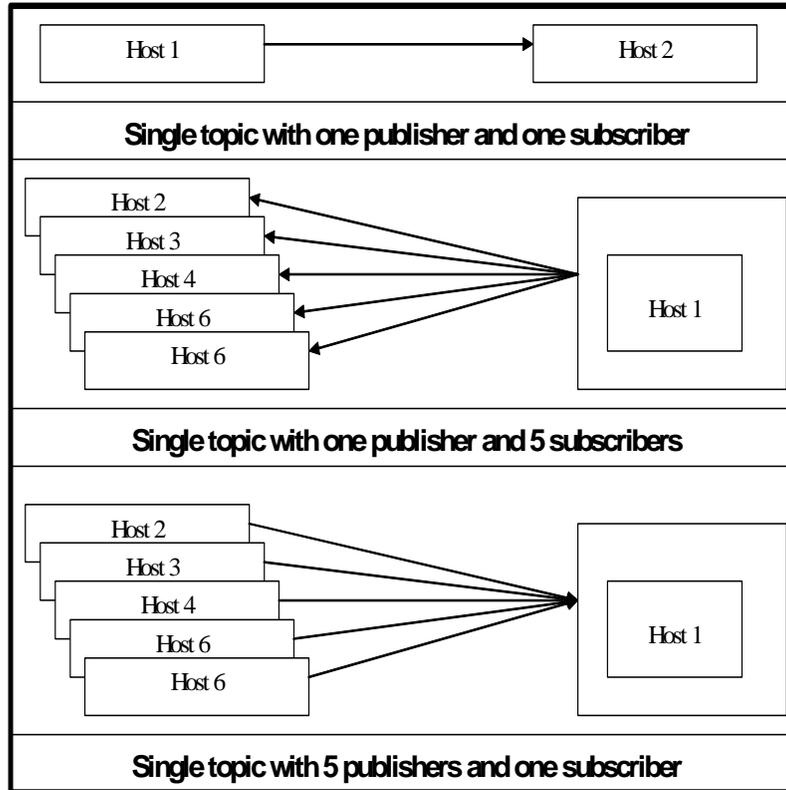


Host Configuration

Host	OS	Description	Network Interface
Alphe1	Solaris8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet
Alphe2	Solaris8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet
Alphe3	Solaris8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet
Alphe4	Solaris 8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet
Serpens	Solaris 8	SUN Ultra-Enterprise 8 processor UltraSparc – sun4u	Fast Ethernet
Alphe4	Solaris 8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet
Alphe5	Solaris 8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet
Alphe6	Solaris 8	SUN Ultra2 w/ Dual 200 MHz UltraSparc - sun4u 128 MB RAM	Fast Ethernet



Test Setups

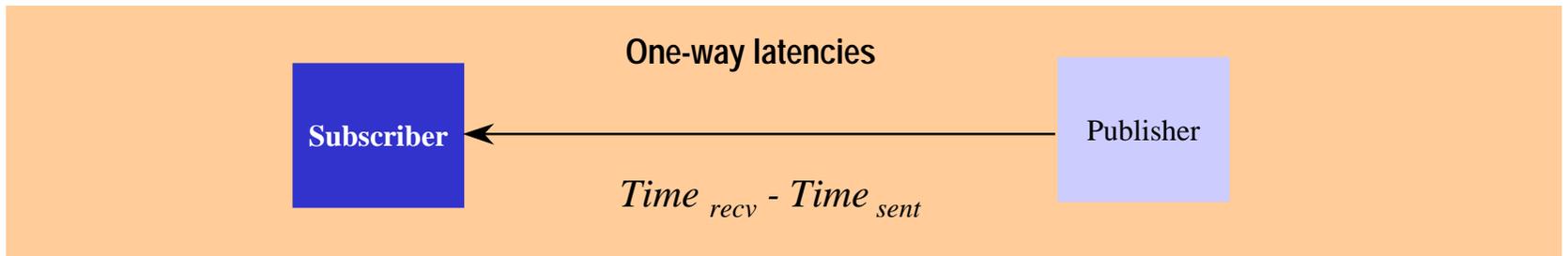
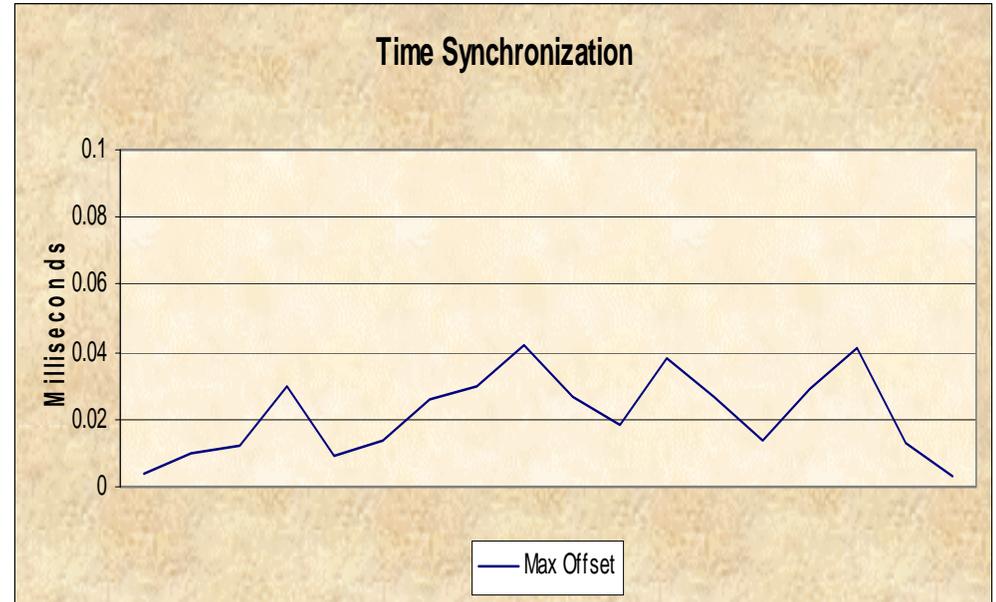


Configuration Setup



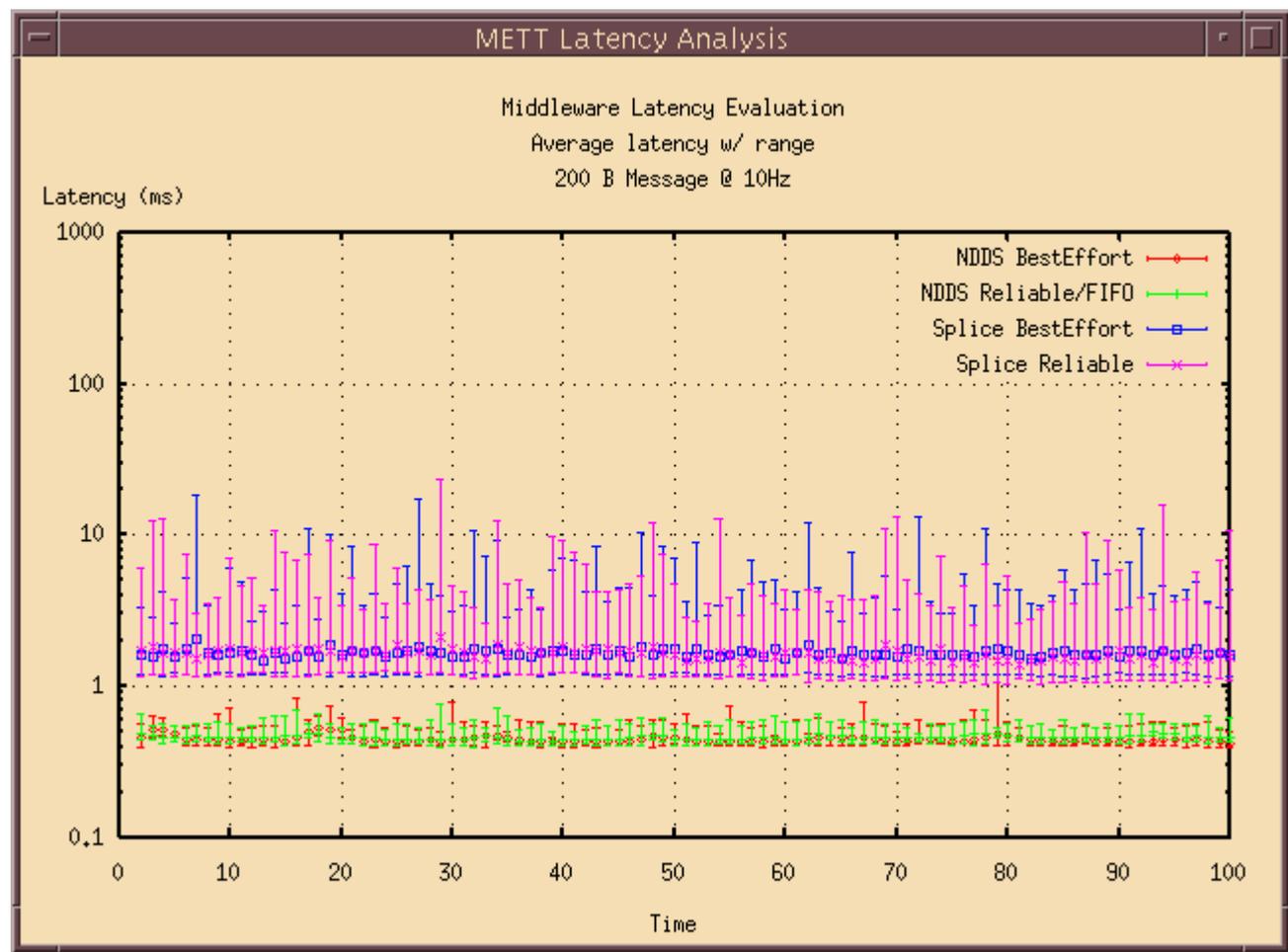
Methodology

- NTP for clock sync
- NDDS 3.0m configuration
 - Reliable queue size 1
- Splice 2
- Vendor tool code generation
 - C libraries





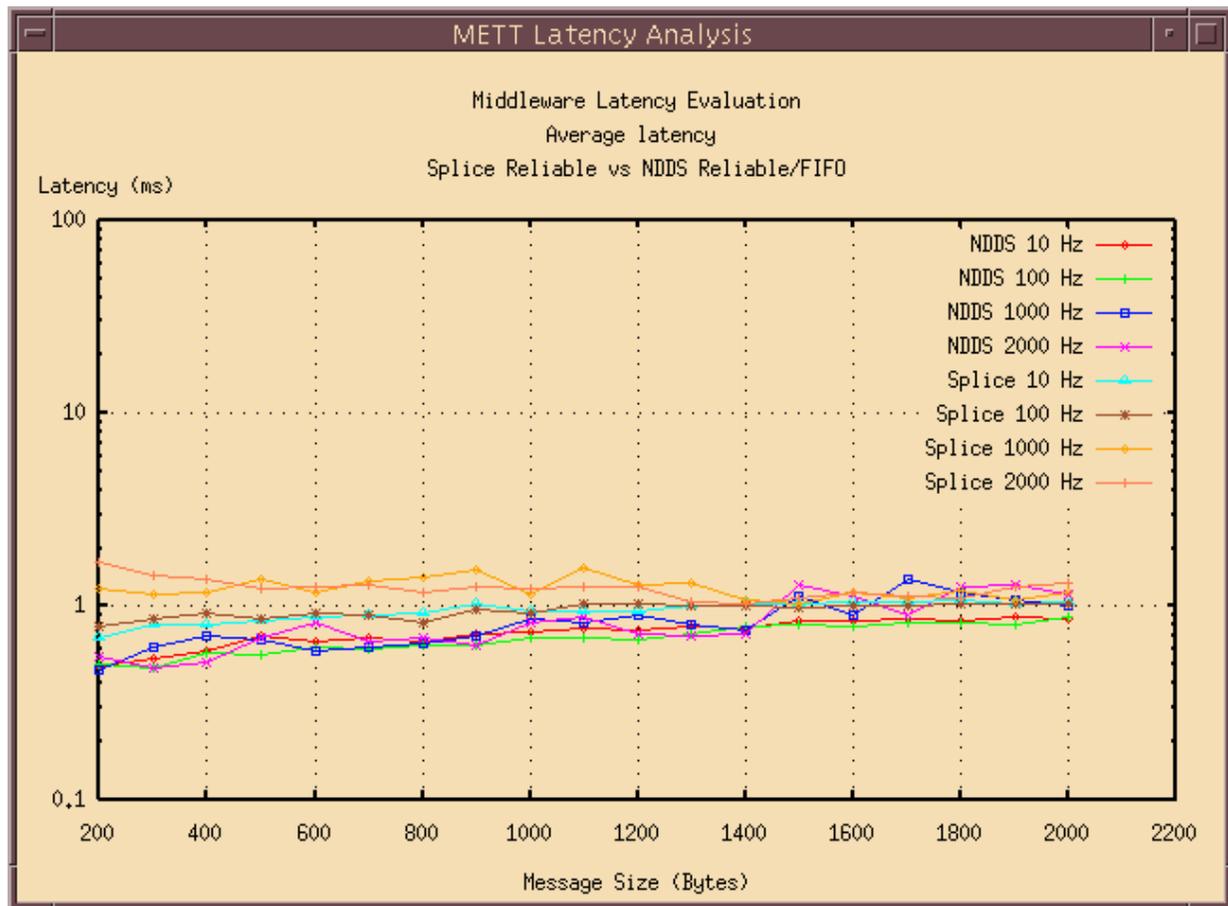
Small Message Low Rate



Latencies of 200 Byte Message Delivery at 10 Hz



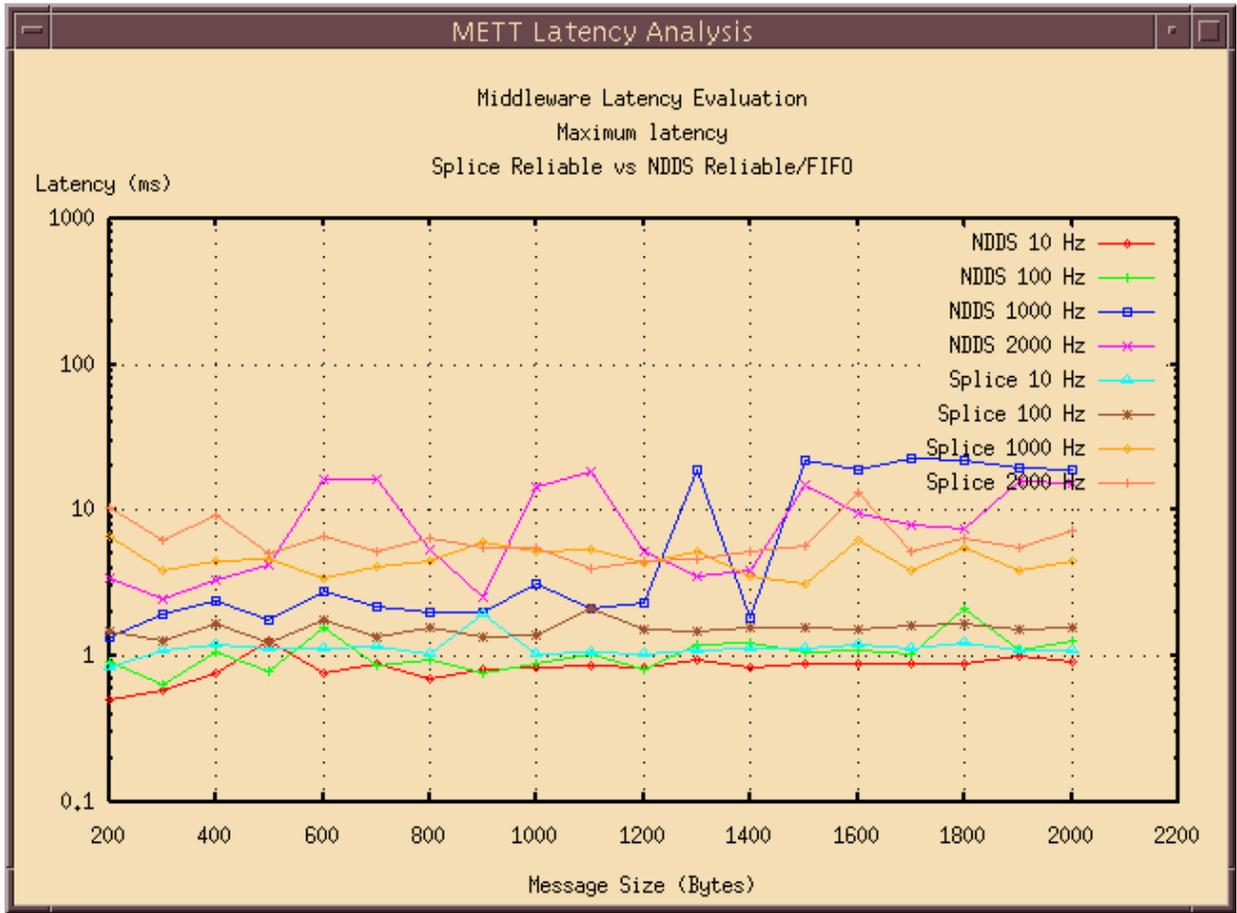
Reliable Delivery - Average Latencies



Average Latencies of Different Message Sizes at Different Rates



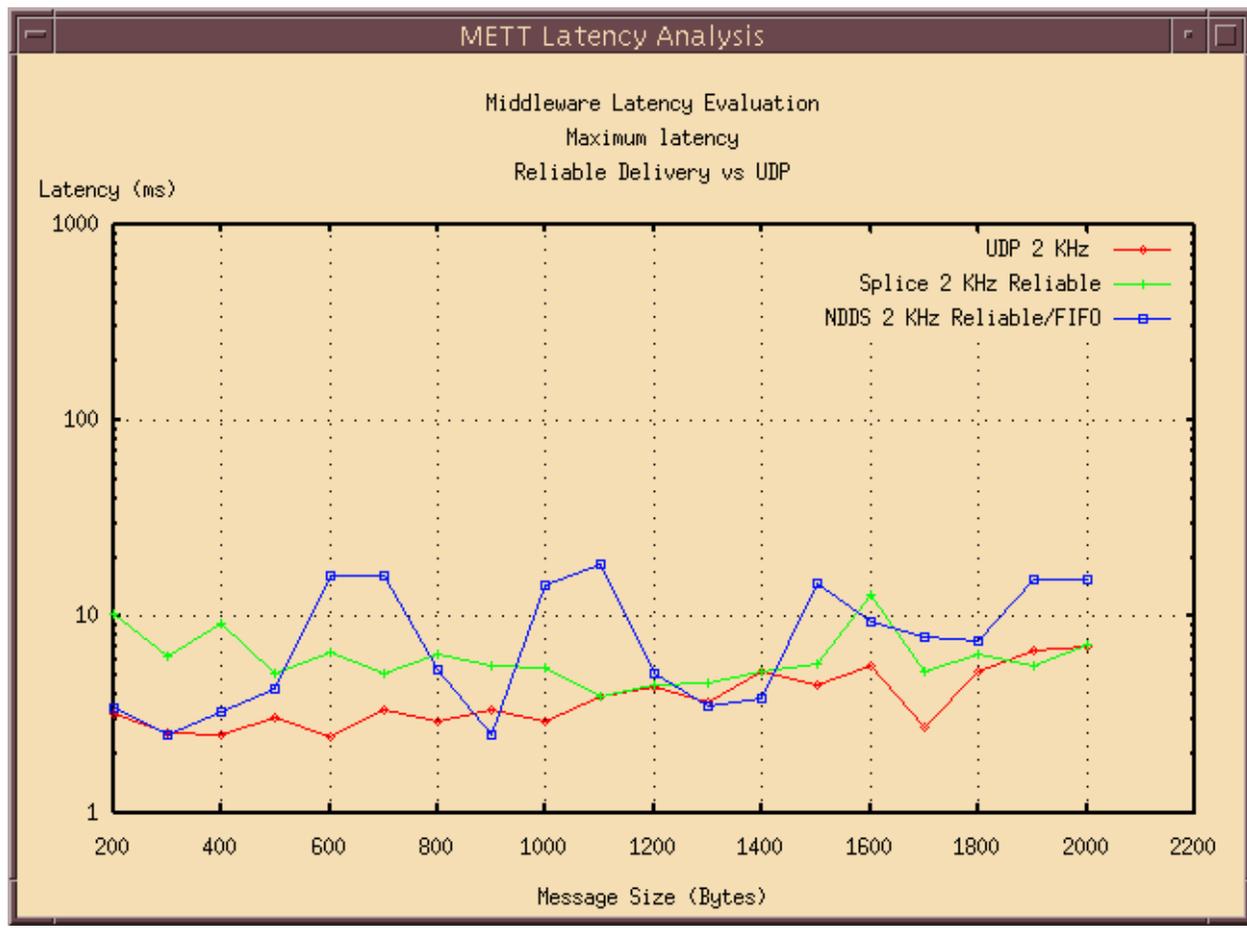
Reliable Delivery - Maximum Latencies



Maximum Latencies of Different Message Sizes at Different Rates

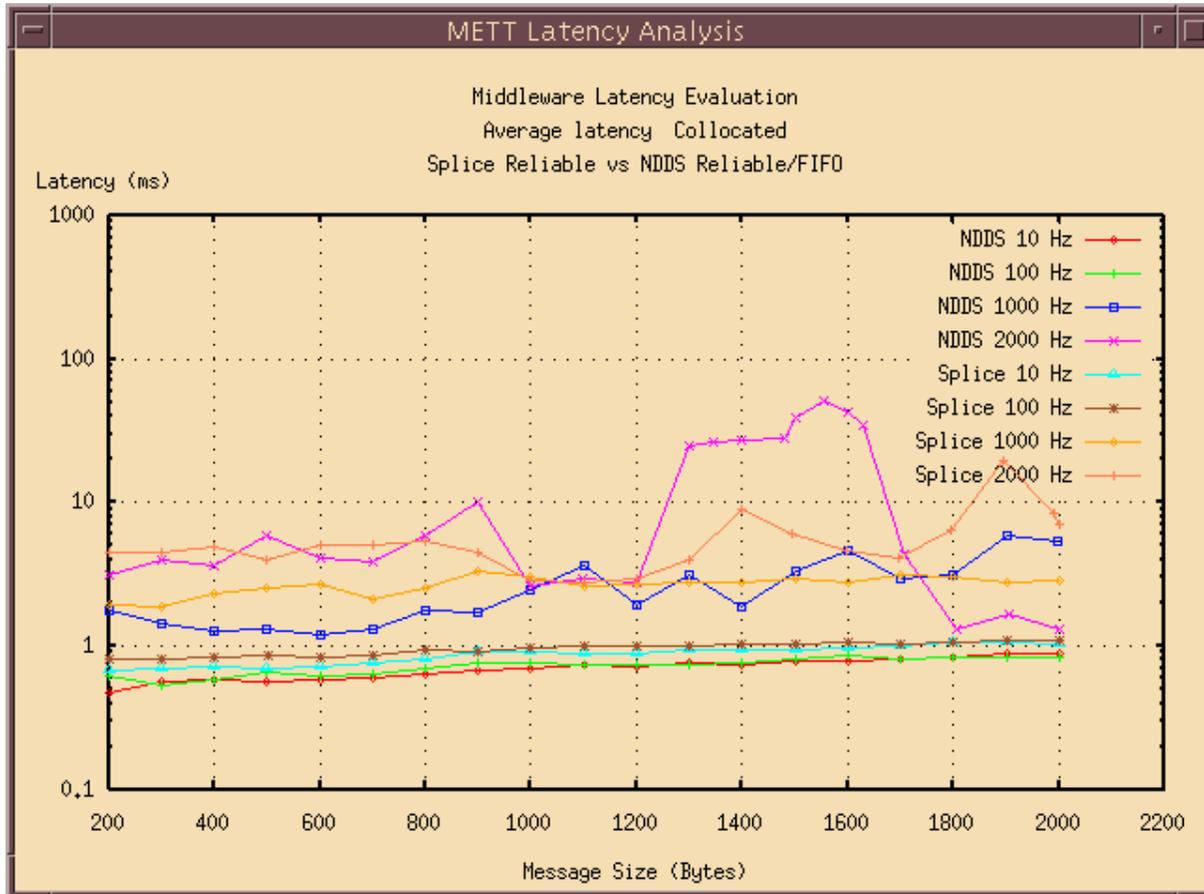


UDP, NDDS, & SPLICE - Maximum Latencies



Maximum Latencies for SPLICE Reliable, NDDS Reliable/FIFO, and UDP at 2000 Hz

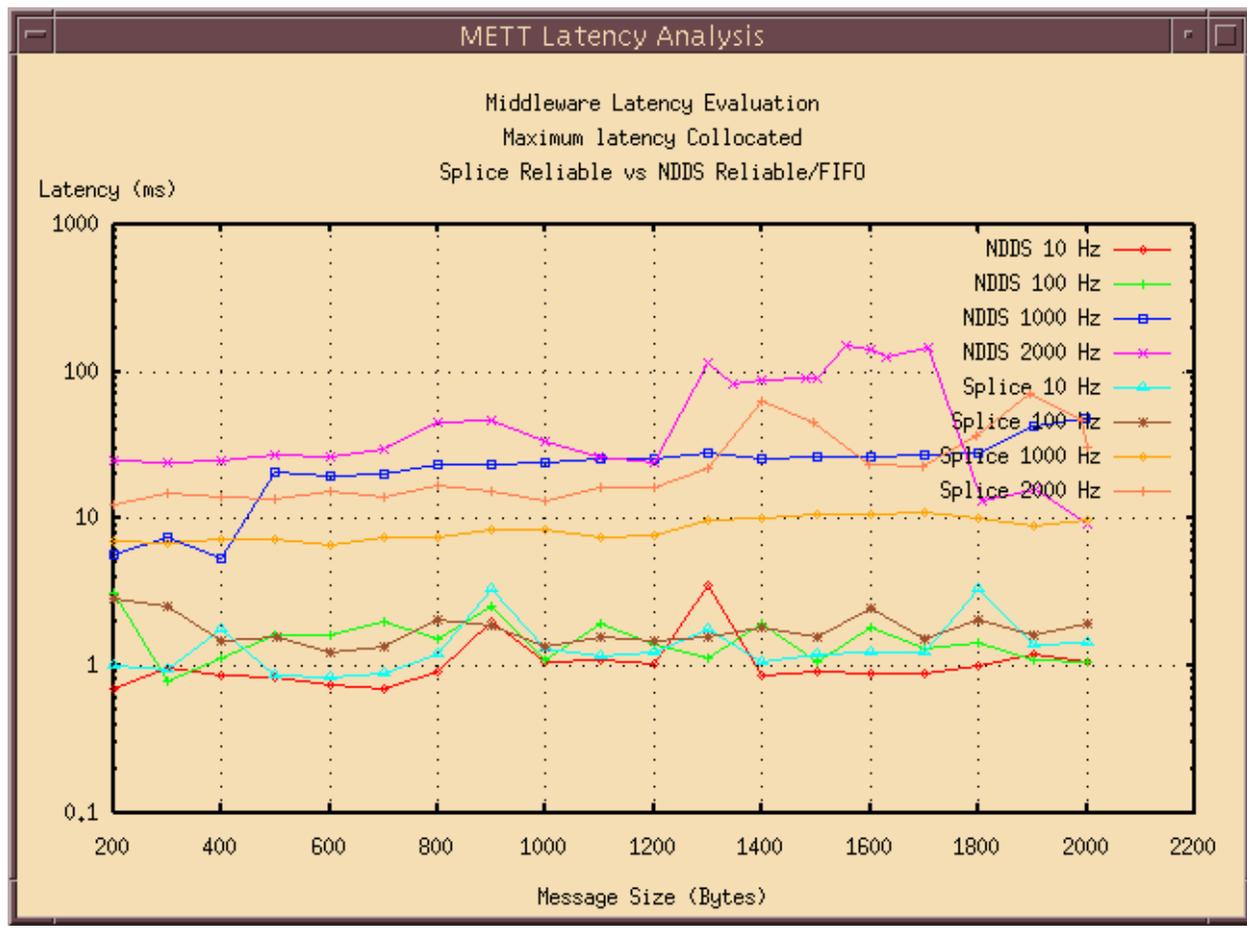
Simultaneous NDDS & SPLICE - Average Latencies



Average Latencies of Different Message Sizes at Different Rates with Collocated Hosts



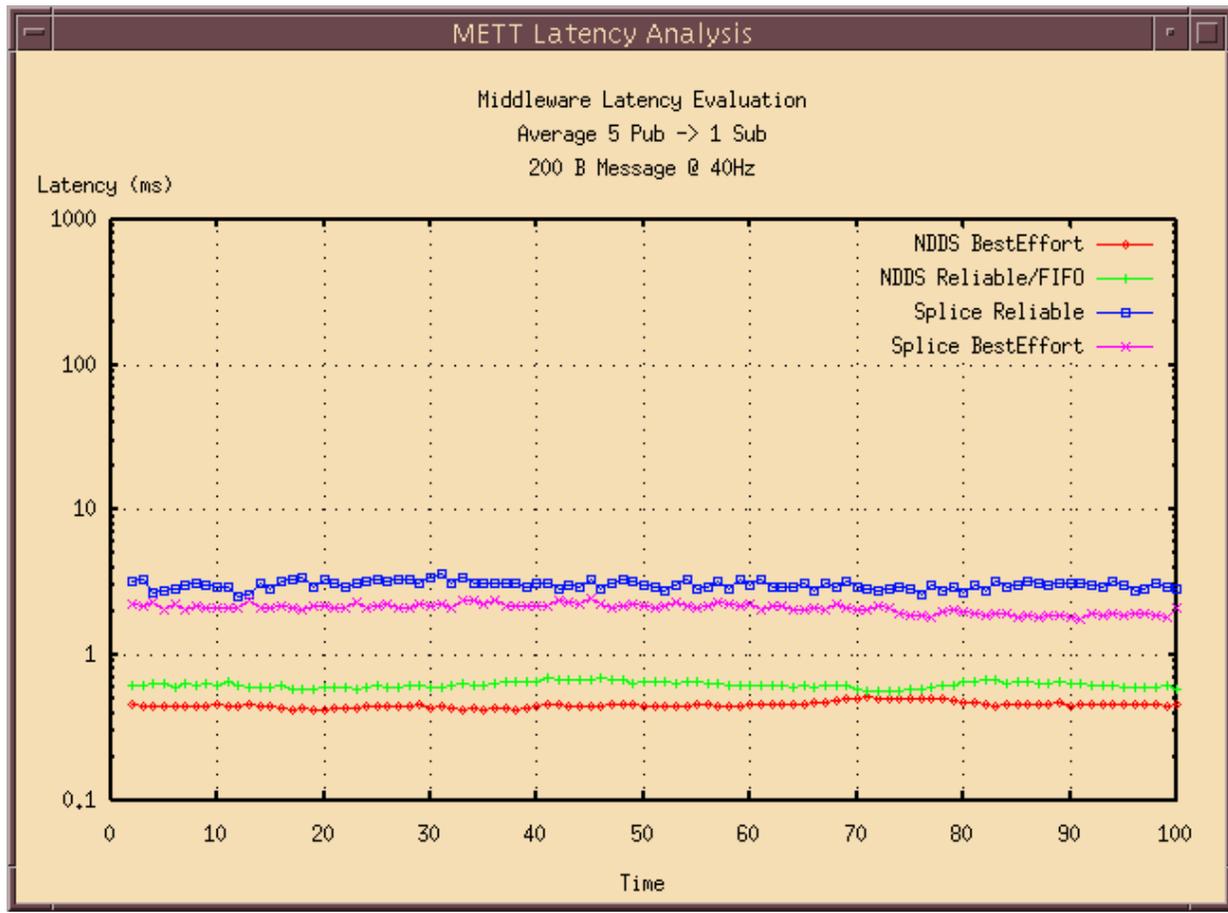
Simultaneous NDDS & SPLICE - Maximum Latencies



Maximum Latencies of Different Message Sizes at Different Rates with Collocated Hosts



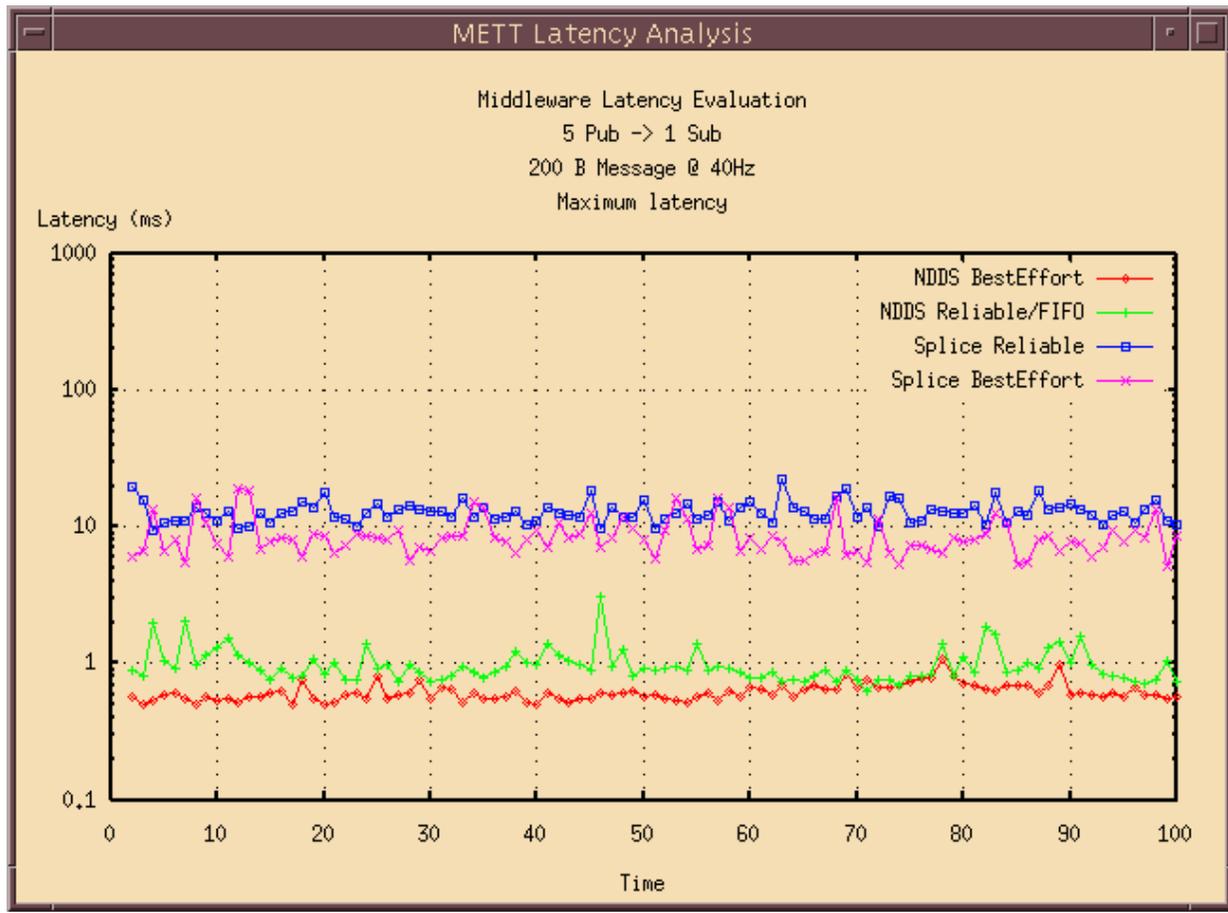
5 Publishers to 1 Subscriber - Average Latencies



Average Latencies of Five Publishers Sending Data To One Subscriber



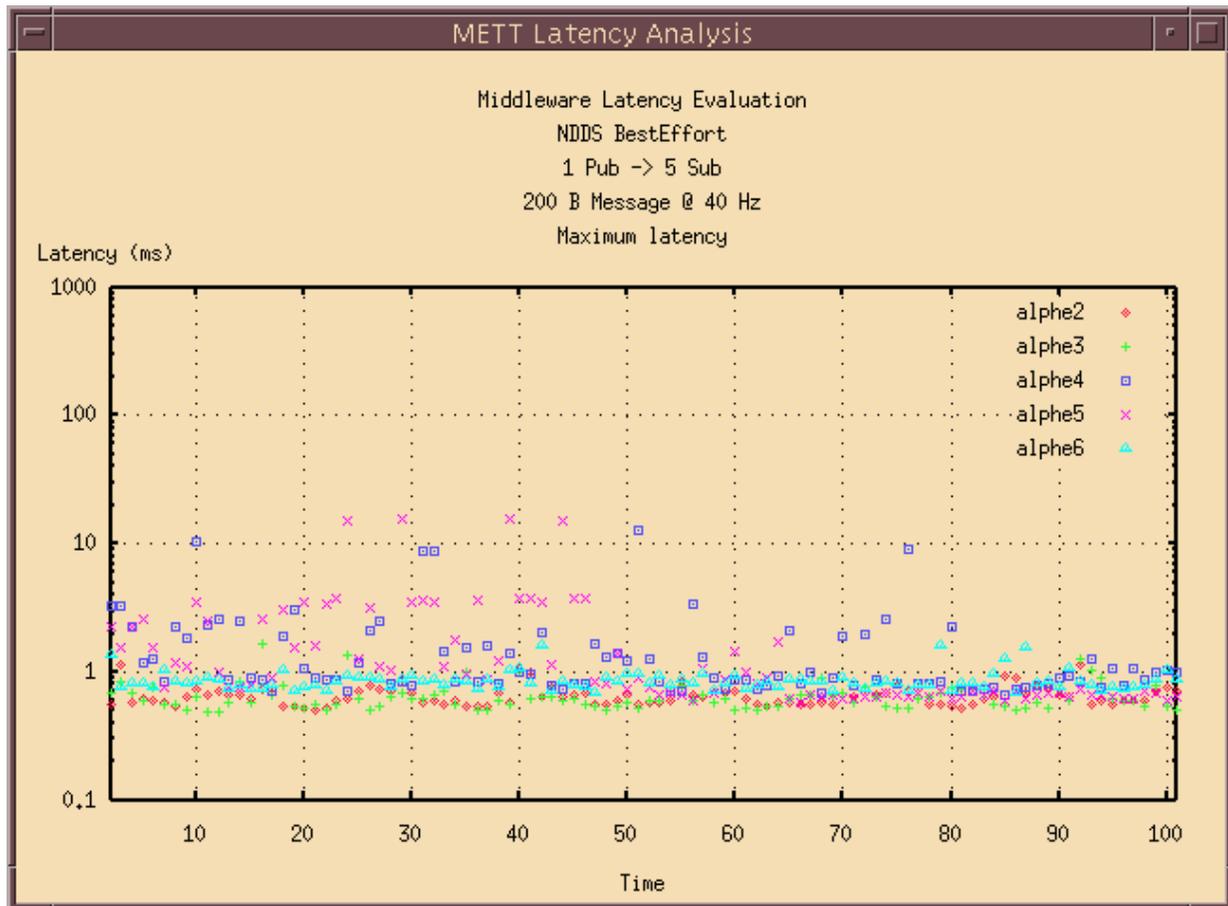
5 Publishers to 1 Subscriber – Maximum Latencies



Maximum Latencies of Five Publishers Sending Data to One Subscriber.

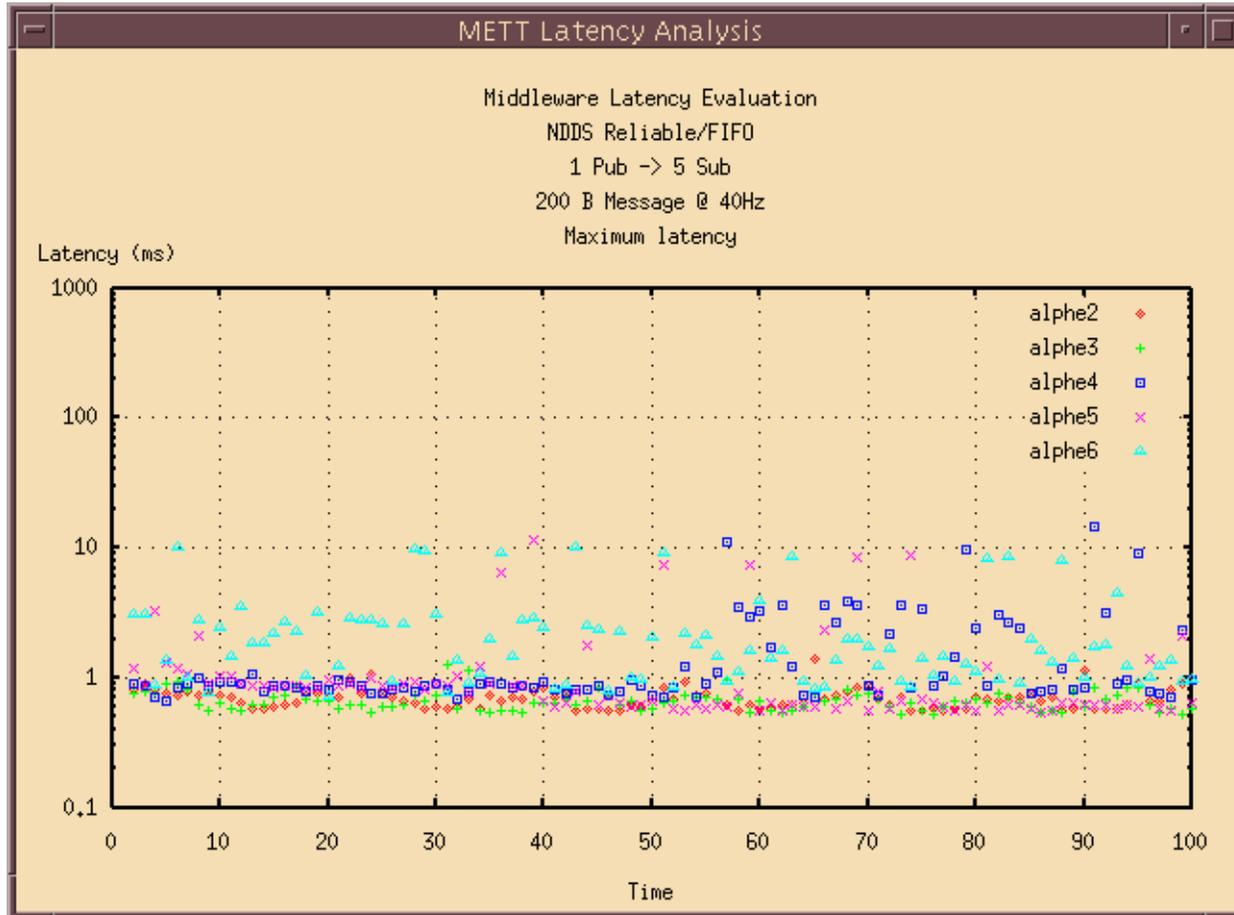


NDDS 1 Publisher to 5 Subscribers Best Effort – Maximum Latencies



Maximum Latencies of One Publisher Sending Data to Five Subscribers – NDDS Best Effort

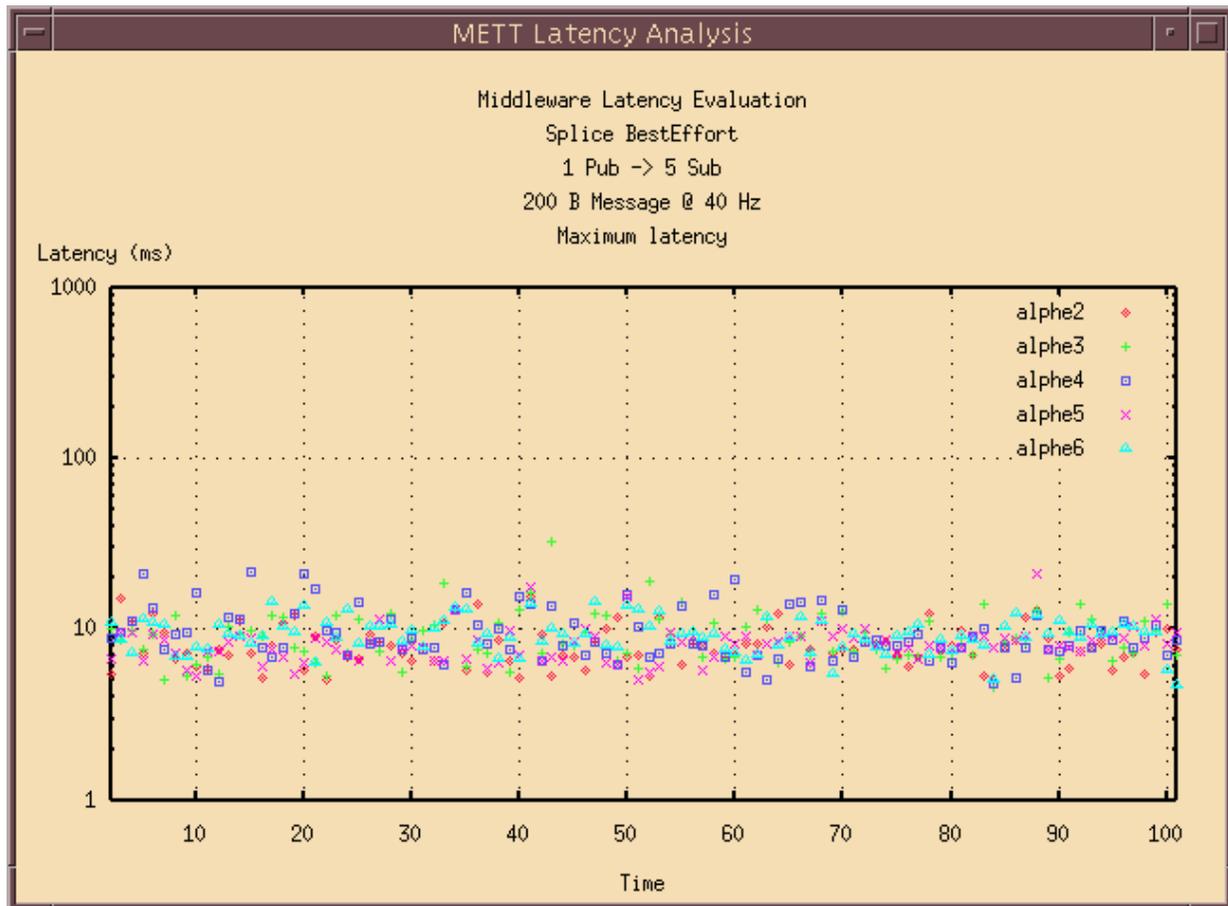
NDDS 1 Publisher to 5 Subscribers Reliable - Maximum Latencies



Maximum Latencies of One Publisher Sending Data to Five Subscribers – NDDS Reliable/FIFO



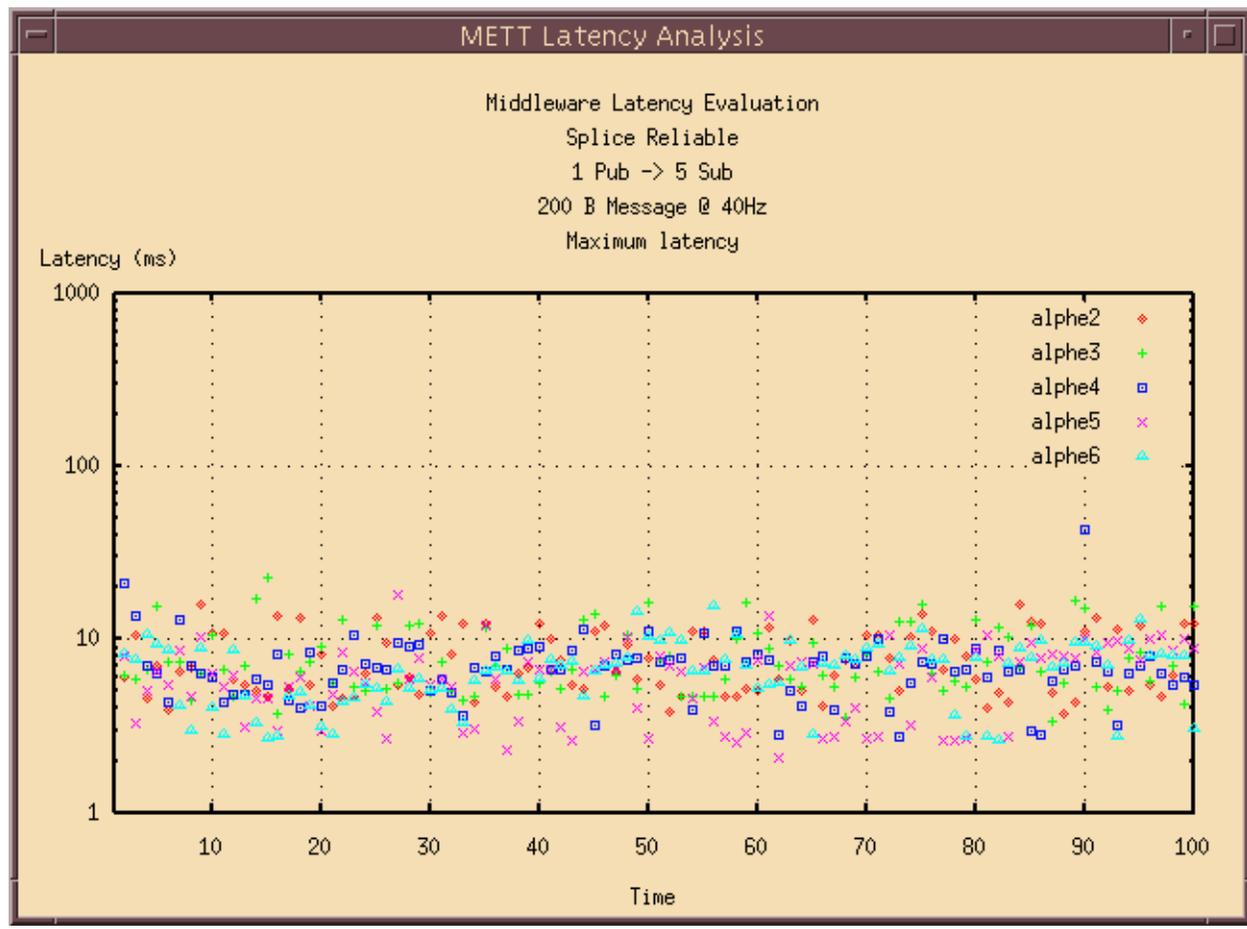
Splice 1 Publisher to 5 Subscribers Best Effort - Maximum Latencies



Maximum Latencies of One Publisher Sending Data to Five Subscribers – SPLICE Best Effort



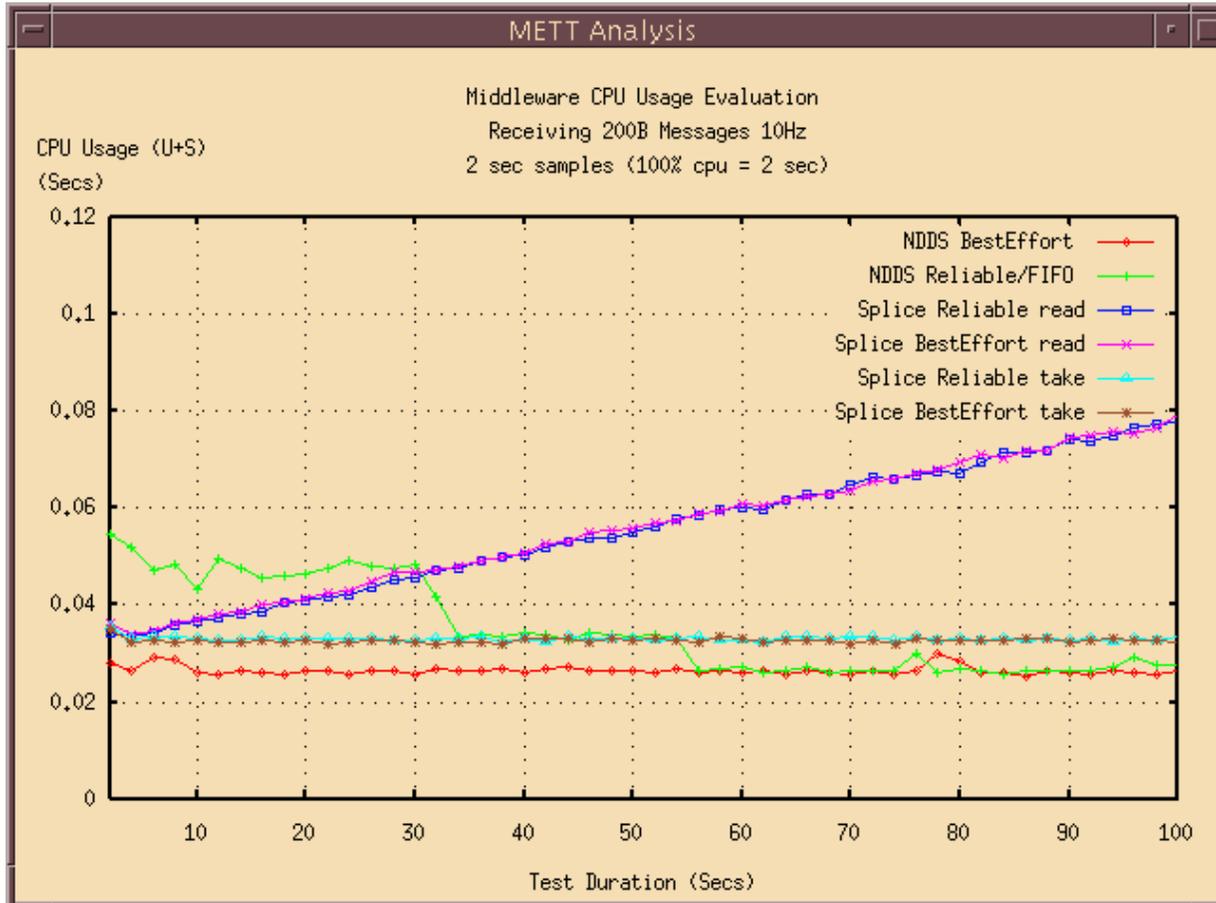
Splice 1 Publisher to 5 Subscribers Reliable - Maximum Latencies



Maximum Latencies of Publisher Sending Data to Subscribers – SPLICE Reliable



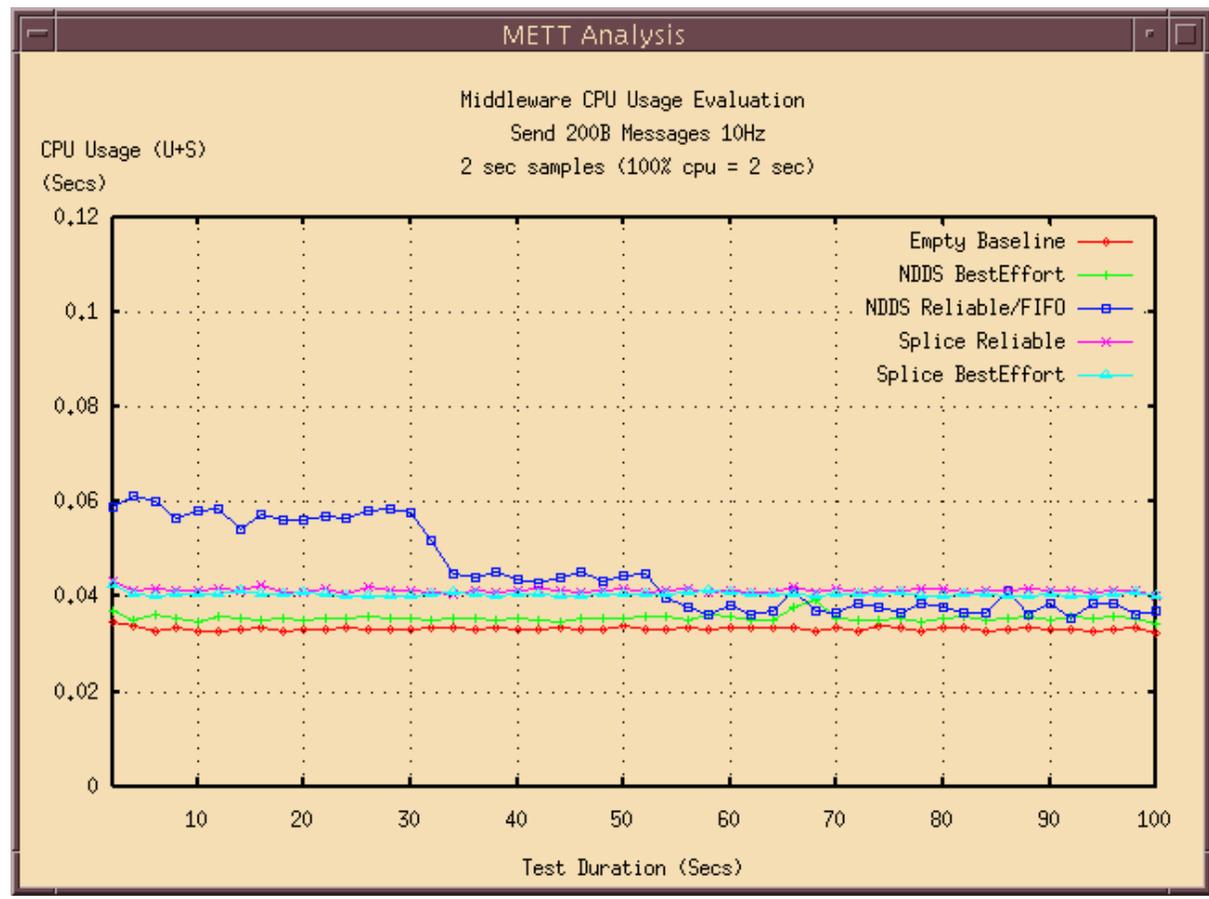
Subscriber CPU Usage



Middleware Solaris CPU Usage For Subscribers
Take and read



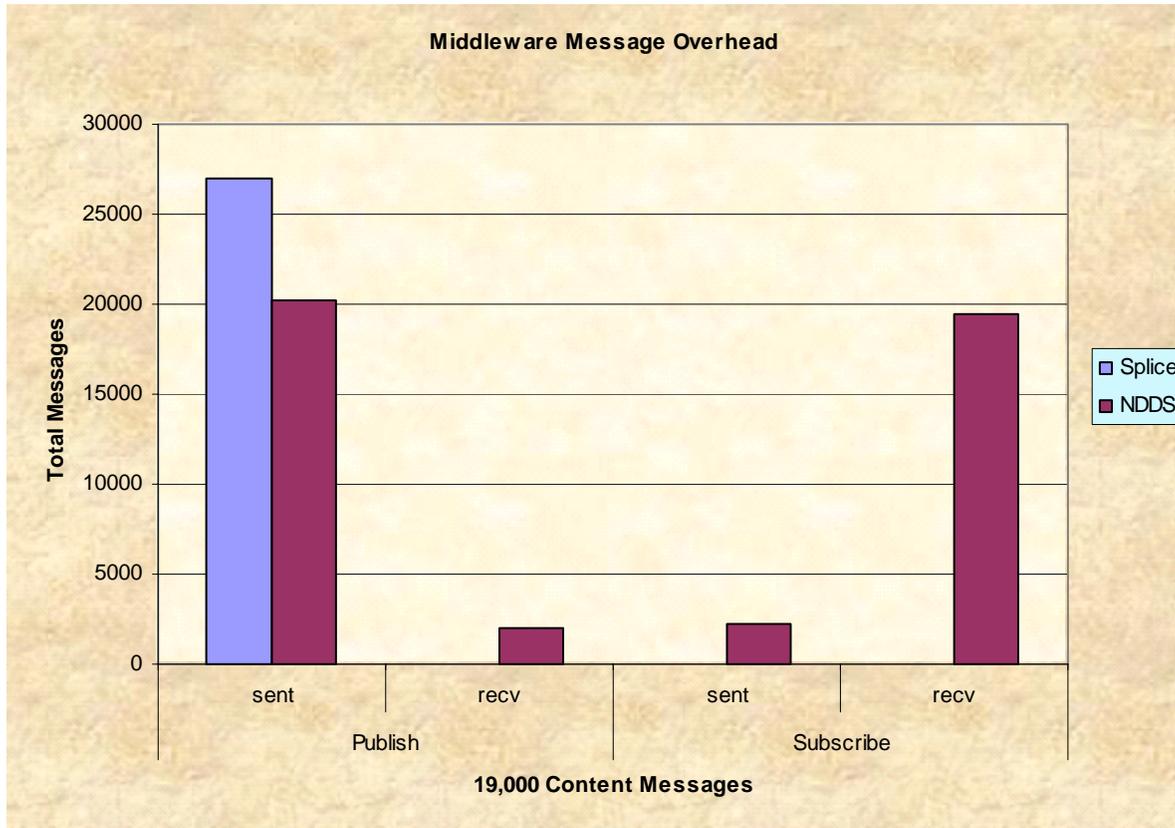
Publisher CPU Usage



Middleware Solaris CPU Usage For Publishers
Tool overhead included



IPC Message Overhead



Middleware Message Overhead



Summary

- ◆ Overall, both products perform well within the range of performance requirements of typical US Surface Navy Domain combat systems.
- ◆ Current indications are that both will provide good performance and scalability as a publish-subscribe middleware for combat system applications.
- ◆ Based on the results to date, it appears likely that performance of the products may not be a major differentiator, but other features not examined in this evaluation, such as life-cycle cost and support may influence the selection of one product over the other.



What's Next?

- ◆ **Testing of DDS-compliant products as they become available**
- ◆ **Evaluation of marshalling overhead associated with different, domain applicable IDL representations**
- ◆ **Evaluation of cross-platform performance issues, including marshalling**
- ◆ **Performance with large message sizes**
- ◆ **Performance on real-time operating systems such as LynxOS, RT Linux**