

Reliable UDP (RDP) Transport for CORBA

Voula Fotopoulos

Catherine Heaberlin

January 10, 2002

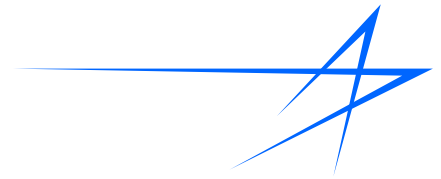
(voula.fotopoulos@lmco.com, catherine.t.heaberlin@lmco.com)

LOCKHEED MARTIN



Naval Electronics & Surveillance Systems - Surface Systems

Agenda



- ***Background – AEGIS Combat System***
- ***Distribute communication requirements and issues***
- ***Overview of Reliable UDP Communication***
- ***Reliable UDP transport for CORBA implementation challenges***
- ***Performance results***
- ***Conclusion***

Aegis Combat System

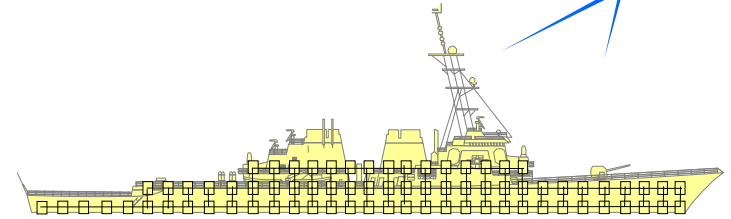
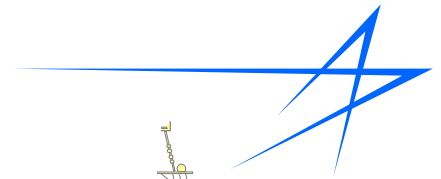
“The shield of the fleet...”

- ***A Highly Integrated Total Ship Combat System - A “System of Systems”***

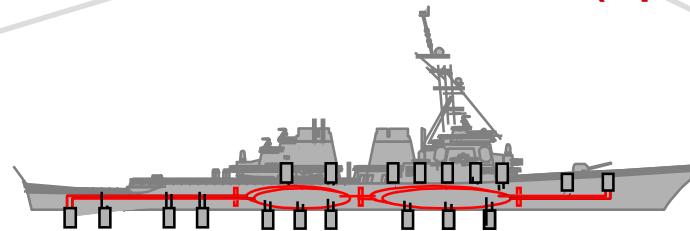


- ***Aegis Weapon System (AWS) primary mission is the Anti-Air Warfare (AAW) capability of the Combat System***
 - ***Detection, control and engagement for air targets***
 - ***Very large and complex***
 - ***Real-Time intensive with very demanding loop closure and control system response time requirements***
- ***Long-Standing Development/Production Program***
 - ***CG-47 Ticonderoga Class Cruisers deployed***
 - ***DDG-51 Arleigh Burke Class Destroyers ongoing***
 - ***Evolving requirements drive continual improvements via Baseline upgrade program***

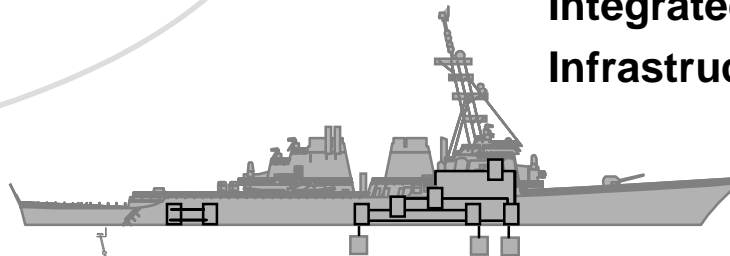
Aegis Processing Roadmap



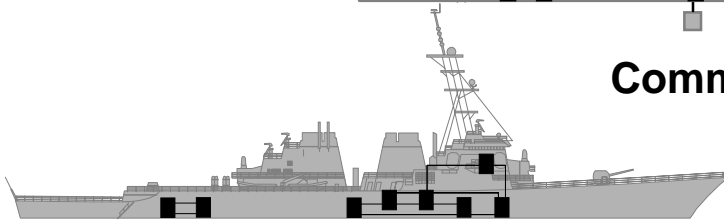
**Fully Distributed
(Open Architecture)**



**Integrated COTS Processing
Infrastructure**

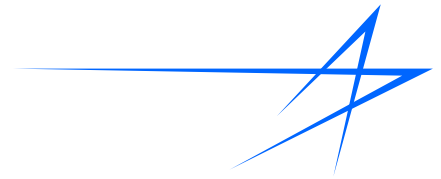


Commercial Co-Processing



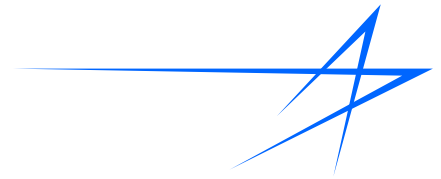
**Military Processors and
Interconnects**

Requirements



- ***Short latency and high probability of delivery for a relatively small number of time-critical messages in the system***
- ***Low CPU utilization everywhere to preserve resources needed for tactical processing***
- ***High throughput in a few high-volume pathways***
- ***Commercially supported products***

Issues



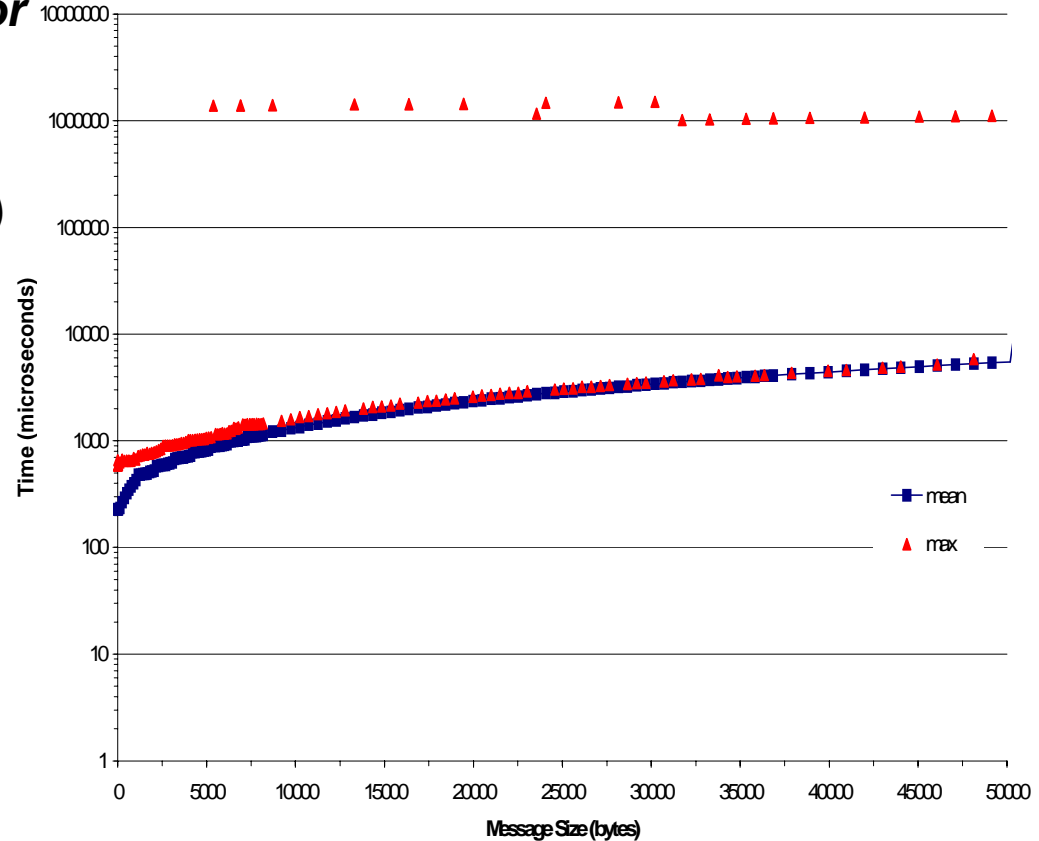
- **Commercial CORBA software out of the box is not suitable for all real-time systems**

- **Standard CORBA uses IIOP (mapping of GIOP over TCP/IP)**

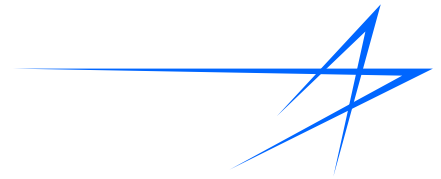


- **TCP/IP obstructs real-time predictability**
 - **Retries and delays in acknowledgments**
 - **Connection loss detection**

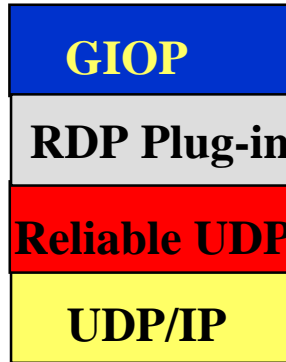
TCP Send with 1 byte Response Latency: Remote, 100,000 iterations
[LynxOS 3.1.0, Fast Ethernet]



Solution and Advantages

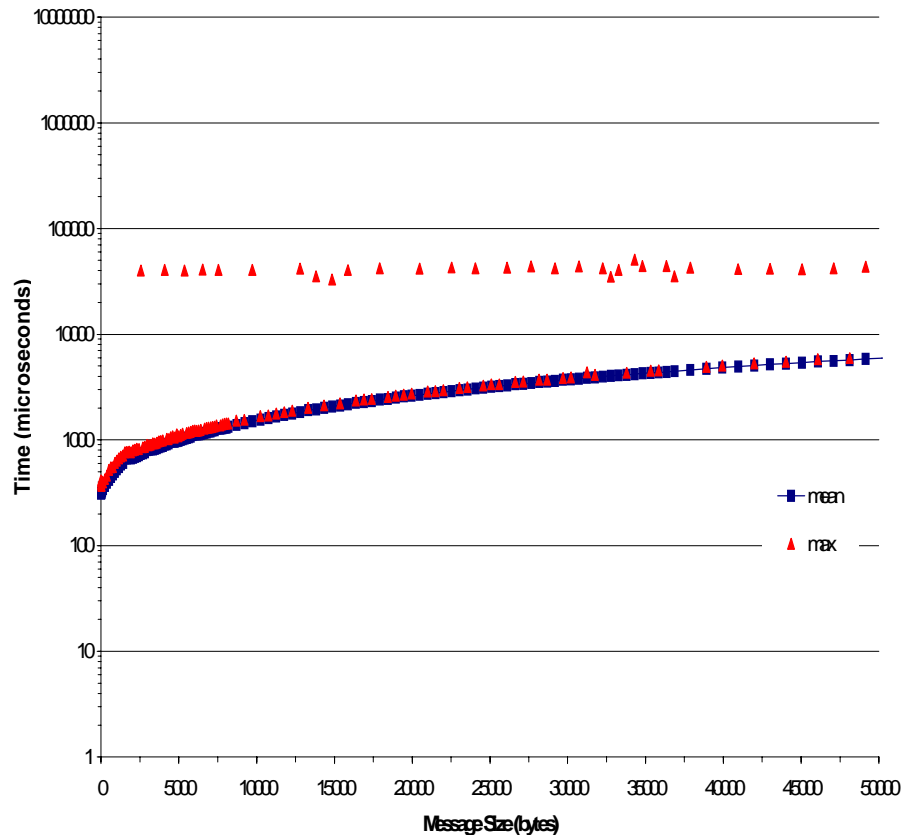


- **Implement a Reliable UDP/IP (RDP) transport as an alternative to TCP/IP for an ORB**



- **Control maximum latencies with configurable timeouts and retries**
- **Replace of existing capabilities with commercial products**
- **Retain the standards-based application interfaces of CORBA while adapting alternative transports for different environments**

Reliable UDP Sand with 1 byte Response Latency: Remote, 100,000 iterations
Ack Timeout 50ms [LynxOS3.1.0, Fast Ethernet]

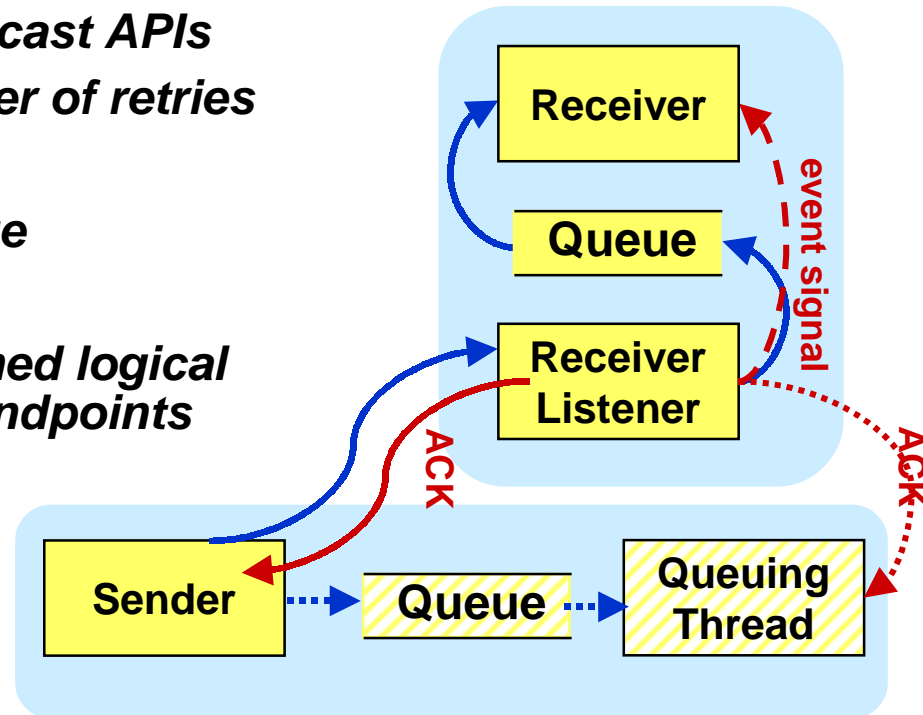


Reliable UDP Communication

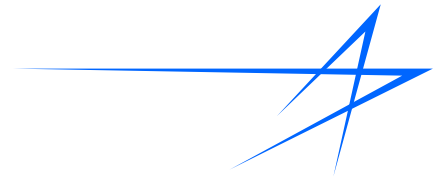


- **Reliable/Unreliable Unicast and Multicast APIs**
 - Configurable timeouts and number of retries
 - Flow control and buffer sizing
 - Optional outgoing message queue
- **Unidirectional messaging over a named logical channel that connects two or more endpoints**
- **Fault tolerant Directory Services provides registration services and endpoint updates**
- **Low overhead above standard UDP/IP**

Unicast

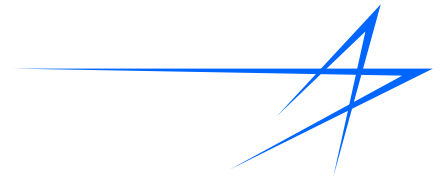


Reliable UDP (RDP) Transport for CORBA



- *OMG is progressing towards the adoption of a standard for a extensible transport framework*
- *Currently vendors are providing transport frameworks*
 - *Used ORBexpress RT 2.3.2 from Objective Interface Systems (OIS) on LynxOS 3.1.0*
- *Implementation*
 - *Developed concrete C++ classes that were derived from abstract classes provided by the OIS transport replacement framework*
 - *Integrated Reliable UDP into these classes*
 - *Registered the classes with the ORB*

RDP Implementation Challenges

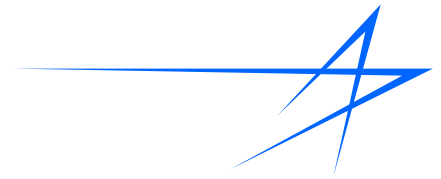


- ***Establishing a mapping of GIOP to the new transport***
 - ***Required working around violations of the GIOP assumptions (i.e. transport is connection-oriented, reliable, and byte stream oriented)***

- ***Defining addressing information***
 - ***Two strings used for construction of channel names***
 - ***“server_name” is the channel name used for establishing connections***
 - ***“client_name” is a superset of the two channel names used for communicating requests and responses***

- ***Identifying Protocol Properties that are used to control configurable aspects of the transport***
 - ***Acknowledgement timeout, retries, buffer sizes, flow control, thread priorities, keep alive, interface name specification***

RDP Implementation Challenges



■ ***Adding Connection Semantics***

■ ***Connection***

- ***Defined as two unidirectional channels***
- ***Used for reading and writing requests and responses***

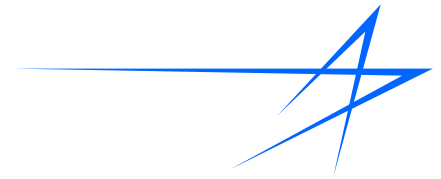
■ ***Listen for Connection Requests***

- ***Active object permanently opens the receiving side of “server_name” channel and waits for requests***
- ***Initiates a new connection for each received request***

■ ***Establish Client Connection***

- ***Open sending side of “server_name” channel***
- ***Create a connection with sending and receiving channel names***
- ***Sends connection request with unique channel names***
- ***Closes “server_name” channel, allowing other clients access***

RDP Implementation Challenges

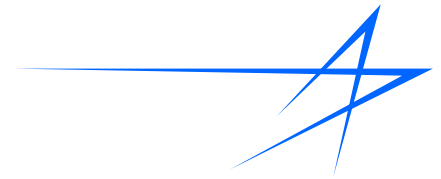


- ***Reading and Writing***
 - ***Minimize size dependent overhead***
 - ***Controlling for message size constraints***

- ***Failure and recovery semantics***
 - ***Determine if connection has failed or has been closed***
 - ***Control of Reliable UDP reconfiguration processing***

- ***Memory Management***
 - ***Avoid memory leaks***

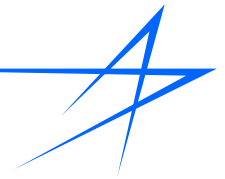
Performance Results



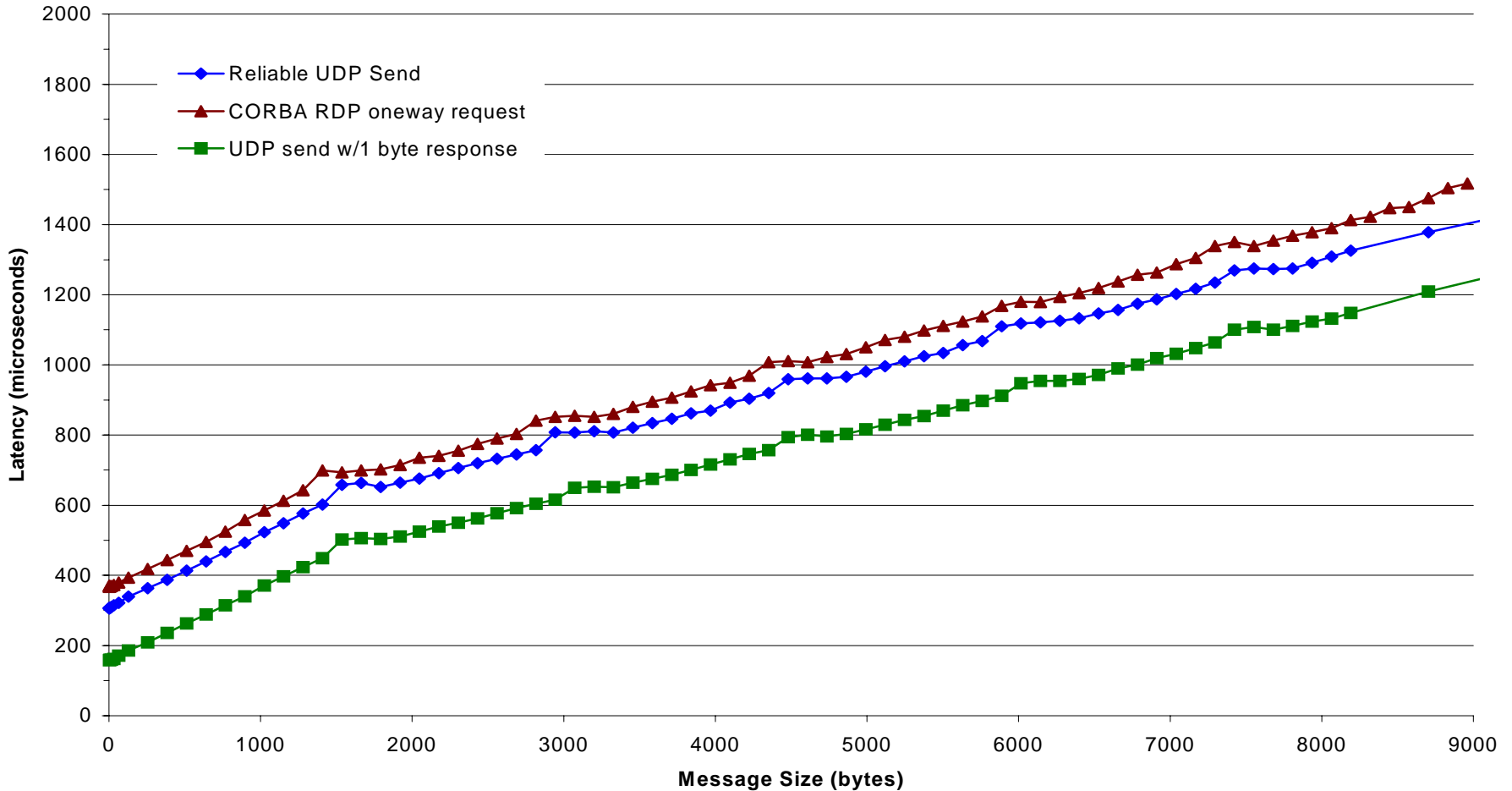
- ***Oneway Request Latency***
 - ***CORBA with RDP (sync_with_server)***
 - ***Reliable UDP Send***
 - ***UDP/IP Send with 1 byte response***

- ***Synchronous Request Response Latency***
 - ***CORBA with RDP***
 - ***Reliable UDP Send with 1 byte response***
 - ***CORBA with TCP***

Oneway Request Latency

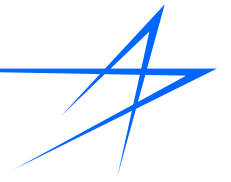


CORBA RDP ONEWAY Request Latency (sync_with_server): Remote, Octet Sequence
[LynxOS 3.0.1, Fast Ethernet, ORBexpress 2.3.2]

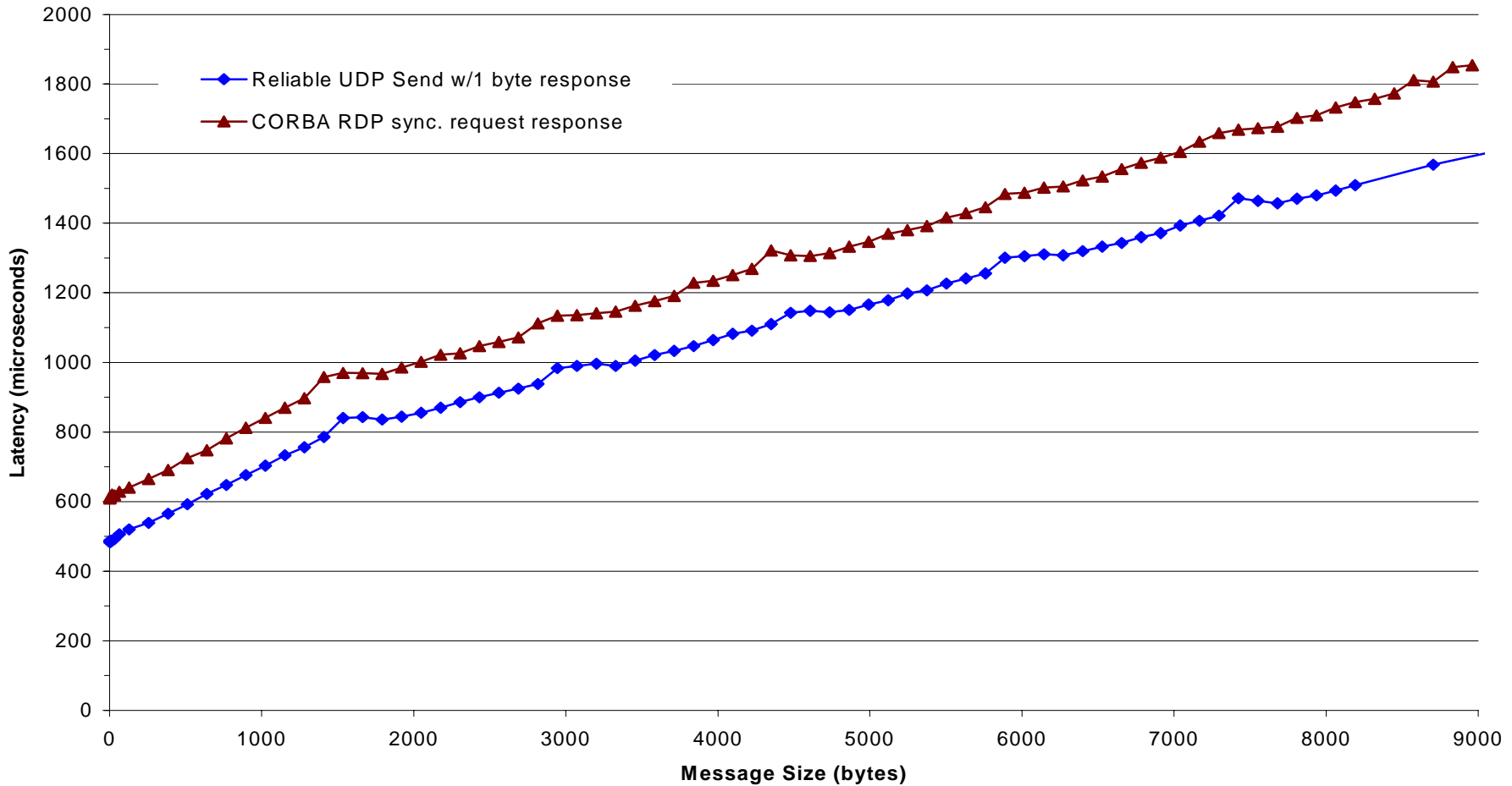


ORB with RDP overhead ranges from 36 .. 104 μsec for message sizes less than 8Kb

Sync Request Response Latency

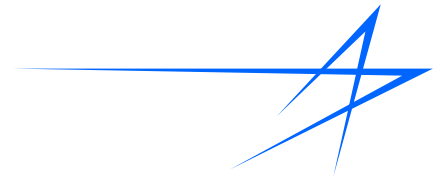


CORBA RDP Synchronous Request Response Latency: Remote, Octet Sequence
[LynxOS 3.0.1, Fast Ethernet, ORBexpress 2.3.2]

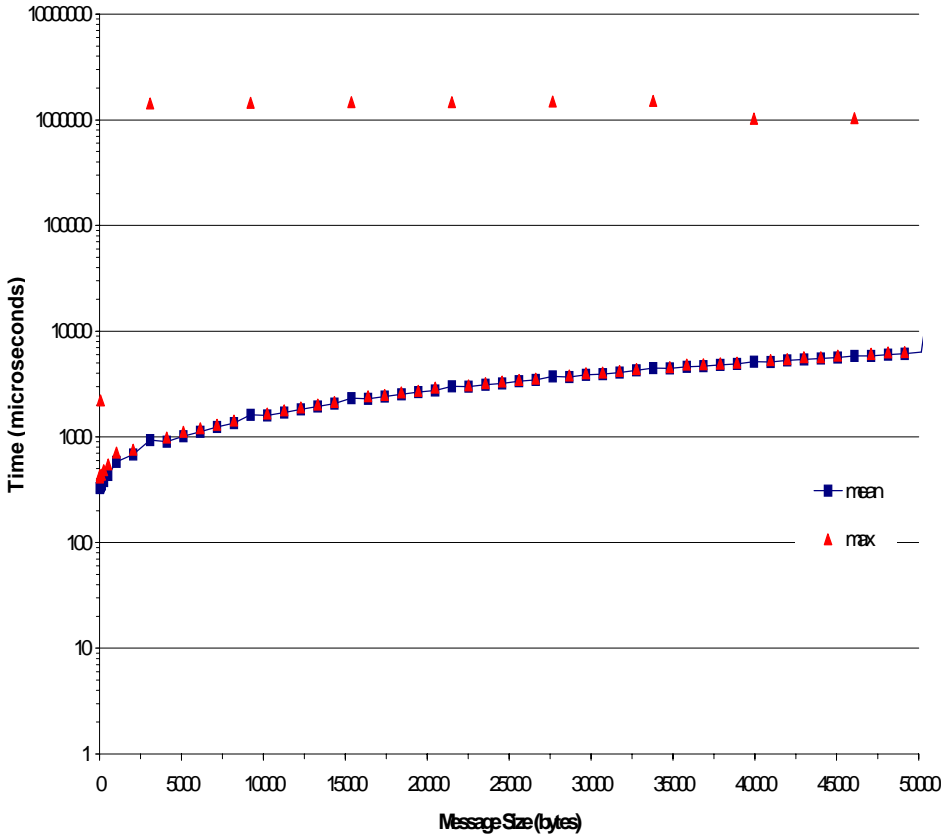


ORB with RDP overhead ranges from 120..239 μsec for message sizes less than 8Kb

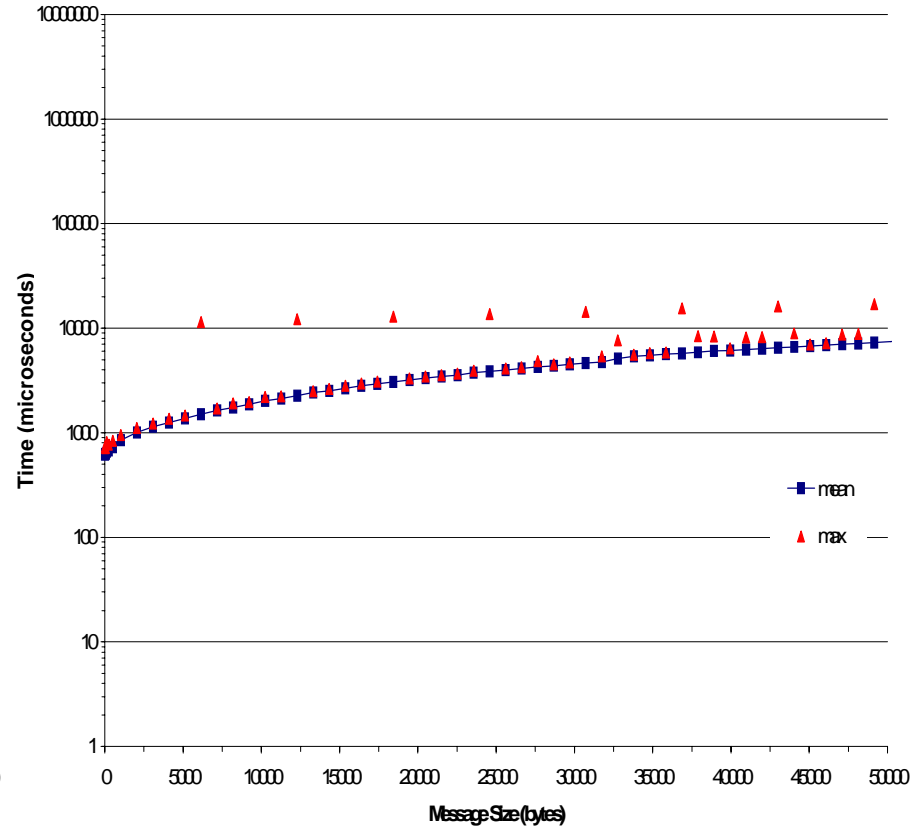
CORBA TCP vs. CORBA RDP



CORBA TCP Synchronous Request Response Latency: Remote, Double Sequence
[LynxOS3.1.0, Fast Ethernet]



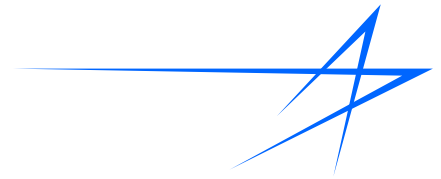
CORBA RDP Synchronous Request Response Latency: Remote, Double Sequence
RDP QoS Ack Timeout 20ms, Retries 3
[LynxOS3.1.0, Fast Ethernet]



CORBA TCP results show excessive maximum latencies greater than 1 second

CORBA RDP maximum latencies are controlled by timeout durations and retries

Conclusion



- *It is feasible implement an alternative transport to TCP/IP for CORBA to support real-time distributed communications*
- *CORBA with RDP does not suffer from non-deterministic behavior and excessive maximum latencies as CORBA with TCP*
- *Important issues relevant to extensible transport frameworks*
 - *Easy to understand design*
 - *Zero copy required*