Meeting the Demands of Changing Operating Conditions at Runtime Using Adaptive Programming Techniques for Distributed, Realtime Embedded Computing

Rick Schantz (schantz@bbn.com)

Joe Loyall (jloyall@bbn.com)

BBN Technologies
Cambridge, Ma.

HPEC Workshop 2002
September 25, 2002

BBN TECHNOLOGIES
A Verizon Company

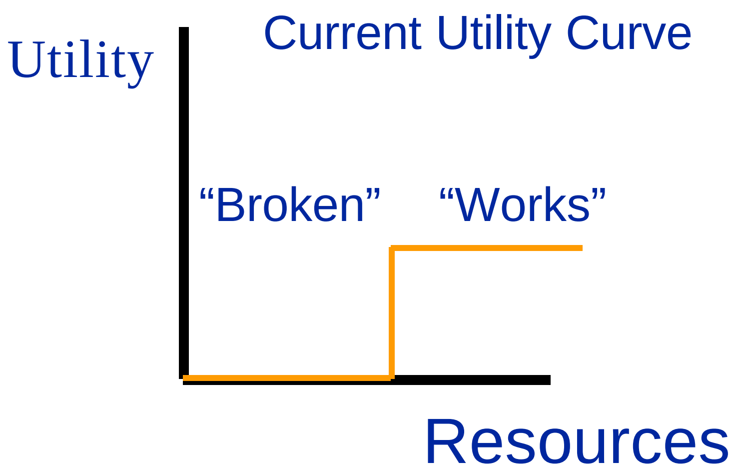
Outline

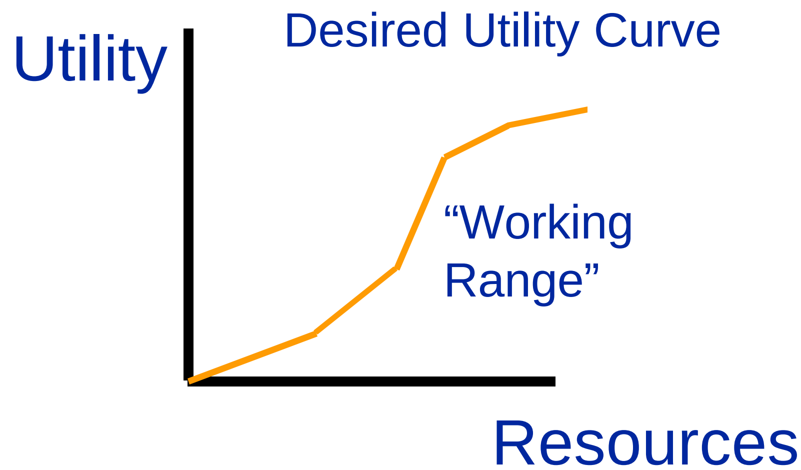
- A Point of View & Background
- Technologies for Managed Behavior in Rapidly Changing Environments
- Examples we've built, tested and evaluated
  - WSOA, UAV
- Some Lessons Learned and Challenges Going Forward

Overview

- High Performance Isn't Only About Achieving High Speed (but that as well)
- Its also about priority, precision and safety ...and sustaining high performance over changing environments
- We need to maintain an appropriate capability across significant events for the capability to be truly useful and applied to critical problems
- Systems operating in and across the real physical universe (embedded systems) encounter much more volatility
- It's necessary to build systems differently on a more flexible, manageable technology base to reflect this change
- Instead of users adapting to what systems can deliver, systems need to easily adapt to what the situation demands

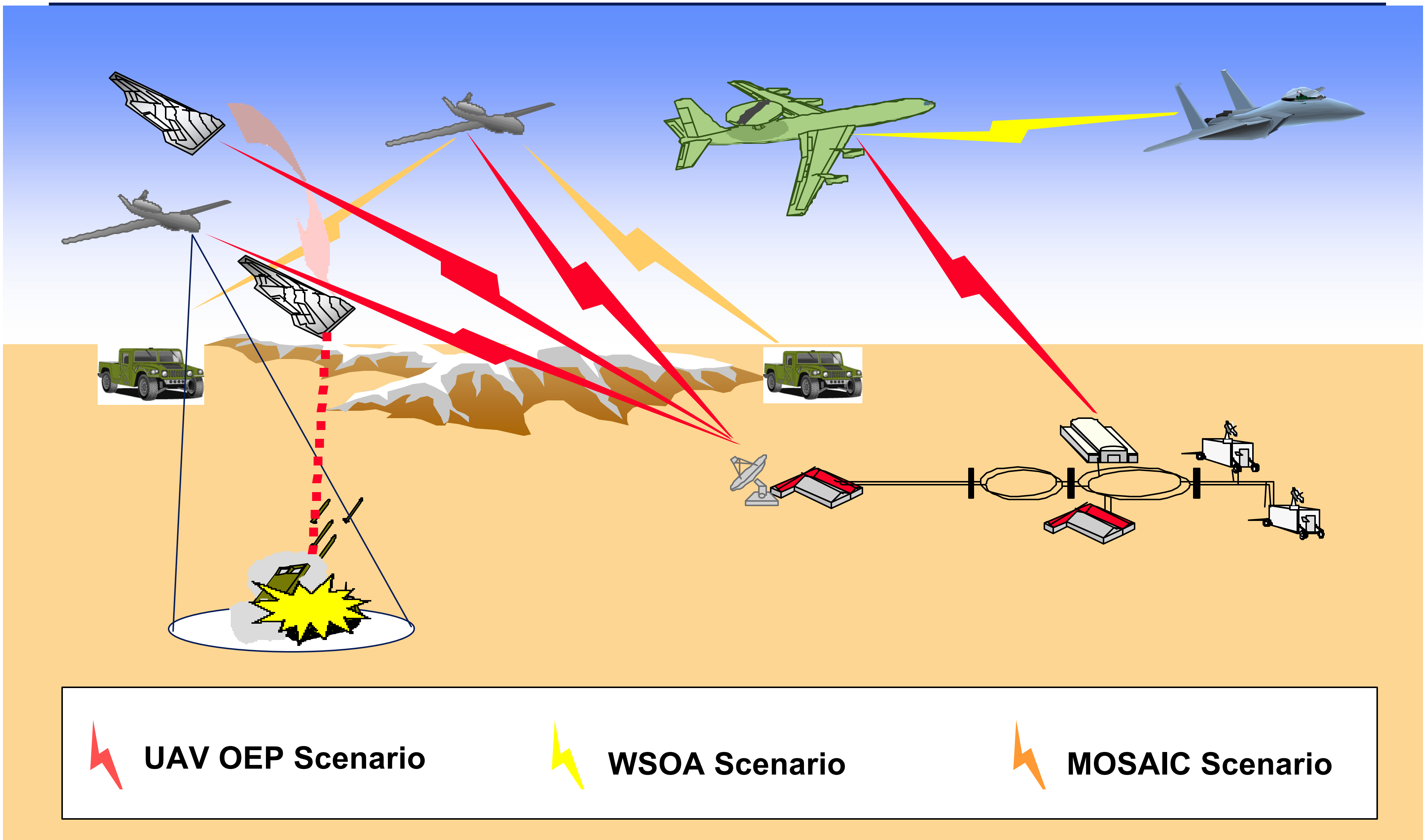
Network Centric Applications Need to Be Aware of Their Operating Context and Adapt Their Behavior to Match





- DRE contexts are more volatile than backplanes and desktops, and less likely to be overprovisioned
- Requirements may change with the current situation
- Truly dependable systems can be expected to do the "right/best thing" under the prevailing circumstances at all levels of available resources
- This requires support for adaptive, runtime behaviors and attention to finer grained real time resource management decisions
- Middleware provides and enables the additional structure for organizing adaptive behavior and tradeoffs of the different QoS dimensions

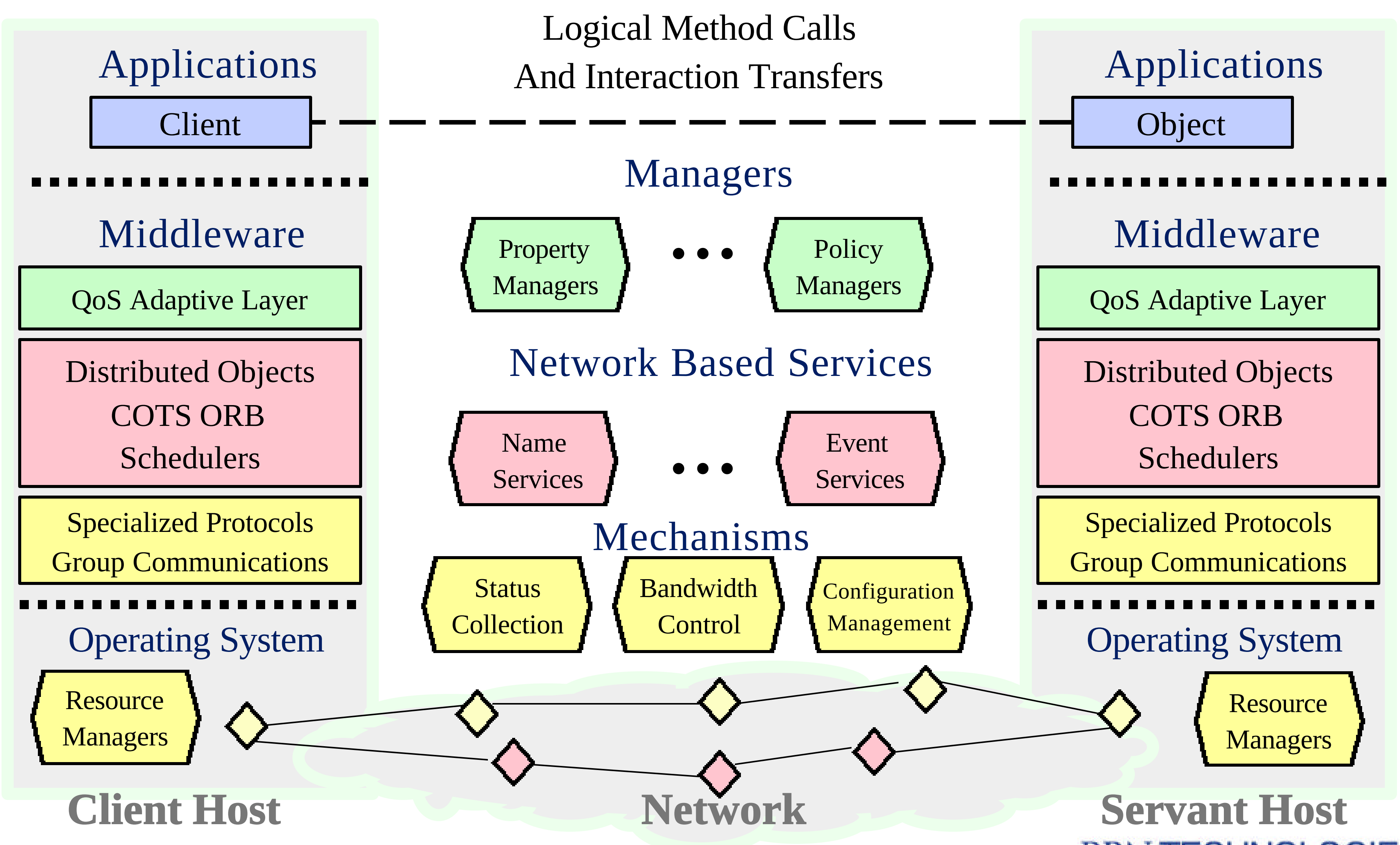
Embedded Application Context



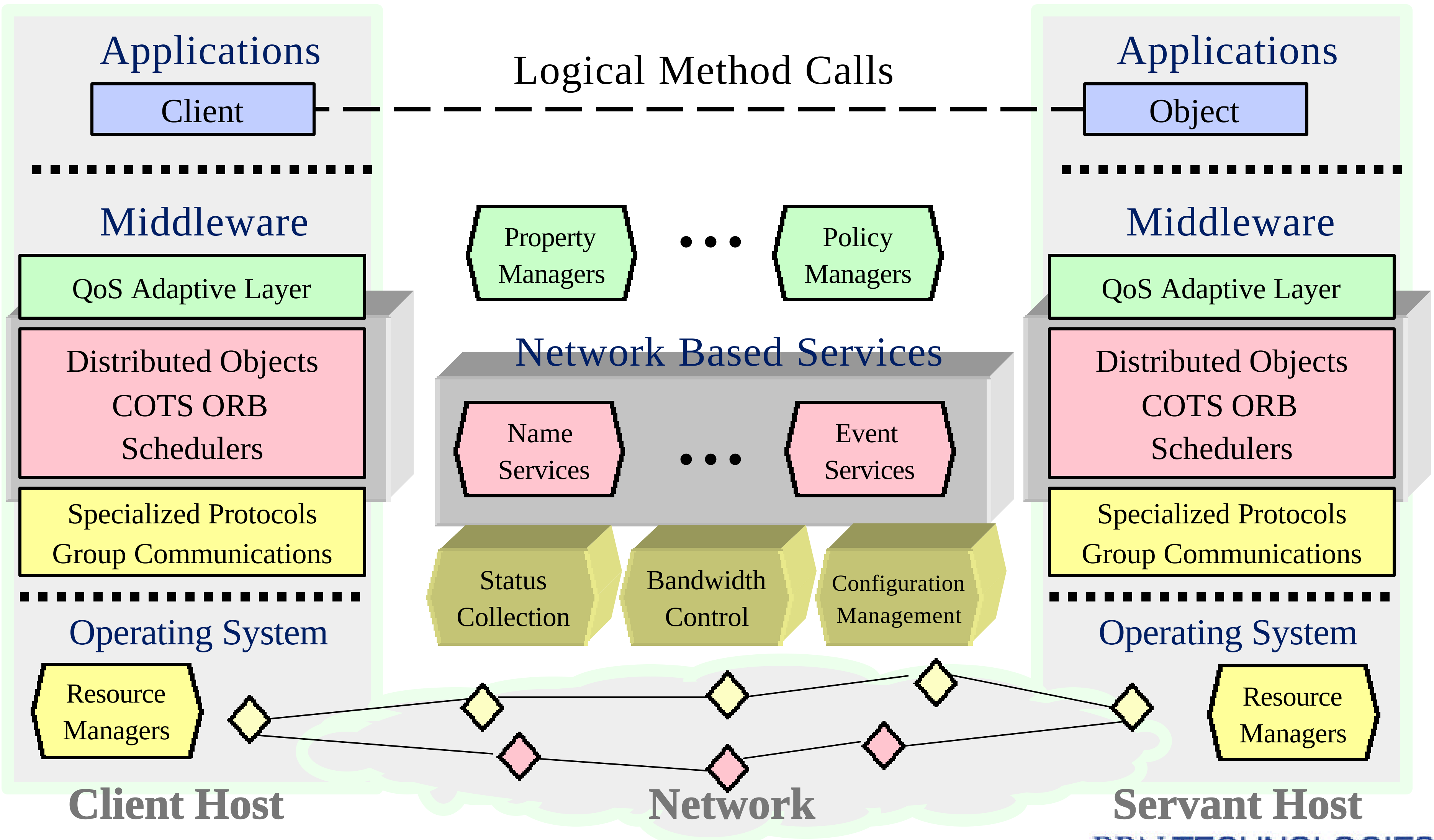
Outline

- A Point of View & Background
- Technologies for Managed Behavior in Rapidly Changing Environments
- Examples we've built, tested and evaluated
  - WSOA, UAV
- Some Lessons Learned and Challenges Going Forward

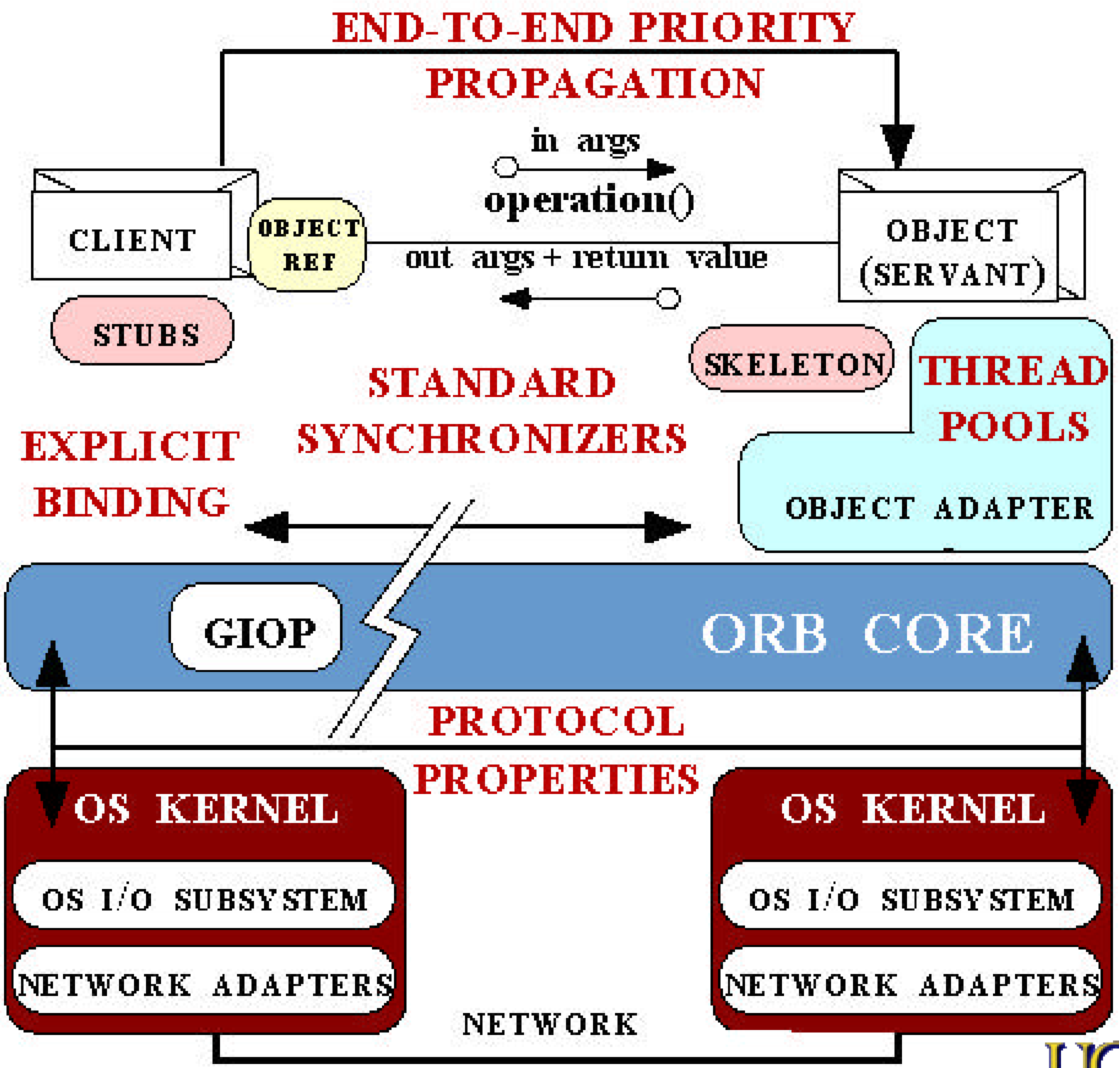
Network Centric QoS Interface and Control as Part of a Layered Architecture



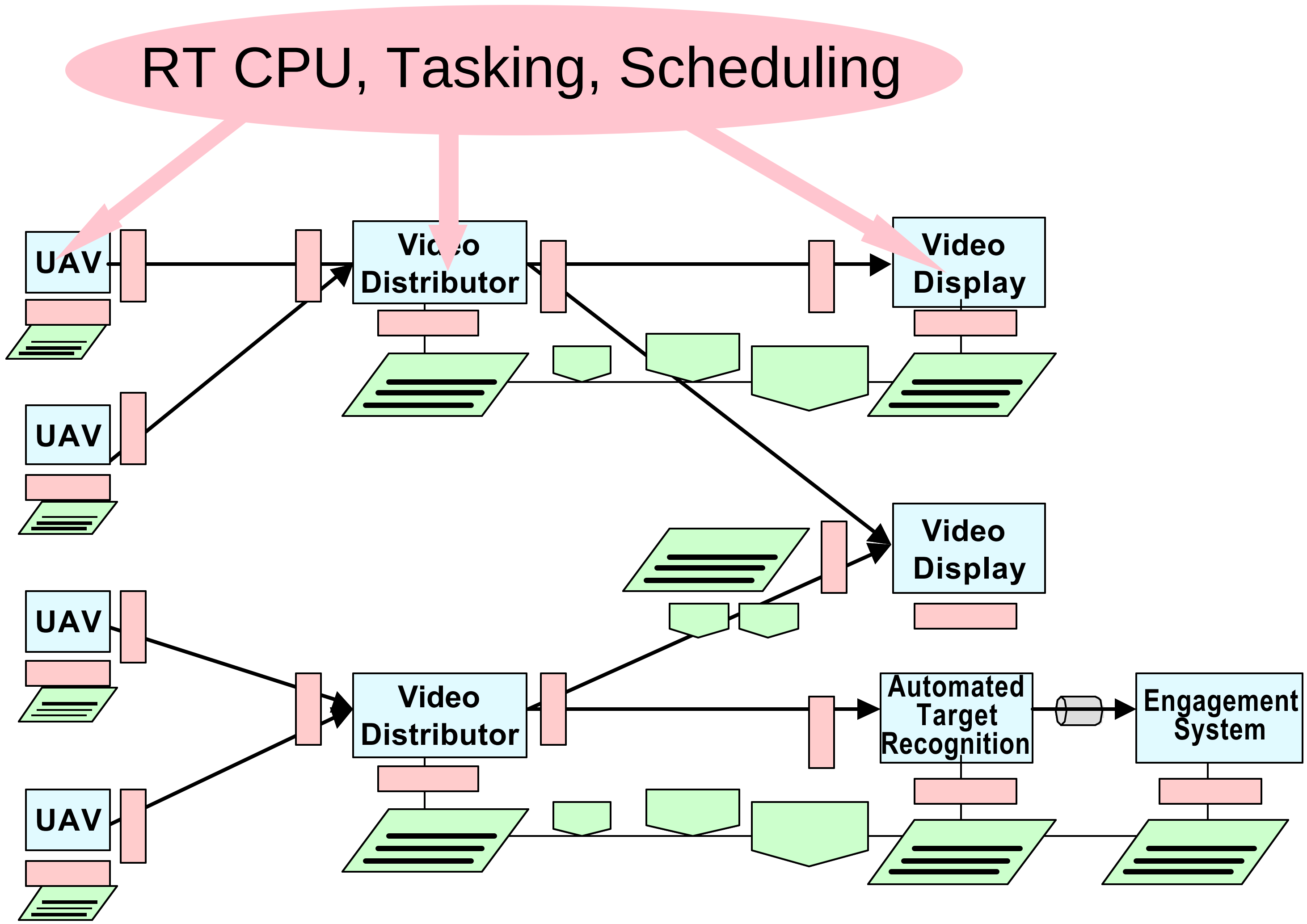
Lower Level Middleware and Infrastructure Control



TAO: A Real-time CORBA Compliant ORB

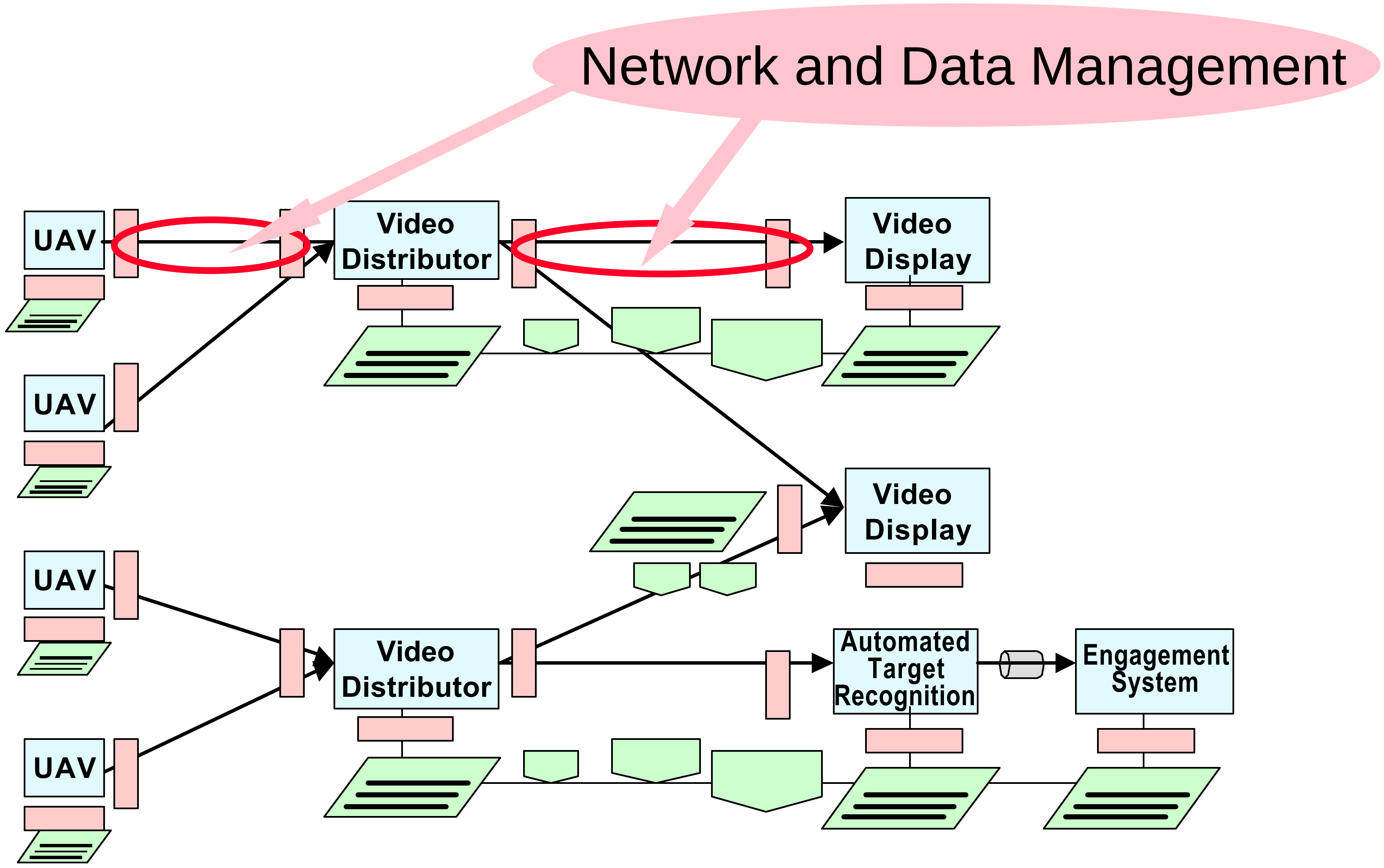


End to End Resource/QoS Management



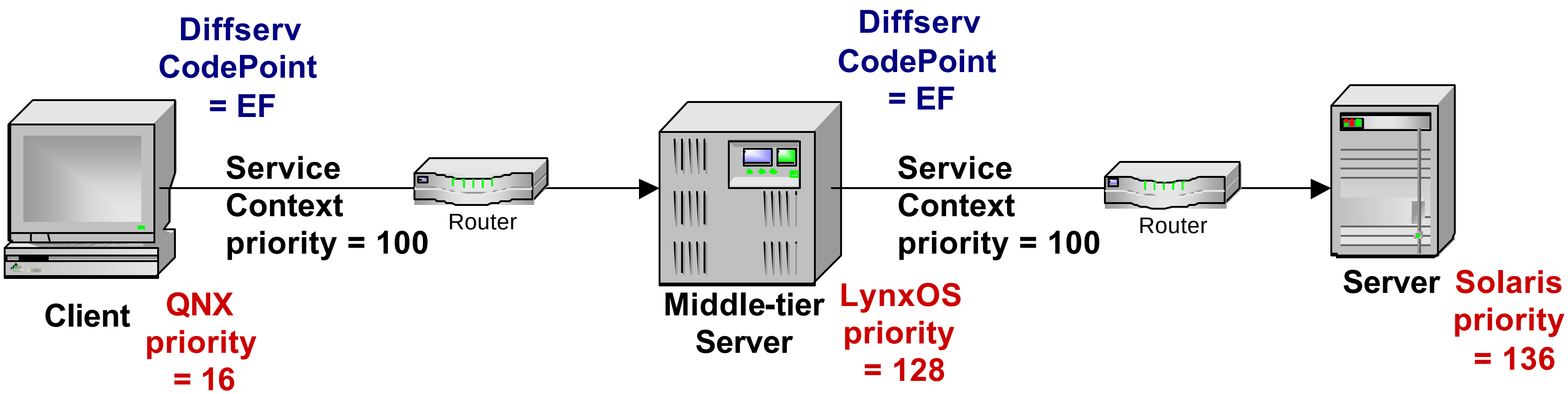
BBN TECHNOLOGIES
A Verizon Company

End to End Resource/QoS Management



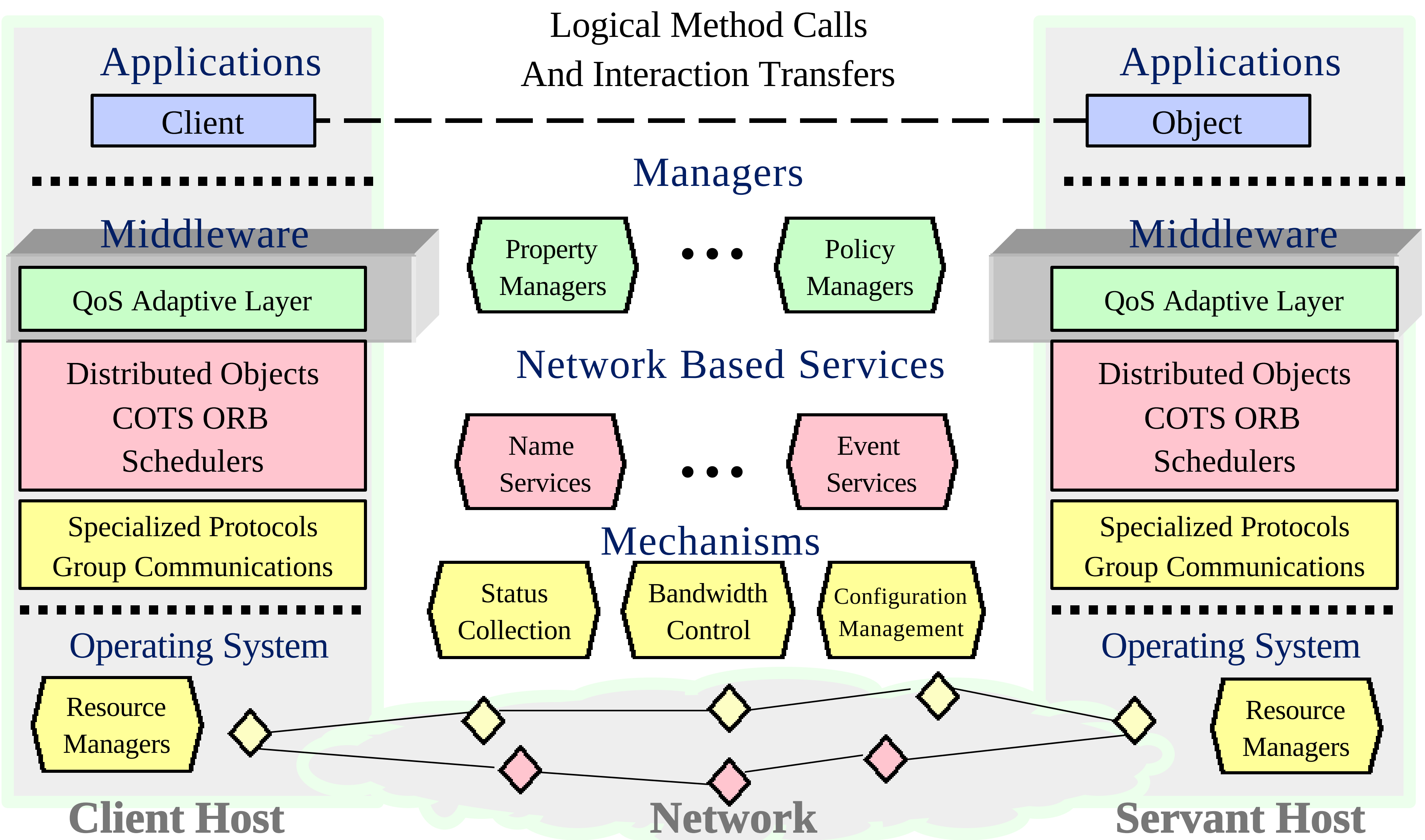
Examples: RTCORBA with Diffserv Capability Preserving End-to-End Priorities

- Existing priority in RTCORBA used for OS-level task scheduling across distributed nodes
- Our enhancement to RTCORBA uses this priority to set Diffserv field in IP packets associated with a specific CORBA call
- Network treats packets differently based on value of Diffserv field; can be used as another mechanism for end-to-end QoS

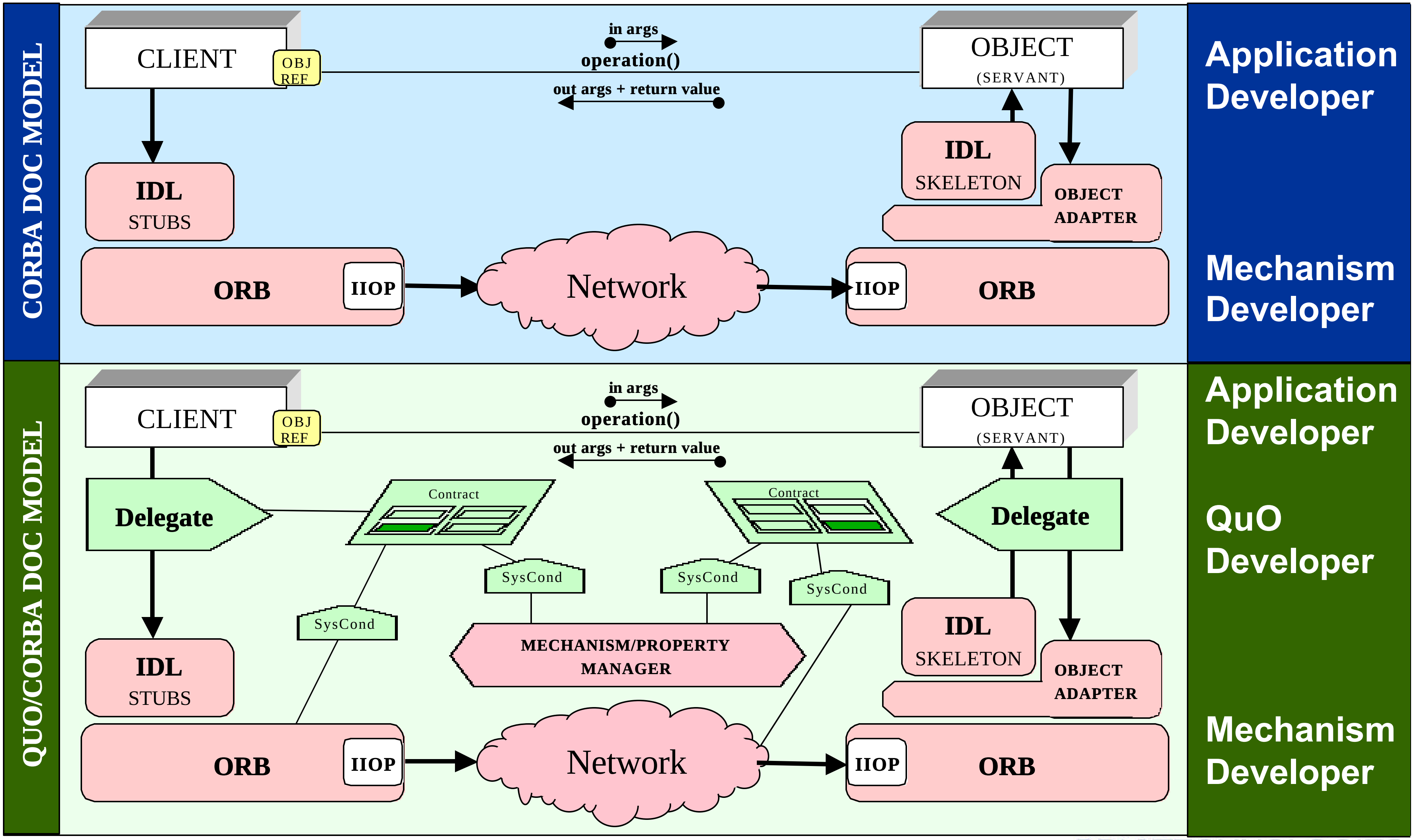


BBN TECHNOLOGIES
A Verizon Company

Formalizing Adaptive Behavior

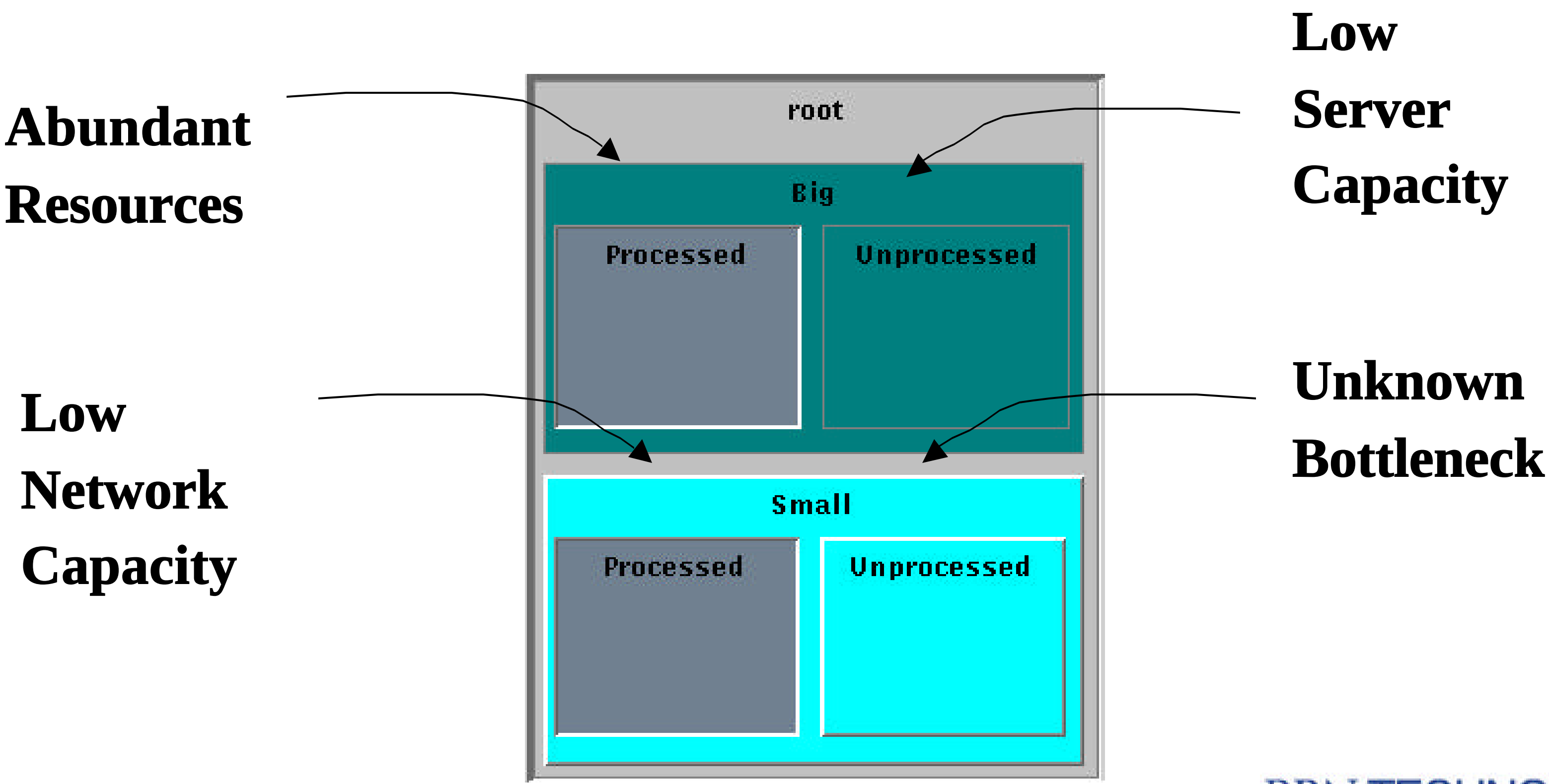


QuO is middleware that offers an application the ability to adapt to a changing environment in which it is running



Contracts Summarize System Conditions into Regions Each are Appropriate for Different Situations

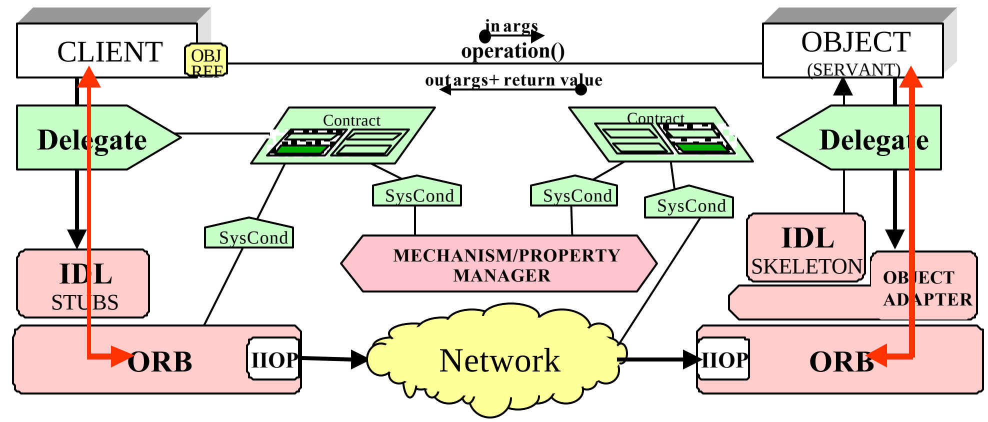
- Contract defines nested regions of possible states based on measured conditions
- Predicates using system condition objects determine which regions are valid
- Transitions occur when a region becomes invalid and another becomes valid
- Transitions trigger adaptation by the client, object, ORB, or system



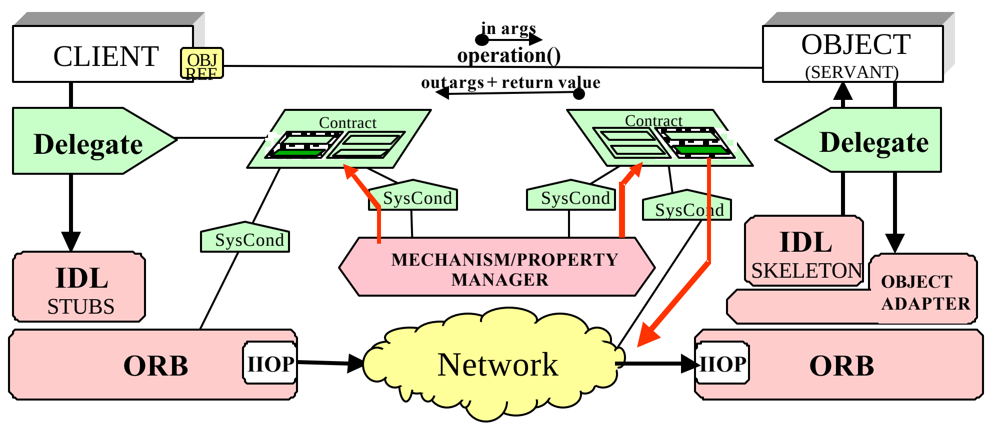
Panel From QuO GUI

In-Band and Out-of-Band Adaptation and Control Using QuO

- *In-band* adaptation provided by the delegate and gateway
  - A delegate decides what to do with a method call or return based upon the state of its contract
  - Gateway enables control and adaptation at the transport layer



- *Out-of-band* adaptation triggered by transitions in contract regions
  - Caused by changes in the system observed by system condition objects

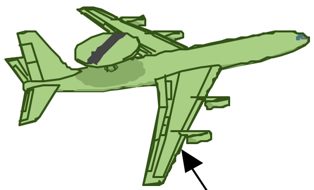


BBN TECHNOLOGIES
A Verizon Company

Outline

- A Point of View & Background
- Technologies for Managed Behavior in Rapidly Changing Environments
- Examples we've built, tested and evaluated
  - WSOA, UAV
- Some Lessons Learned and Challenges Going Forward

WSOA: Enroute Adaptive Planning

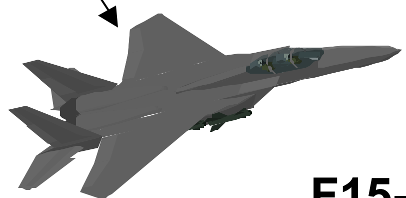


Airborne C2 Node

- Compiles Virtual Target Folder
- Retasks Enroute Strike
- Collaboration with Warrior to replan route
- IDL Interface

JTIDS Net

- Link-16 GIOP
- Browser Requests
- Low Volume Imagery

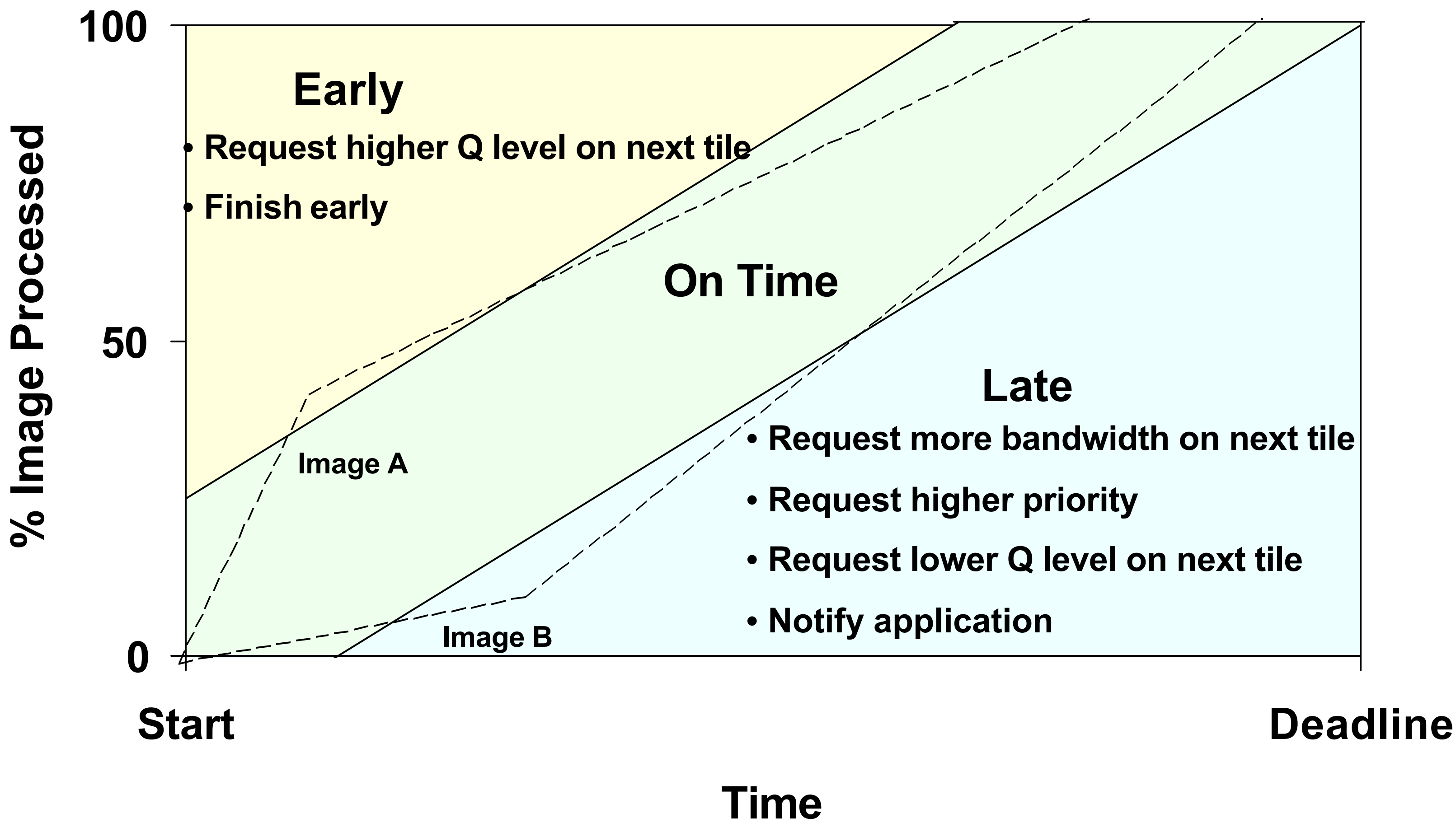


F15-Warrior

- "Browser" Requests for Target and Imagery data
- Collaboration with C2 Node for Target Review and Mission Replan
- Previews Updated Mission Enroute
- IDL Interface

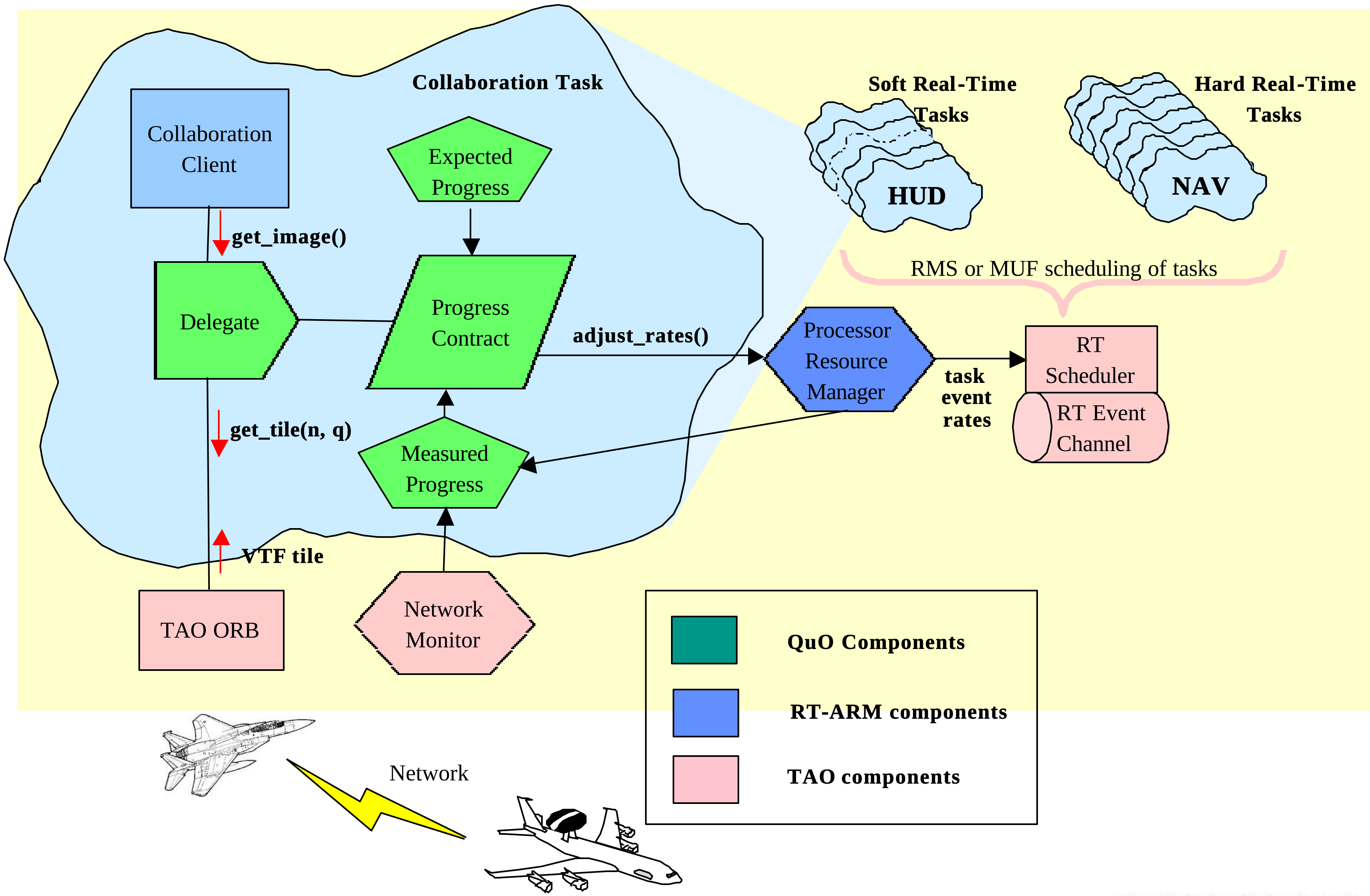
* Boeing WSOA Quarterly Review August 9 2000

QoS Adaptation Domain



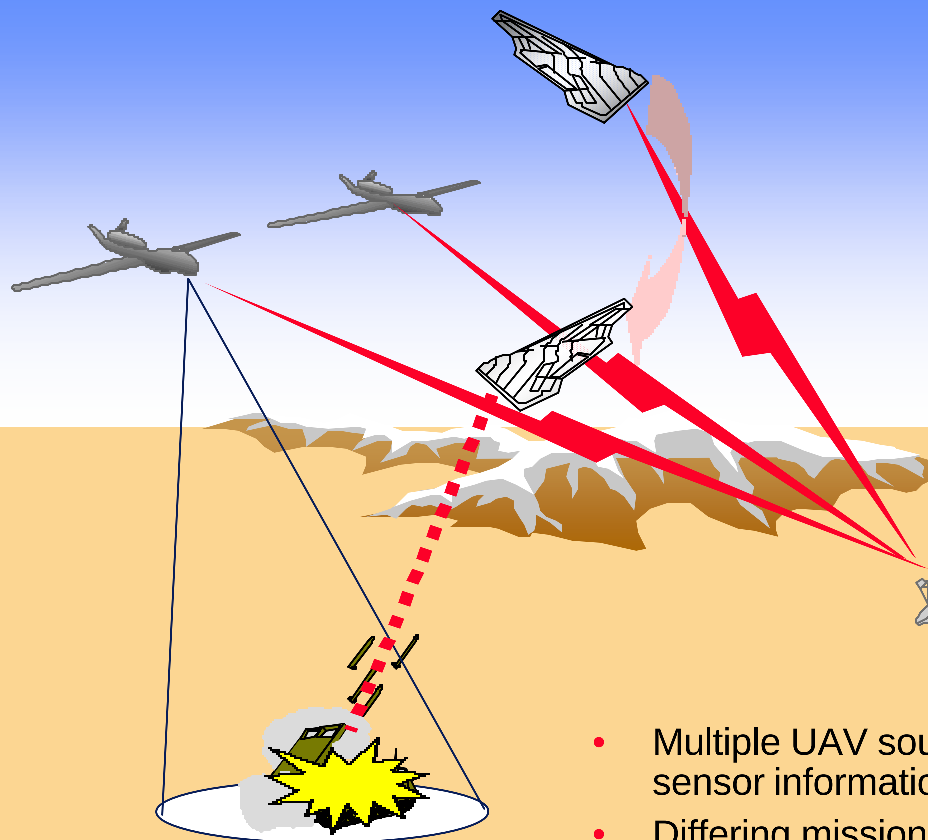
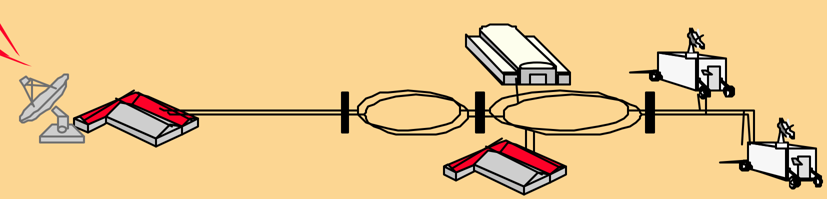
BBN TECHNOLOGIES
A Verizon Company

Adaptive Behavior Integrated with Advanced Resource Management



BBN TECHNOLOGIES
A Verizon Company

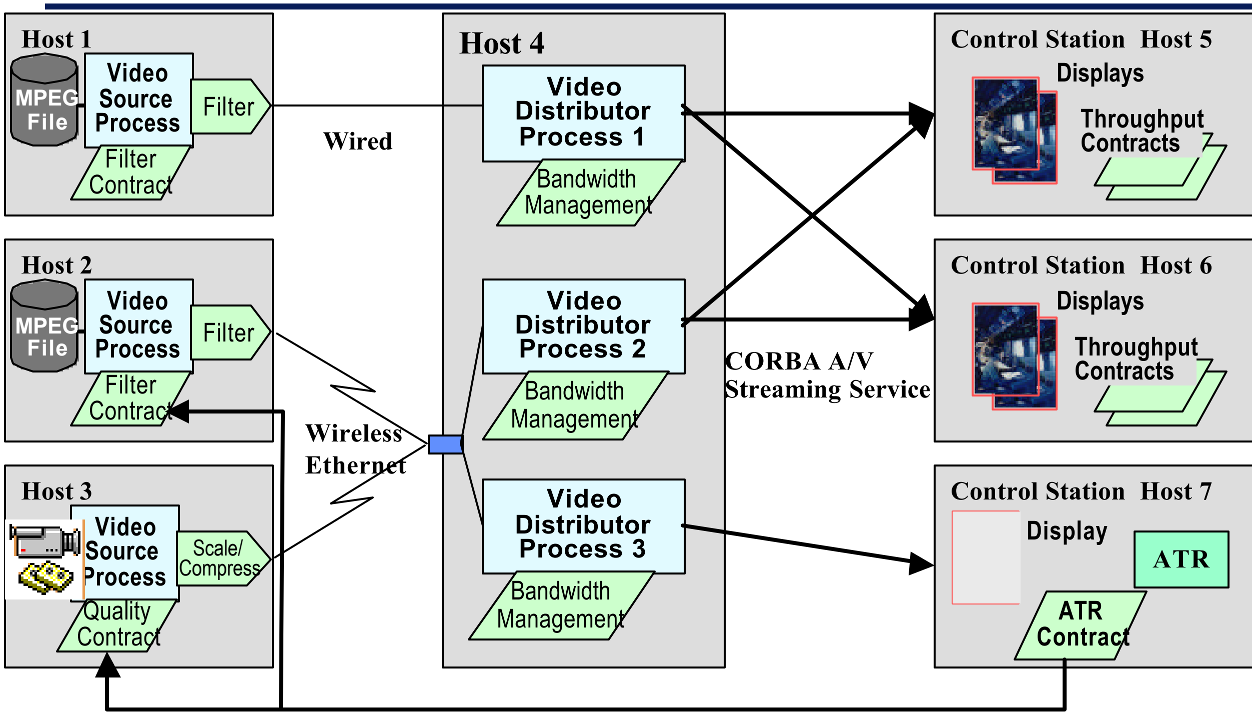
The UAV Concept of Operations

- Sensors on the UAVs gather video, radar, and other information and transmit them to control stations
- Operators at stations send commands to the UAVs to pilot them, control their sensors, and to locate and prosecute targets

- Multiple UAV sources requires management of resources for delivery of sensor information control commands
- Differing missions require tradeoffs of data content and form
- Fidelity of sensor information must be sufficient for manual or automatic recognition of target
- End-to-end delivery, processing, and use of sensor information must be frequent and fast enough to support prosecution of time-critical (possibly mobile) targets

Instantiating an Experimental Configuration



Maintaining QoS requirements *under dynamic conditions* , making appropriate tradeoffs using QuO contracts

Uses off-the-shelf components

- QuO adaptive middleware
- Real-time DOC middleware
  - TAO ORB
  - Naming Service
  - A/V Streaming Service
  - AQoSA
- DVDViewer
- Simulated ATR

Heterogeneity

- Data formats - MPEG, PPM
- Mechanisms
  - RSVP, DiffServ
  - Filtering, scaling, compression
- Networking
  - Wired Ethernet
  - Wireless Ethernet

BBN TECHNOLOGIES
A Verizon Company

Adaptation Mechanisms for CPU and Network Overload

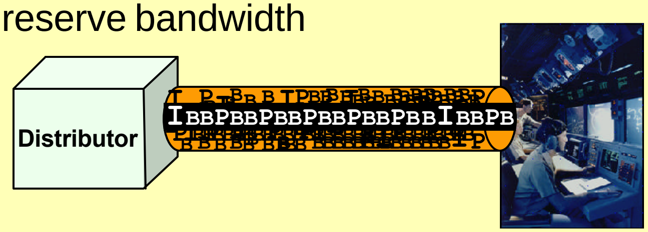
Mission requirements of UAV scenario

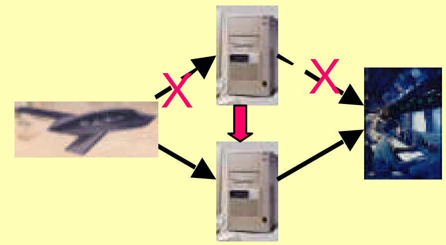
Timeliness
- **Maintain an out-of-the-window view of UAV imagery**

Importance
- **Frames must be dropped in reverse order of importance**

Fidelity
- **Highest fidelity frames must be delivered**

NETWORK RESERVATION
- Condition: Excessive Network load
- Action: Use IntServ and DiffServ to reserve bandwidth





LOAD BALANCING
- Condition: Excessive CPU load
- Action: Migrate distributor to a lightly loaded host

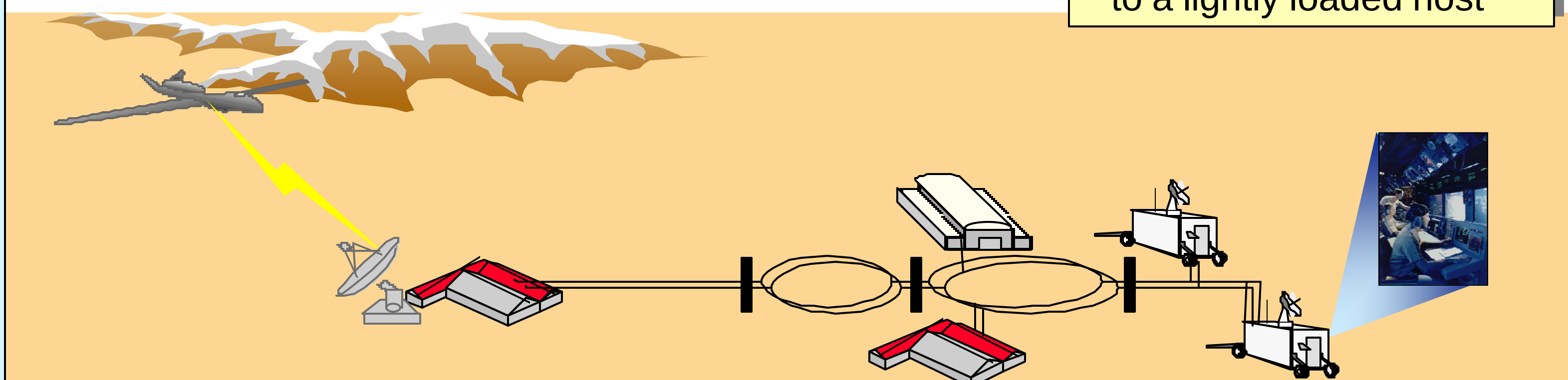
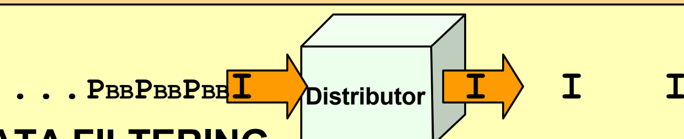


IMAGE MANIPULATION
- Condition: Excessive Network load
- Action: Scale image to smaller size

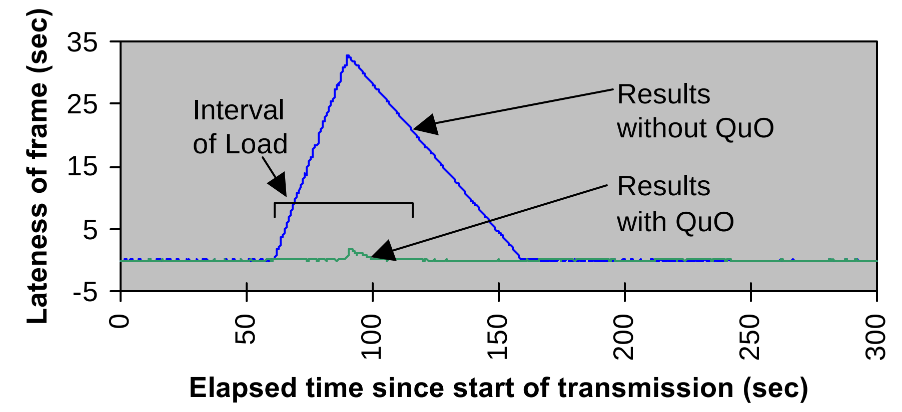


DATA FILTERING
- Condition: Excessive Network or CPU load
- Action: Drop selective frames

Experiment Metric – Latency Control

Experiment 1

- Sender, distributor, and receiver running on three Linux boxes, each with a 200 MHz processor and 128 MB of memory.
- 5 minutes (300 seconds) of video
- Introduce CPU load 60 seconds after start, remove after 60 more seconds
- Transport is TCP (reliable)



Adaptation	Delay (sec)	
	Mean	Maximum
None	5.391	32.696
Frame Filtering	0.067	1.930

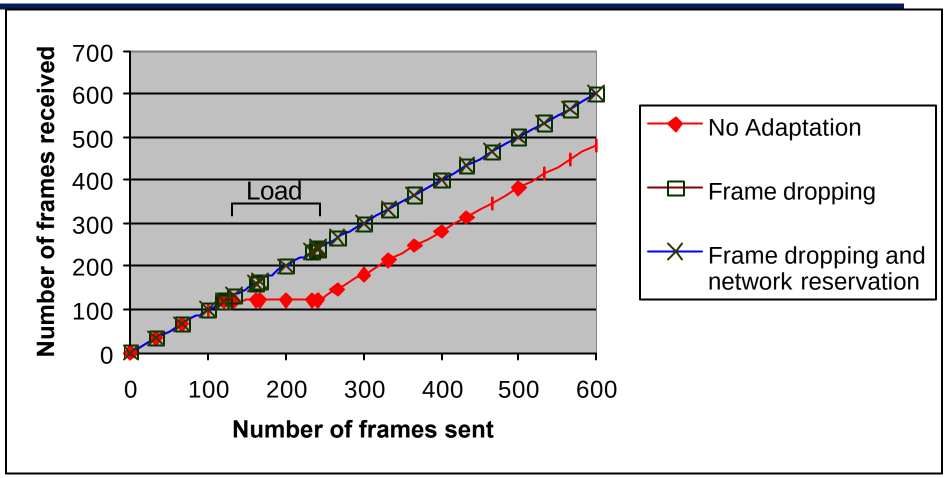
Benefit Metrics

- **Lower latency** in the presence of load
  - **Average 0.067 sec vs. 5.391 (80x imp.)**
  - **Worst case 1.930 sec vs. 32.696 (17x imp.)**
- **Control** over delivery of important data in the presence of load
  - **With no adaptation, delay was arbitrary**
  - **With adaptation, we chose to sacrifice less important frames to get better QoS for more important frames**

Experiment Metric – Control of Data Loss

Experiment 2

- Sender and distributor (933 MHz Pentium III, 512 MB RAM); receiver (200 MHz Pentium II, 144 MB RAM); 10 Mbps link; UDP
- 5 minutes (300 seconds) of video, with network load introduced after 60 seconds for 60 seconds (600 total I frames sent)
- Three runs
  - Control, no adaptation
  - Frame dropping adaptation only
  - Frame dropping and network reservation



Adap-tation	No. I frames lost	% getting through w/load	Avg. delay - no load (ms)	Avg. delay - load (ms)	Max. delay (ms)
None	119	1.65%	56.33	NMF	NMF
Frame Filter-ing	0	100%	57.01	122.15	143
FF + RSVP	0	100%	58.15	88.53	106

NMF – No meaningful figure. Most frames never arrived.

Benefit Metrics

- **Control** over loss of important data
  - **100% of important data arriving vs. 1.65%**
- Improved **performance** with adaptation combo
  - **FF+RSVP has 28% lower delay under load than FF alone (infinitely better than no adaptation)**

Applicability Metrics

- **Low overhead** of QuO adaptation
  - **Extra avg delay: 1.2% (FF), 3.2% (FF+RSVP)**
  - **Std. Dev: 5.19 (none), 5.25 (FF), 4.60 (FF+RSVP)**

BBN TECHNOLOGIES
A Verizon Company

Experiment Metric – Graceful Degradation

Experiment Motivation

- Full network resources will frequently not be available to applications
  - **Simply not enough to support full video**
  - **Contention with other video sources**
- Applications need to be able to work with degraded resources

Experiment

- Sender, distributor, and receiver on 750 MHz Pentium III with 512 MB RAM; 10 Mbps link
- 5 minutes (300 seconds) of video, with network load introduced after 60 seconds for 60 seconds (600 total I frames sent)
- Partial reservation, frame filtering alone, and in combination

Adap-tation	No. I frames lost	% getting through w/load	Avg. delay* (ms)	Std. Dev.*
FF only	6	95.04%	93.26	110.28
Partial Resv Only	69	43.90%	118.54	217.56
FF + Partial Resv	1	99.18%	76.83	84.81

*Lost frames not included in delay and std. dev. figures

Benefit Metrics

- Combination has **lower data loss**
  - 17% of the data loss of FF; 1.4% of Partial Resv.
- Combination has **lower average latency**
  - 17.6% lower than FF; 35.2% lower than Part Resv.
- Combination has **lower standard deviation**
- **Scale:** Can support 5+ partial reservations in the bandwidth of one full reservation

Outline

- A Point of View & Background
- Technologies for Managed Behavior in Rapidly Changing Environments
- Examples we’ve built, tested and evaluated
  – WSOA, UAV
- Some Lessons Learned and Challenges Going Forward

Lessons Learned and Open Research Issues

- High Performance also means working under dynamically changing requirements and unanticipated conditions
- It is feasible to operate with less than a full complement of resources, so long as they are targeted at the critical parts
- There is a context sensitive nature to "what's the best behavior"
- Late binding is an avenue to many innovative approaches
- Layered solutions with integrated parts are an important development strategy, especially for large, complex problems. This involves information sharing and cooperative behavior across and between these layers
- ***Blending Reliability, Trust, Validation, and Certifiability without sacrificing effective real time performance***