BBN TECHNOLOGIES

Evolution of Middleware Services toward Realtime and Embedded (Cyber-Physical) Environments: The BBN experience

Dr. Rick Schantz

February 5, 2009

Outline

- Introduction: Who am I? Why am I here?
- Some Background and Observations on Middleware and Network-centric Applications Perspectives

- Multiple Points of View on QoS Management from Recent Activities
  - Realtime properties*
  - Cyber-defense*
  - Certification

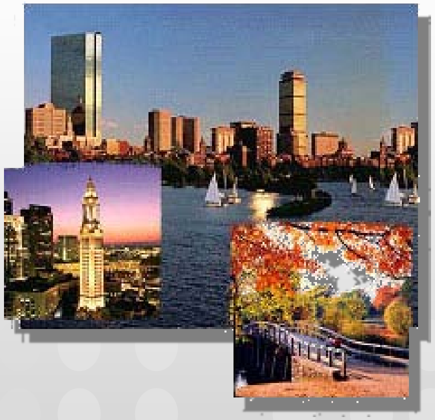
(noting the repeatable R&D cycles of invent/develop, real-world evaluations, transitions)

- Looking Forward: Some Conclusions
- Q/A

Who is BBN Technologies

- *An advanced technology research and development firm, specializing in Information, Computer, & Physical Sciences*

- *Known for technical excellence and challenging conventions to provide new and fundamentally better solutions to complex technical problems*
- *Providing effective, real-world solutions and satisfying our customers and have been key to our success for over 50 years*
- *Our staff consists of ~ 700 professionals*
  - *2/3 with advanced degrees and security clearances*



- *We maintain principal offices in Cambridge, MA and the Washington, DC area*

All of our offices can support classified work

History of Innovation

BBN TECHNOLOGIES

1950s

Acoustic Design for UN General Assembly Hall

AI Program for Pattern Recognition

1960s

Demonstration of Time Sharing

LOGO Programming Language

ARPANET-First Multi-node Packet Switched Network

1970s

1st Person-to-Person Network Email

@ Sign for Email Addresses

Acoustic analysis of JFK Assassination Tapes

Analysis of Nixon Watergate Tapes

First ARPANET Distributed Operating System

First Symmetric Multi-processor

First TCP for UNIX

1980s

First Electronic Mail

Defense Data Network

National Science Foundation Network (NSFNET)

Natural Language Computer Interface

CRONUS Distributed Object Computing Environment

Distributed Interactive Simulation (SimNet)

Collaboration Planning Technology

1990s

Secure email for DoD

Multi-Gigabit Router

Information Assurance

Broadband Wireless Technology

Genetic Algorithm Scheduling Tools

Collaborative Planning for Desert Storm

ATM Switch

40K Word Speech Recognition System

Quality of Service for Objects Middleware

Safekeyper Certificate Management

2000s

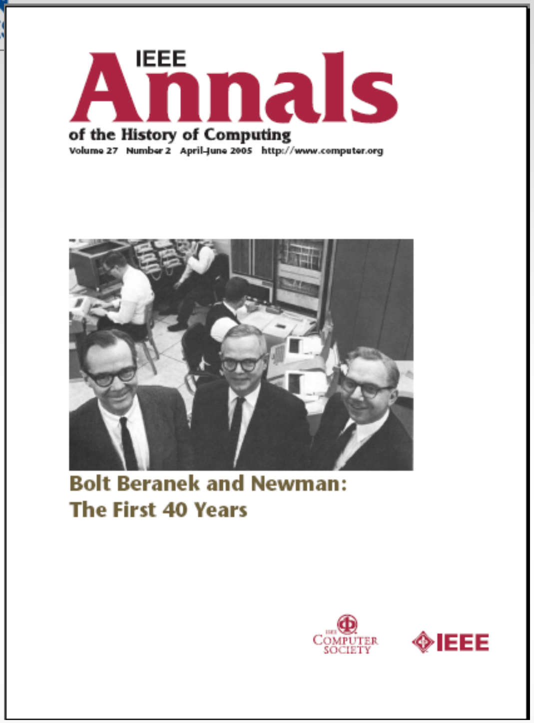
Call Director Natural Language Routing

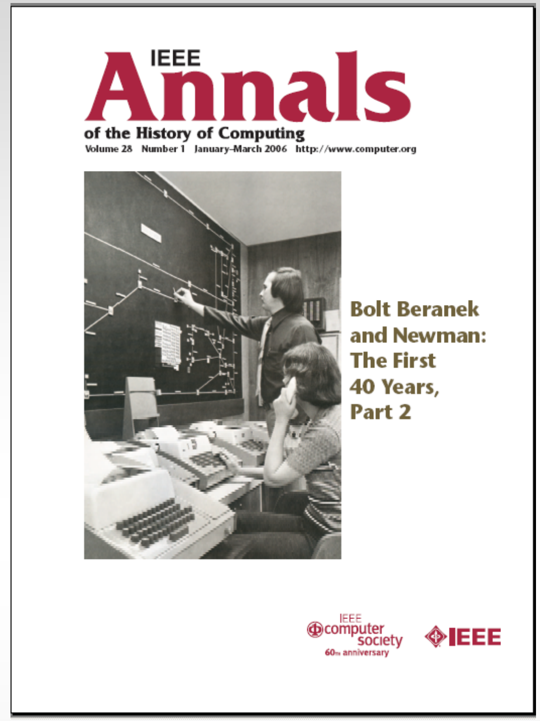
DARPA Agent Markup Language

Microthunder Urban Environment Surveillance System

Quantum Cryptographic Network

...

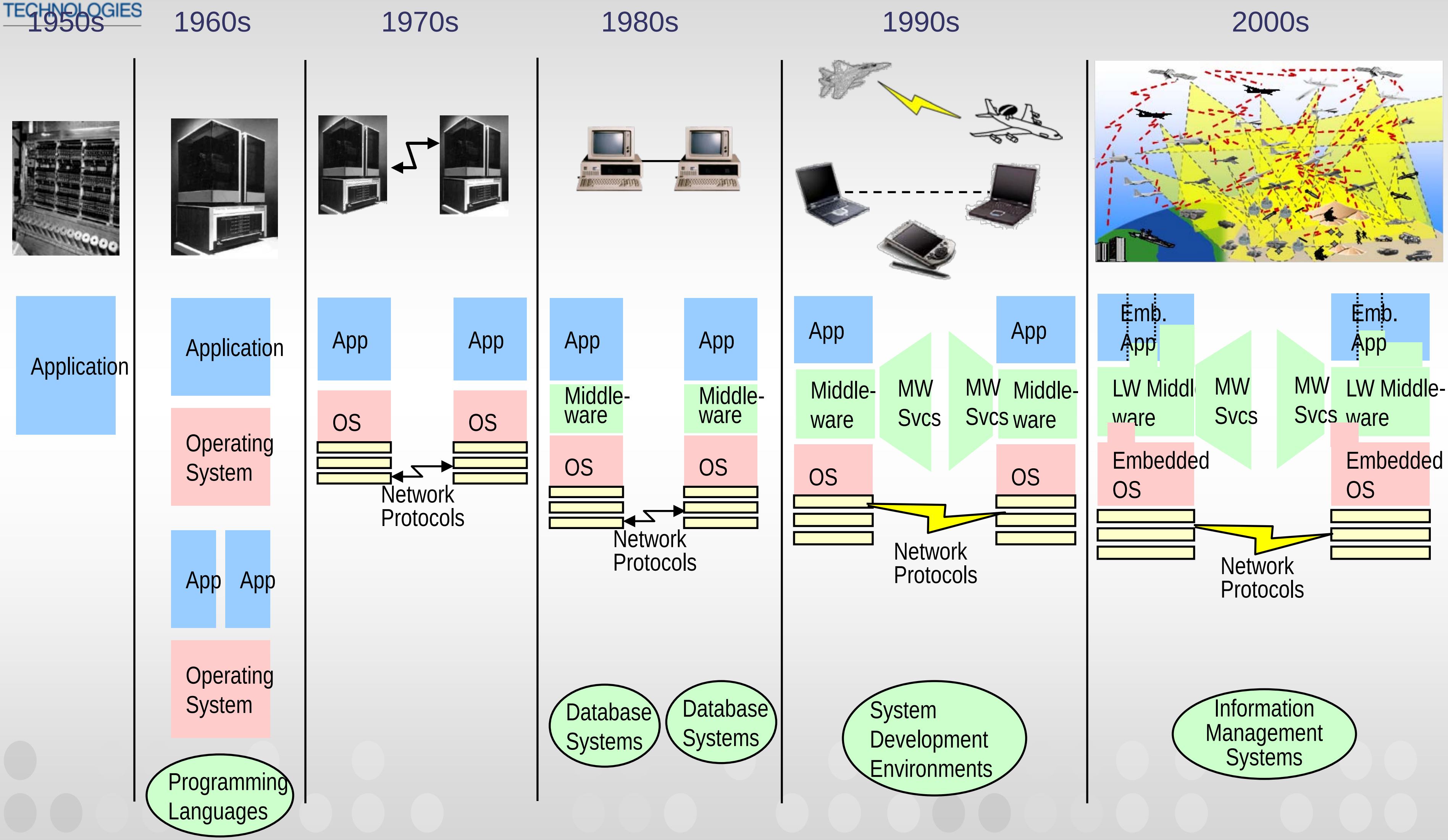




http://www2.computer.org/portal/web/csdl/doi/10.1109/MAHC.2005.23

http://www2.computer.org/portal/web/csdl/doi/10.1109/MAHC.2006.6

Historical Context: Software Infrastructure Enables Application Capabilities

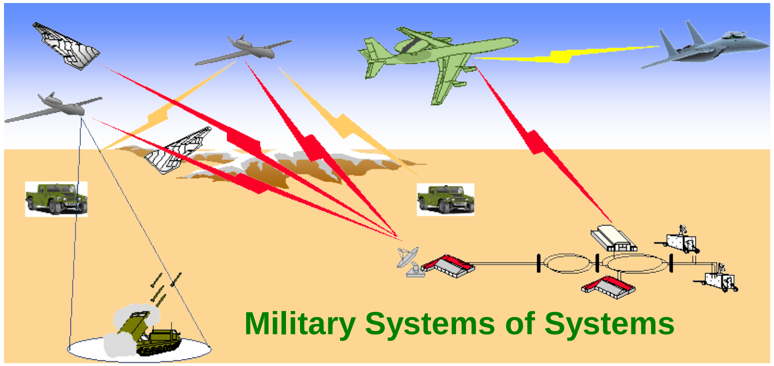


1950s Fifty Years of Distributed Systems Software Architecture Evolution 2008+

Distributed Real-time Embedded (DRE) Systems Context

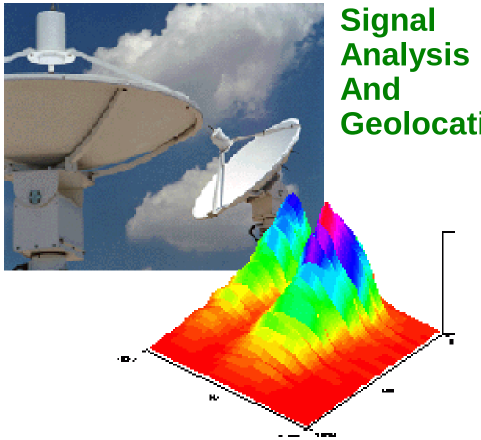
- Applications are distributed and network centric
- Stringent QoS requirements, including predictable and efficient data transfer and control
- Resources are constrained and shared
- Operate in dynamic environments

Avionics Mission Planning



Military Systems of Systems

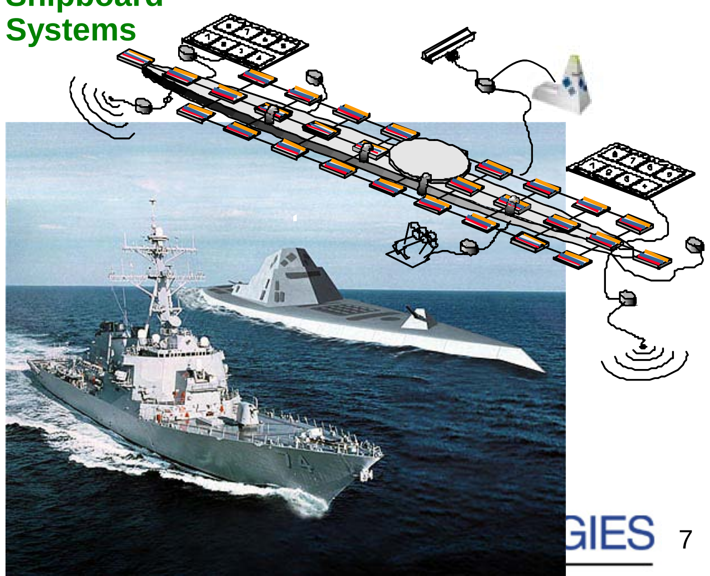
Industrial Production



Signal Analysis And Geolocation

Disaster Response Systems

Shipboard Systems



Background: Underlying Forces at Work

BBN TECHNOLOGIES

- Everything is a computer
- Everything is a networked computer
- Everything is potentially interdependent
- Things connect to the real physical world
- Increasing heterogeneity, distance and mobility

Leading to Current Trends and Directions

- Need for Integrated/Managed End-to-End Behavior
  - Multi-dimensional QoS
- Multi-Layered Architectures, Network-centric Services Oriented & Systems of Systems
  - Coordinated and provided thru advanced Middleware solutions
- Evolutionary Designs Over Varying and Changing Configurations
  - Static → Dynamic; Adaptive
- (More) Advanced Software Engineering and Open Standards
  - (trying to keep pace)

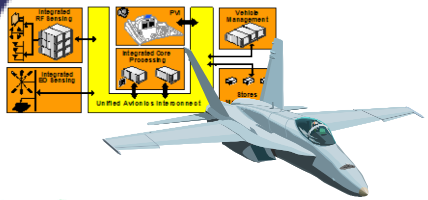
Outline

- Introduction: Who am I? Why am I here?
- Some Background and Observations on Middleware and Network-centric Applications Perspectives
- Multiple Points of View on QoS Management from Recent Activities
  - Realtime properties*
  - Cyber-defense*
  - Certification

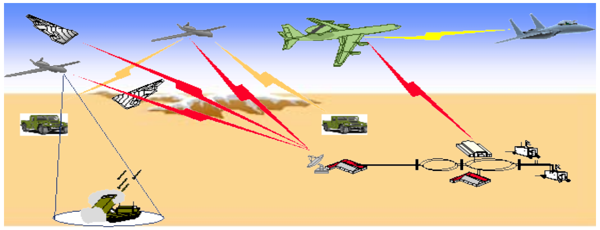
(noting the repeatable R&D cycles of invent/develop, real-world evaluations, transitions)

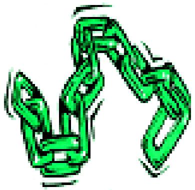
- Looking Forward: Some Conclusions
- Q/A

Need for QoS Adaptive Systems, Applications, Middleware & Networks



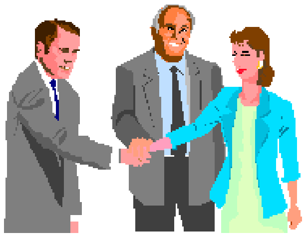
Static QoS provisioning is the rule in embedded systems, but ***dynamic QoS*** is the need





Real End-to-end QoS is important
- QoS provisioning at a location/component (e.g., node, network) is necessary, but not sufficient
- Ultimate consumer of information determines the QoS requirements, even if source is remote
- End-to-end QoS is only as good as what can be provided thru each bottleneck at every particular point in time (over-provisioning often wears out with time)

Necessary to **specify, measure, control, adapt & mediate** QoS (at design time, (re)configuration time, & run time)

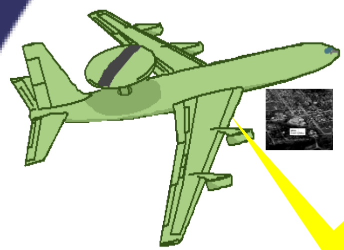


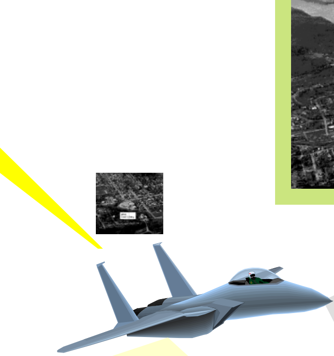
- The QoS desires of multiple applications might not be able to be satisfied with available resources
- QoS policies will often conflict, e.g., security and real-time performance
- Conditions, mission modes, and objectives will change

Need an adaptive middleware framework at the seams you can grow with to support QoS enabled solutions for DRE systems
- Separate QoS concerns from functional concerns
- Avoid the programming of point solutions and further entangled applications
- Avoid premature tradeoff binding, promoting change and assembly
- Anticipate evolution and more expansive integration and change
- Scalability anticipating success

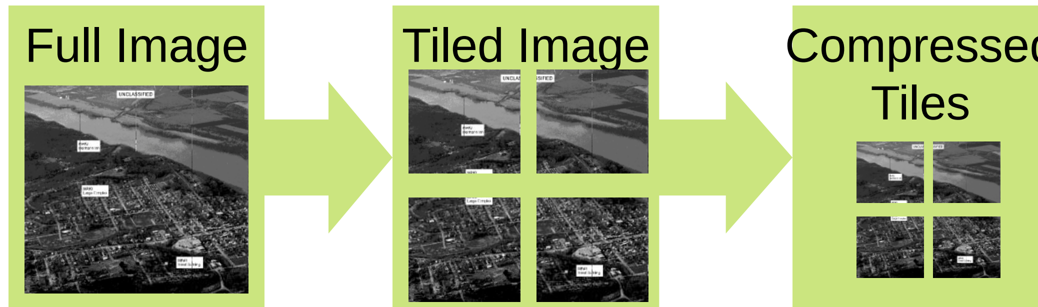
Example DRE Application 1

Avionics Dynamic Mission Planning



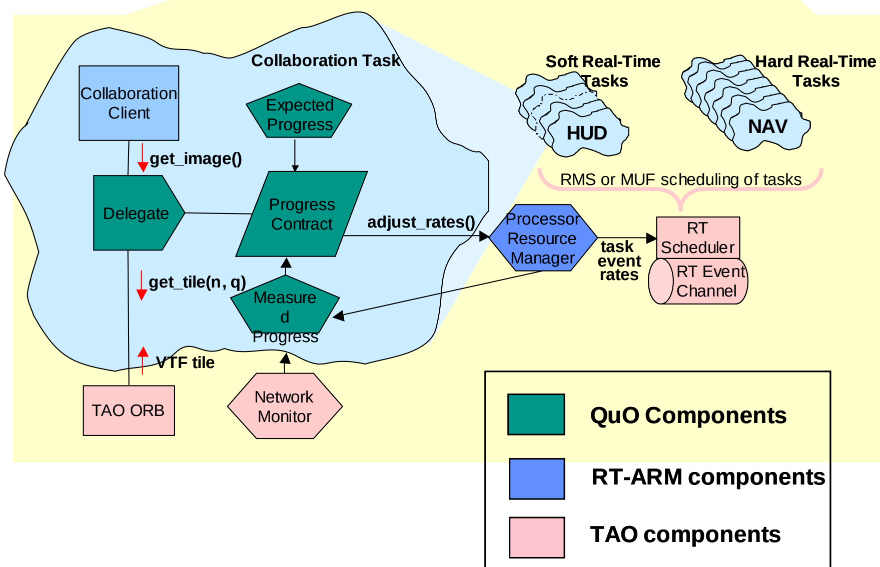


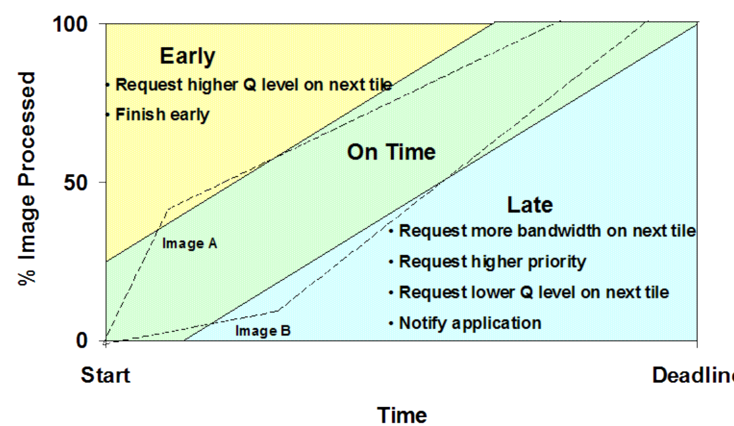
A Net-meeting like mission replanning collaboration between C2 and fighter aircraft



QoS Techniques

- Tiling
- Compression
- Processor Resource Management
- Network Resource Management

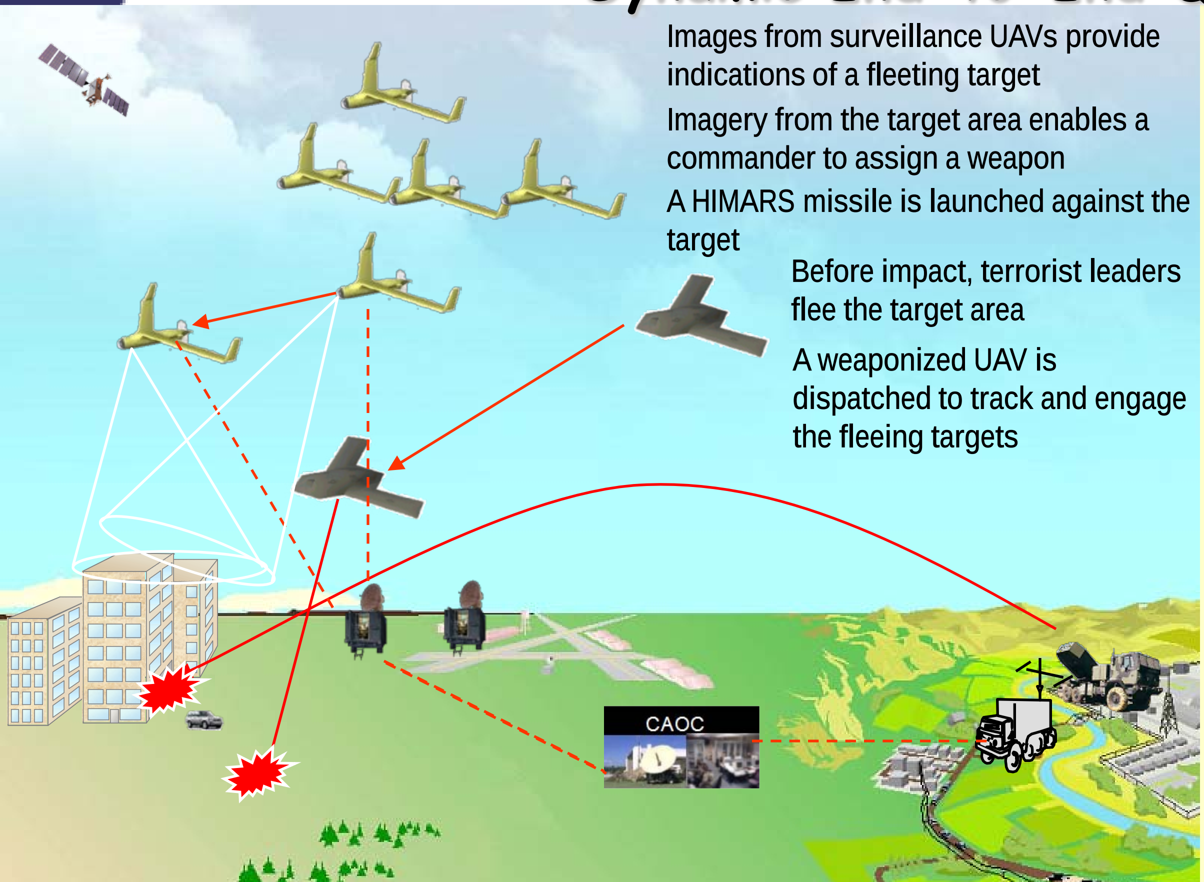




BBN TECHNOLOGIES

Example DRE Application 2

Multi-UAV Surveillance and Target Tracking Requires Dynamic End-to-End QoS Management



End-to-End Mission-Driven QoS Management

Surveillance

- Maximize surveillance area
- Sufficient resolution in delivered imagery to determine items of interest

Target Acquisition and Engagement

- UAV observing target provides high resolution imagery so that target or threat identification is possible

Battle Damage Assessment

- UCAV must provide high resolution imagery until a human operator has determined that it is sufficient
- UAV over target area must continue to provide target acquisition and engagement mission

Heterogeneous, shared, and constrained resources

Multi-layer points of view: System-view, mission-view, application-string view, local resource view

Mission-defined requirements and tradeoffs (e.g., rate, image size, fidelity)

Changing modes, participants, and environmental conditions

The challenge is to program the dynamic control and adaptation to manage and enforce end-to-end QoS

Demonstration Imagery Displays (C2 Receivers)

Name, role and COI of the asset

Color of the border reflects the role of the SimUAV

Image size and rate are a result of QoS information management

QoS Policies

Mission	Relative Priorities
ISR_COI	1
TST_COI	2

Qos Constraints and Tradeoffs								
Roles	Relative Priority	Resource Needed			Quality of Information Needed			
		BW Needed (kbps) (Min-Max)	DiffServ Codepoint	CPU (Receiver) (%)	Rate (Timeliness) IO/Frame Rate	Scaling (Size)	Compression (Accuracy)	Cropping (Precision)
SURVEILLANCE (ISR)	1	50-200	Best Effort	0.1-2.0	0.1 – 0.4	Qtr-Qtr	JPG-JPG	None
TARGET TRACKING (TT)	6	150-600	Expedited Forwarding	1.5-5.5	1-1.5	Half-Half	None - JPG	None
BATTLE DAMAGE ASSESSMENT (BDA)	4	300-400	Assured Forwarding	1.5-3.0	0.25-0.5	Full-Full	None-JPG	None-30%

More Detail

QoS Instrumentation: Policies and Sensors

Asset Identification Information

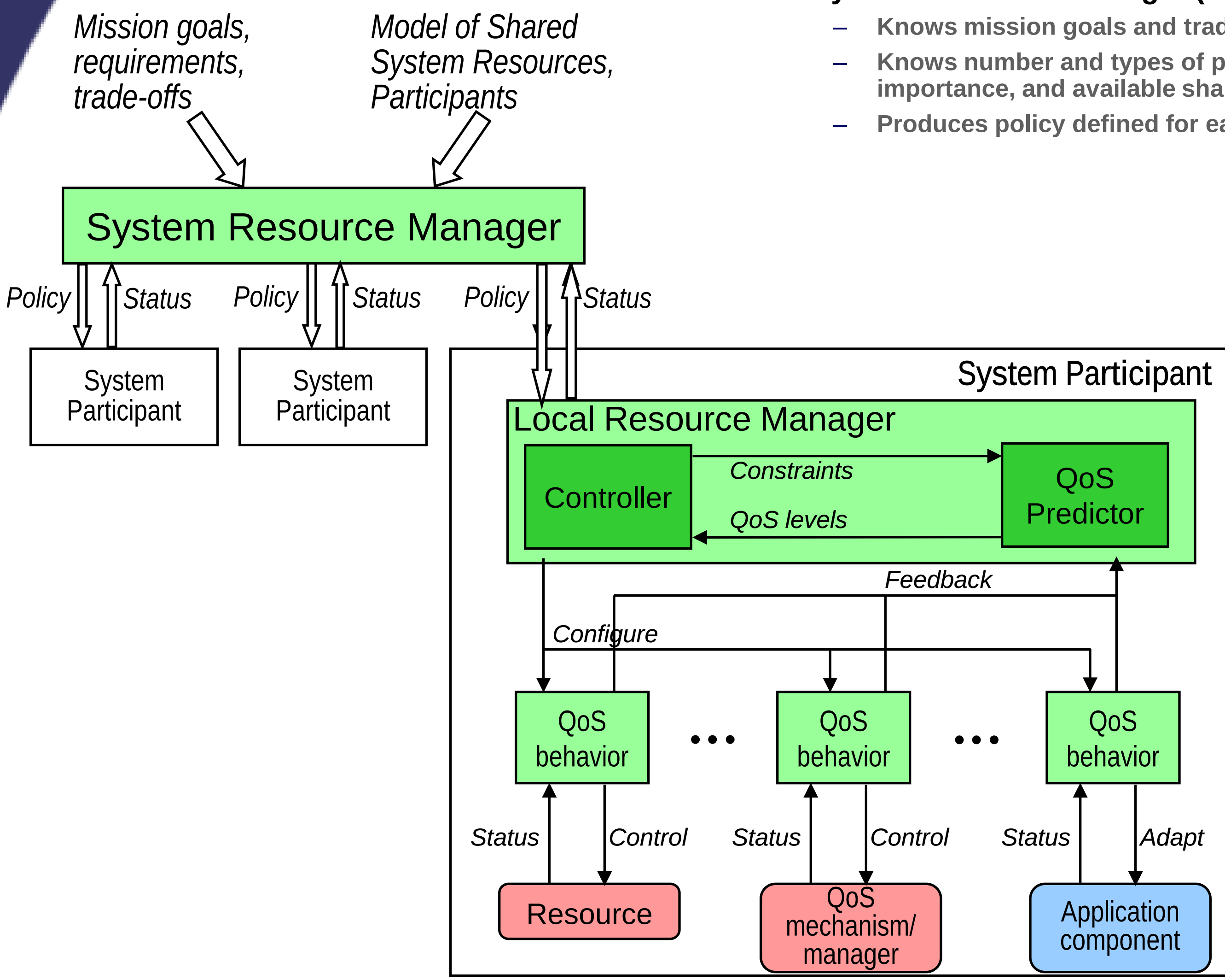
Resource Information: Allocation vs Usage

QoS Policy Information

Latencies in the IMS

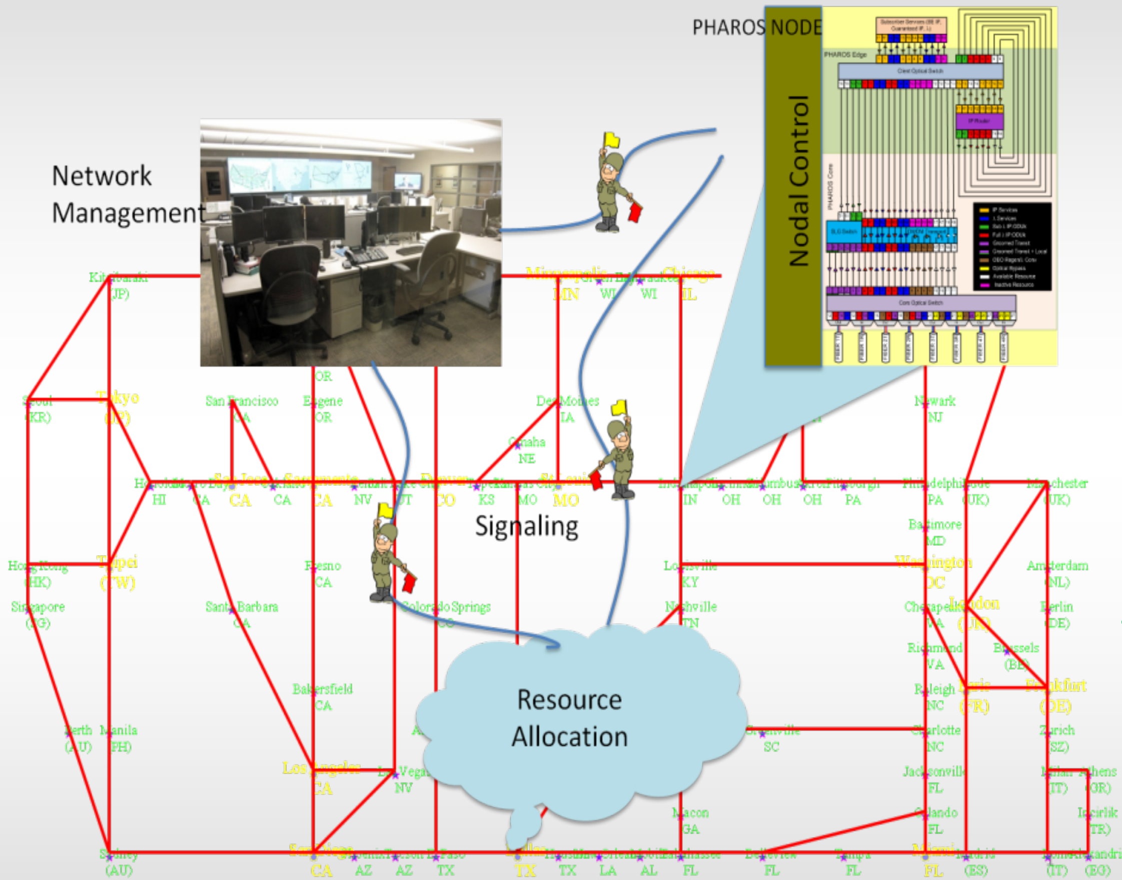
BBN TECHNOLOGIES

Multi-Layer QoS Management Architecture



- **System Resource Manager (SRM) near C2 node**
  - Knows mission goals and tradeoffs
  - Knows number and types of participants, roles and relative importance, and available shared resources
  - Produces policy defined for each participant
- **Local Resource Manager (LRM)**
  - Determines how to utilize allocated resources to meet mission goals
  - Configures and monitors QoS behaviors
- **QoS behaviors**
  - Control and monitor individual resources or mechanisms, or adapt application behavior

Changing Building Blocks: An Example



In the PHAROS project we are developing the global backbone network of the future

- Optical– guaranteed IP services with high data rate (up to 10Gb/s), low latency (125ms global one way), low jitter (25ms global one way)
  -- even more aggressive for non-IP (i.e., wavelength services)
- Agile– fast service set up: sub-second provisioning and re-provisioning (post-failure) instead of truck-roll
- Dependable—guaranteed bandwidth services are protected against up to 3 network failures
- Efficient– Resources allocated for protected paths are globally optimized

Robust and resilient against partitioning, DoS and other network attacks

- Separation of data and control plane, differentiated control channels, no interpretation of data, authentication of service requests, strict ingress monitoring
- Dynamic redundancy and cross-checking in control and management protocols

Outline

- Introduction: Who am I? Why am I here?
- Some Background and Observations on Middleware and Network-centric Applications Perspectives
- Multiple Points of View on QoS Management from Recent Activities
  - Realtime properties*
  - Cyber-defense*
  - Certification

(noting the repeatable R&D cycles of invent/develop, real-world evaluations, transitions)

- Looking Forward: Some Conclusions
- Q/A

Generations of Security Research

BBN TECHNOLOGIES

No system is perfectly secure– only adequately secured with respect to the perceived threat.

Prevent Intrusions
(Access Controls, Cryptography, Trusted Computing Base)

Trusted Computing Base


Access Control & Physical Security


Cryptography

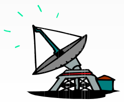
1st Generation: Protection

But intrusions will occur

Detect Intrusions, Limit Damage
(Firewalls, Intrusion Detection Systems, Virtual Private Networks, PKI)


Firewalls


Boundary Controllers


Intrusion Detection Systems


VPNs


PKI

2nd Generation: Detection

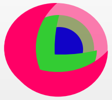
But some attacks will succeed

Tolerate Attacks
(Redundancy, Diversity, Deception, Wrappers, Proof-Carrying Code, Proactive Secret Sharing)


Intrusion Tolerance


Big Board View of Attacks
Real-Time Situation Awareness
& Response


Graceful Degradation


Hardened Operating System

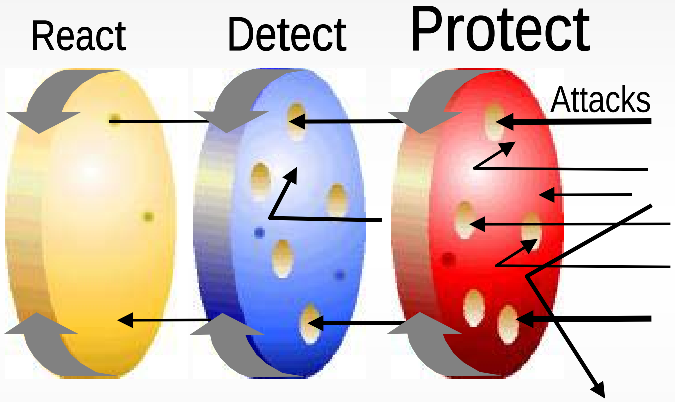
3rd Generation: Survivability/Tolerance

Survivability

BBN TECHNOLOGIES

Premise

- The number & sophistication of cyber attacks is increasing – some of these attacks will succeed



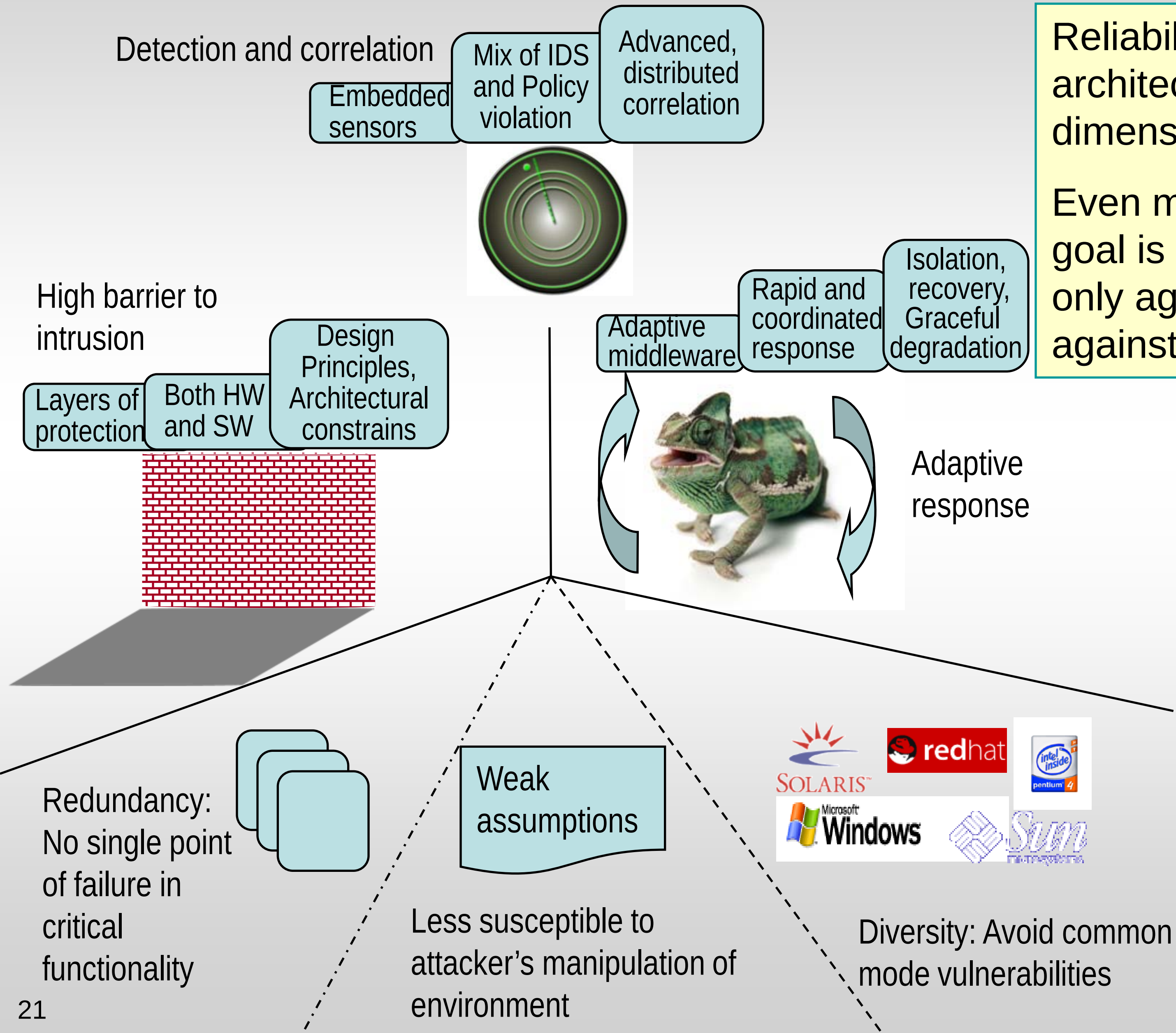
Philosophy

- *Operate through attacks* by using a layered defense-in-depth concept
  - Accept some degradation
  - Protect most valuable assets
  - Move faster than the intruder

Approach

- "Defense Enabling" Distributed Applications
- Based on Adaptive Middleware Technology

Architecting Survivability into Large Systems With Realtime Response



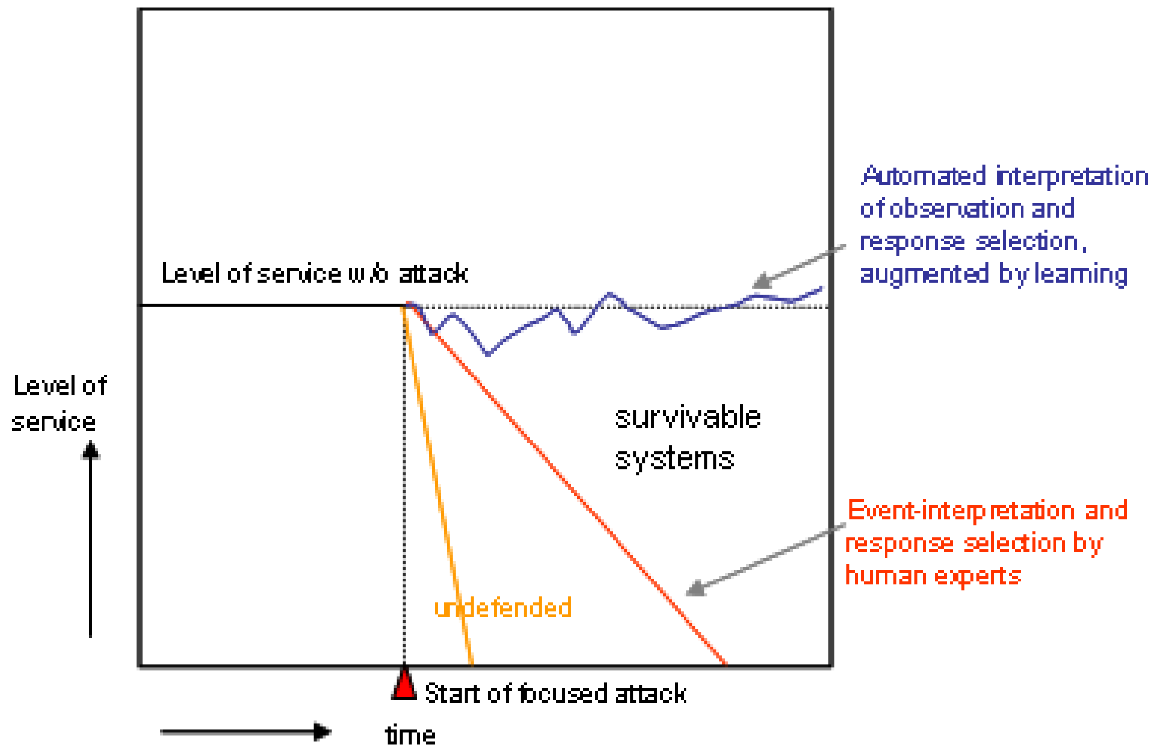
Reliability requires architecting in multiple dimensions

Even more so, when the goal is to be resilient not only against errors, but also against attacks….

General principles for survivability

- Protect as best as possible
- Improve chances of detection
- Adapt to manage gaps

3rd Generation …



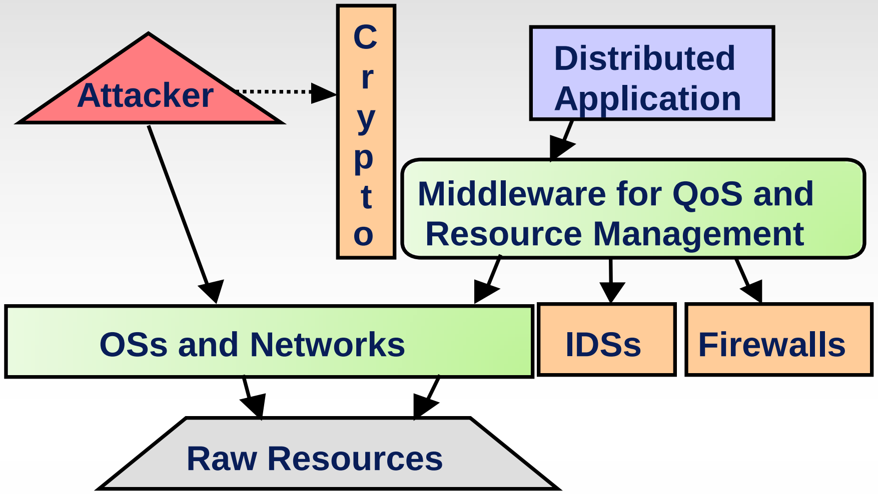
Tolerance and Survivability:

- Assumes that attacks/bad things cannot be totally prevented– some attacks will even succeed, and may not even be detected on time..

- Focuses on desired qualities or attributes that need to be preserved and continued even if in a degraded manner—
  - availability: (of information and service)
  - integrity: (of information and service)
  - confidentiality: (of information)
- Exploring beyond degradation-- regain, recoup, regroup and even improve

- Semi-automated: Survivability architecture captures a lot of low level (and sometimes uncertain and incomplete) information – utilizes advanced reasoning and machine learning

Applications that Participate in their Own Defense (APOD)



APOD Approach: Use middleware to interface with defense mechanisms and integrate defense strategies

Circa: 1999

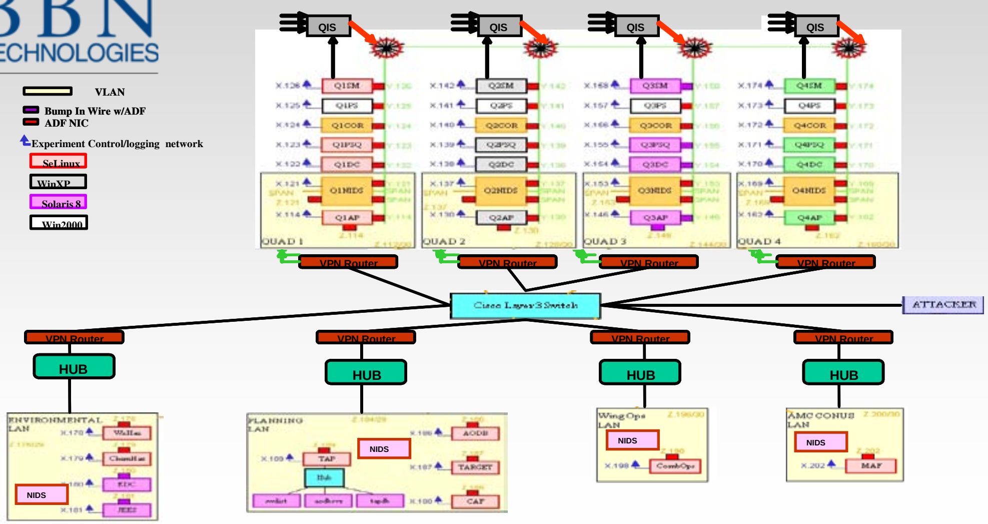
Observations:

- Distributed applications need distributed resources
- There are enablers (middleware, OS, Networks,) as well as defense mechanisms
- But not much coordination between applications and defenses
- Challenge: develop technology to defense-enable applications

Lessons Learned:

- Application's involvement in defense is an important attribute
- Possible to build more survivable application from less secure components running in a less secure environment
- Distributed middleware can be used to coordinate defenses from application's point of view **as long as corrupt application cannot control the defenses (self protection, sophisticated attacks, …)**

DPASA



Circa: 2003-2005

Observations:

- Lots of point solutions (firewalls, access controls, IDSs, replication..), need an architecture to organize
- Time to loss of service under attack is in minutes for state of the art defended systems
- Challenge: Defense enable a military information system that can survive sophisticated attackers for 12 hrs

DPASA Approach: combine elements of protection, detection and adaptive reaction in the survivability architecture

Lessons Learned:

•Survived 75% of attacks, even when the attacker was given insider access and privilege (red team would actually start the system, after placing attack code)

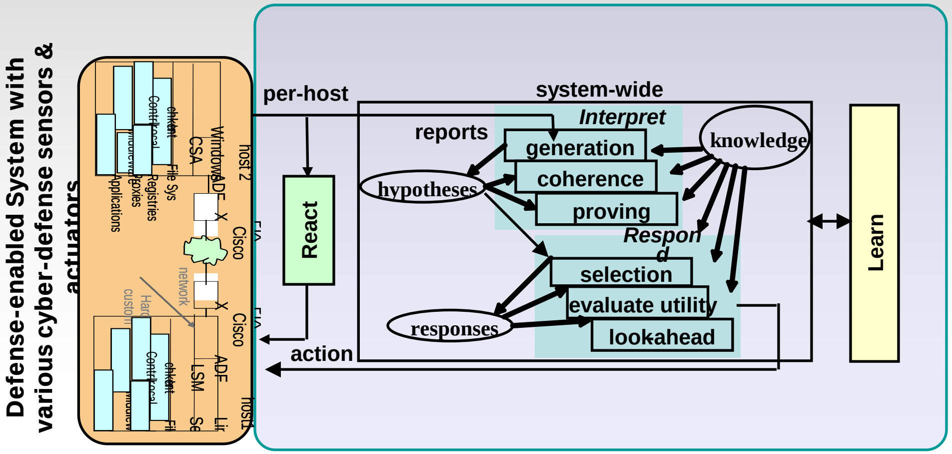
- A number survivability design principles that goes beyond “defense in depth”
  - SPOF elimination, redundancy and diversity, containment, hardware or cryptographic root of trust, Crumple zones (many of these show up in recently published SANS/MITRE/NSA Common Weakness Enumeration (CWE))
- Availability was the only attribute that was successfully compromised
- Flaws in COTS components still a/the major risk (and a fact of life)
- Limiting attacker probing and adding uncertainty helped enormously
- Information reported by defended system can cause information overload– needed experts to interpret
- What will happen if information system spans multiple domains?– need to explore cross domain issues

CSISM

Circa: 2006-2008

Observations:

- Possible to architect a highly survivable system, but the system provides a heavy stream of signals that only experts can interpret
- Involvement of human experts at this level is costly and often impractical
- Challenge: Develop automated mechanisms that would interpret the reports and help decide effective course of action

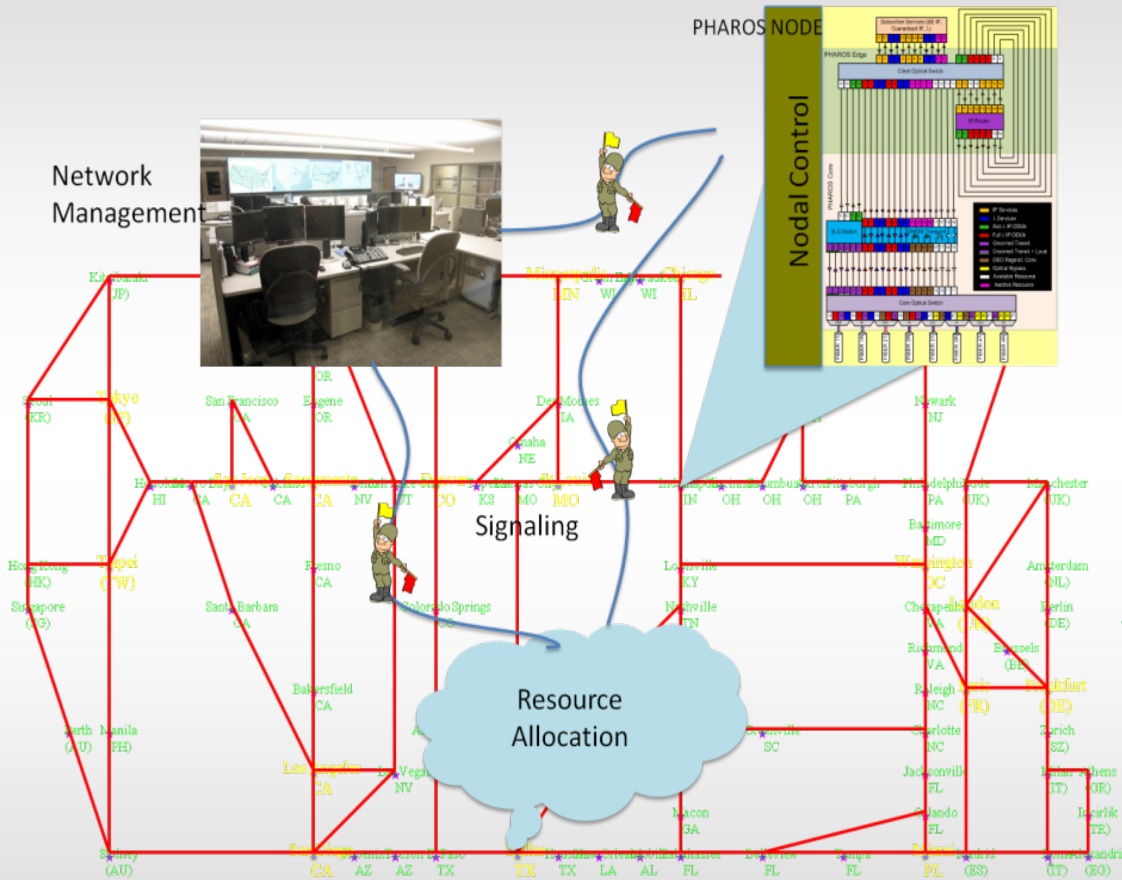


CSISM Approach: 3 level decision making- reactive, deliberate and learned; use theorem proving and coherence to reason about accusatory and evidentiary information contained in reported events

Lessons Learned:

- Possible to minimize on-line involvement of human experts if appropriate knowledge about the system, its defenses, attacker objectives etc are encoded into the reasoning mechanism
- Event interpretation by reasoning about the evidentiary and accusatory information using theorem proving and coherence search is viable, but compute intensive– in red team experiments CSISM were able to decide correctly in 75% cases
- Integrating learned responses on line needs additional research, but off line use of machine learning was useful if good training data is available

Changing Building Blocks: An Example



In the PHAROS project we are developing the global backbone network of the future

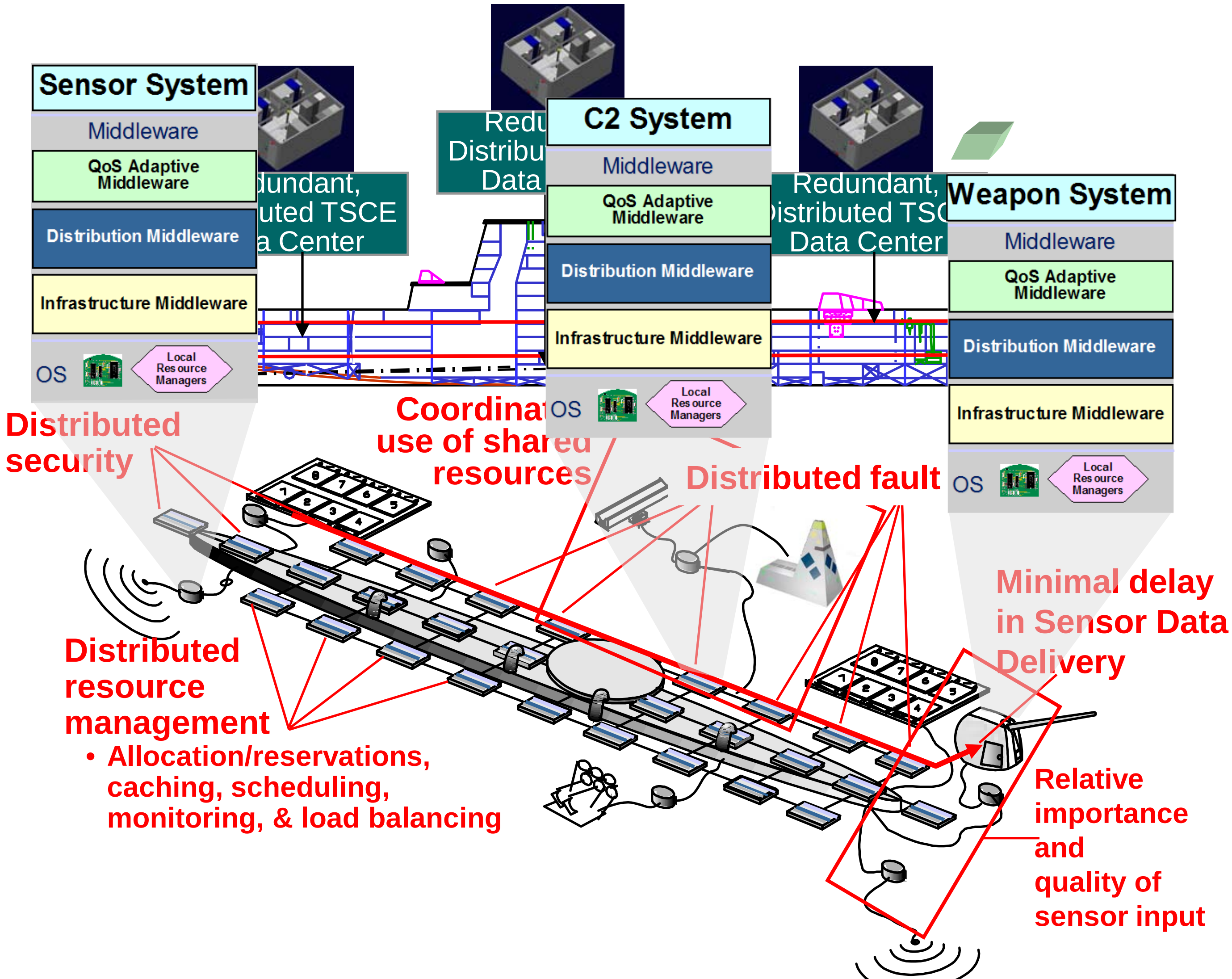
- Optical– guaranteed IP services with high data rate (up to 10Gb/s), low latency (125ms global one way), low jitter (25ms global one way)
  -- even more aggressive for non-IP (i.e., wavelength services)
- Agile– fast service set up: sub-second provisioning and re-provisioning (post-failure) instead of truck-roll
- Dependable—guaranteed bandwidth services are protected against up to 3 network failures
- Efficient– Resources allocated for protected paths are globally optimized

Robust and resilient against partitioning, DoS and other network attacks

- Separation of data and control plane, differentiated control channels, no interpretation of data, authentication of service requests, strict ingress monitoring
- Dynamic redundancy and cross-checking in control and management protocols

Applying Middleware Concepts to the Total Ship Computing Environment

- Total ship computing concept
- Redundant distributed computing bays
- Multiple QoS properties and requirements
- Layers of middleware for software infrastructure
- Multi-layered policy and control



Increased Naval Warfighting Power through Shipboard Resource Management

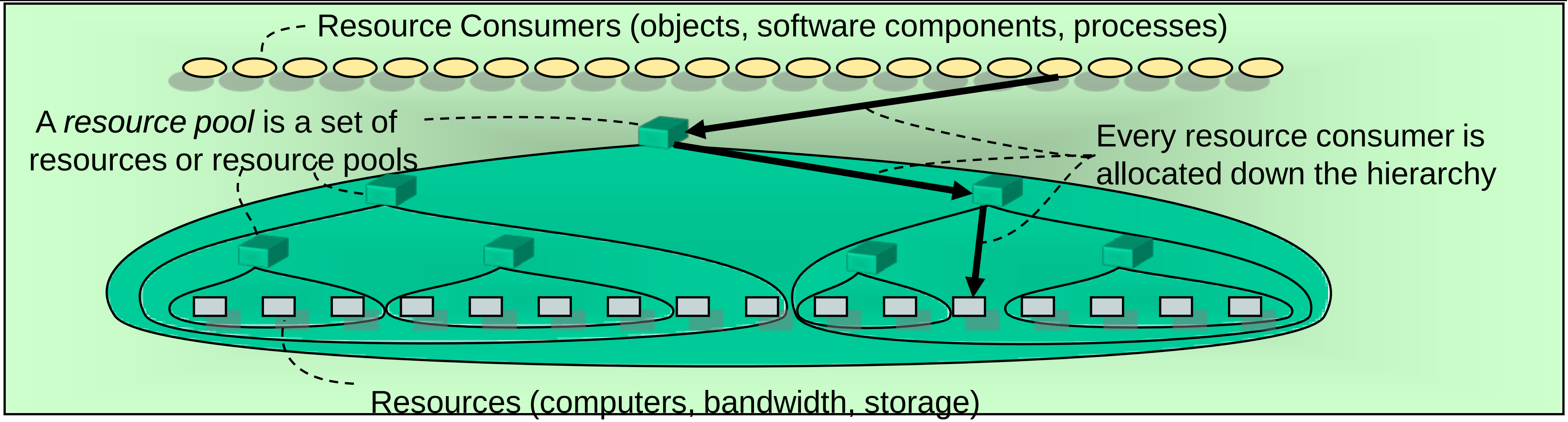
Problem:

Manage ship computing **resources to maximize warfighting capabilities**, in response to both changes in the tactical situation and damage.

Solution:

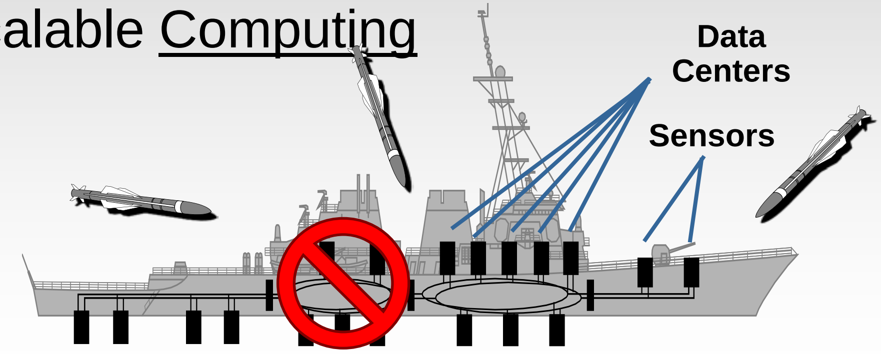
Dynamic resource management:

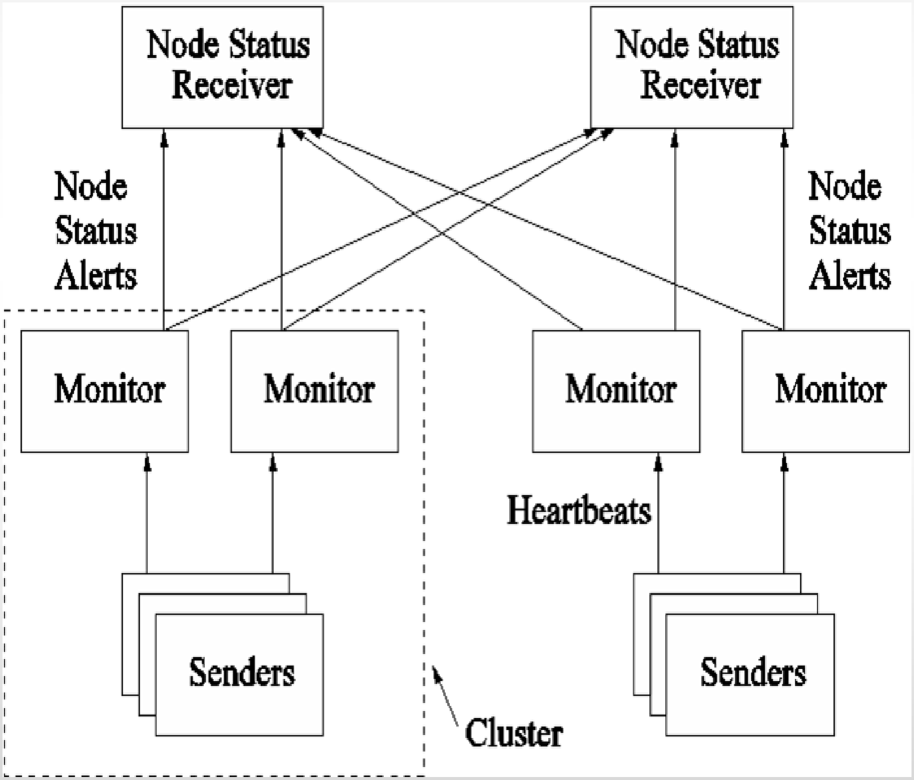
- Monitor health of system components
- Monitor performance thresholds end-to-end
- Maintain copies of component states
- Respond to actual and anticipated limit violations by re-assigning functional threads



High-Assurance, Fast Node Failure Detection

Need a distributed computing environment that can rapidly respond to changing operating conditions.
High speed, decentralized, scalable <u>Computing Node Failure Detection</u>





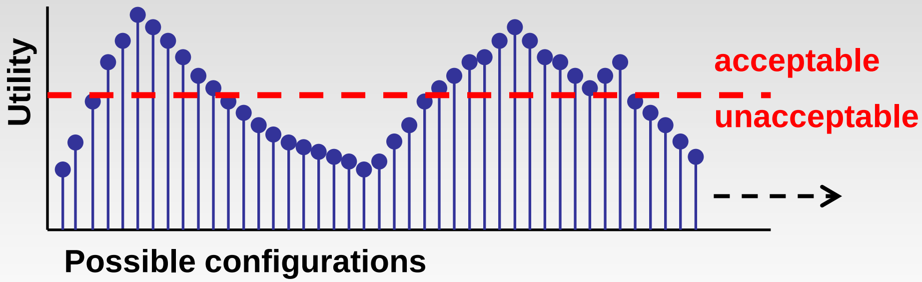
We have a hierarchical failure detection architecture

Challenge: adjust dynamically to changing failure detection performance

4 Dimensions of Dynamic High-Assurance Behavior:

- Low False Positive Rate
- Fast Worst-Case Detection Time
- Low Overhead
- Scalability to thousands of nodes

BBN TECHNOLOGIES



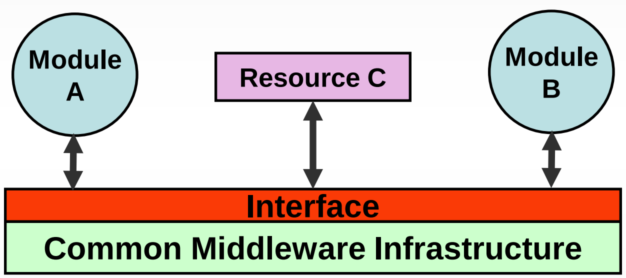
1.Identify “Good” and “Bad” Behavior: *Utility Metric*

Important considerations when defining a utility metric for configurations:

1.System Safety: Does the system satisfy safety constraints?

2.Fault Tolerance: Is system flexible for unforeseen eventualities?

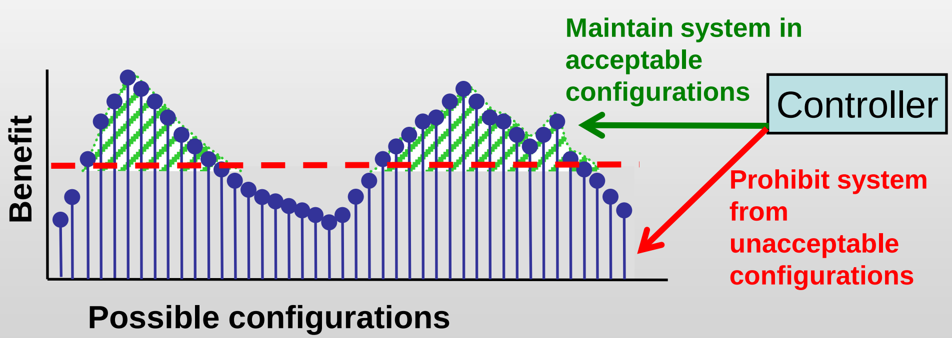
3.Computability: Can the metric be computed in real-time?



2. Component Interaction Control

We can more uniformly and certifiably *control* the resources provided through these interfaces if we provision the resource management functionality as a common middleware infrastructure.

Important considerations include the provisioning of communication, computation resources to operate middleware, scalability



3. Restrict Operation to Certifiable Configurations

Through the use of common middleware infrastructure and utility metric, we want to permit “certifiable” behavior to occur and prevent the system from entering into an “unacceptable” configurations.

Important considerations:

1. Difficult to predict the effects of control operations in real-time.
2. May need to maintain a list of “fail-safe” default configurations.

Looking Forward: Some Interim Conclusions

BBN TECHNOLOGIES

1. Heterogeneity/diversity is your friend, but is still costly
   - On the one hand we often preach it; but in practice we avoid it
   - Extensible, open standards is key to avoid (premature) lockdown
2. We routinely build predictively behaving systems, and we routinely build interoperable systems, but we do not (yet) routinely build predictable interoperable systems
   - Interoperability → sharing
   - Predictability → isolation and dedicated resources
3. Many of the distributed, realtime, embedded environments we engage (will) have certifiability requirements
   - Current approach is completely static and exhaustive testing
   - Interconnection drives dynamic behavior which breaks current approaches
4. We're in the midst of a long march forward, and the "middle" is where a lot of the important new action lies
   - Adopting an evolvable, common interconnection substrate is key and raises all boats
   - Technology provides the means for blurring common boundaries; systems provide the means (and challenges!) for orchestrating more cohesive and enduring operation

Additional Information

Thanks for Listening!

Comments/Questions to Rick Schantz

schantz@bbn.com