

MULTI-LAYER, MISSION-AWARE QoS MANAGEMENT TECHNIQUES for IP APPLICATIONS in a JOINT BATTLESPACE INFOSPHERE

Peter Sholander, Glenn Frank, and Sean Swank
Scientific Research Corporation (SRC)
Atlanta, Georgia

Joseph P. Loyall and Gary Duzan
BBN Technologies
Boston, MA

ABSTRACT

This paper presents an overall architecture and prototype for Mission-Based Network Management in a Joint Battlespace Infosphere (JBI) that combines application, network, and MAC-layer QoS management. It describes an MBNM prototype system for an Air Force Cooperative Attack mission with swarms of unmanned aerial vehicles. The prototype implementation uses BBN's Quality Objects (QuO) software framework to develop a UAV Imagery Application that adapts its behavior based on mission-state and network-state in order to meet its required end-to-end quality of service; SRC's Wireless Ad hoc Routing Protocol (WARP) to implement a bandwidth-based MANET routing algorithm; and the Common Open Policy Service (COPS) as the policy distribution mechanism.

INTRODUCTION

The Joint Battlespace Infosphere (JBI) is a combat information management system that provides individual users with the specific information required for their functional responsibilities during crisis or conflict. A JBI must integrate data from a wide variety of sources, aggregate that info, and distribute it in the appropriate form and level of detail to users at all echelons. This requires the JBI to reside on top of a Global Information Grid (GIG) network (as shown in Figure 1) that can provide "mission-aware" Quality of Service (QoS). It also requires application-independent techniques that can allow both new and legacy applications to tailor their data flows to the current mission-state and the GIG's current network-state.

An operational JBI will require an overarching, yet decentralized, *Mission-Based Network Management* (MBNM) system that coordinates user behavior, across all the echelons of the JBI user community, based on current mission state and needs. MBNM requires an extension to current commercial *Policy-Based Network Management* (PBNM) standards, since those tools are focused on wired networks and static rules-based policies. The existing PBNM standards also do not consider multi-layer QoS

management combining dynamic application-layer adaptation techniques with network-layer and MAC-layer QoS techniques.

This paper presents an overall architecture for MBNM in a JBI. It presents implementation details for a specific MBNM prototype system for an Air Force Cooperative Attack mission that uses swarms of Unmanned Aerial Vehicles (UAVs). The prototype implementation uses BBN's Quality Objects (QuO) software framework to develop an adaptive UAV imagery application that dynamically manages its data content (based on both mission-state and network-state) in order to meet the required end-to-end QoS and SRC's Wireless Ad hoc Routing Protocol (WARP) to implement a bandwidth-based Mobile Ad hoc Network (MANET) routing algorithm. This algorithm supplies feedback on the end-to-end path bandwidths to the adaptive UAV imagery application. The Common Open Policy Service (COPS) is used as the policy distribution mechanism.

The composite MBNM prototype system is able to automatically trade off image-transfer latency, image quality and relative drop-priority as the various UAVs move through their mission cycles of "Ingress", "Target Search", "Target Identification", "Strike" and "Battle Damage Assessment". For example, the MBNM system can be configured to provide the highest quality, lowest latency image transfer to Strike Mode UAVs. The MBNM system automatically adapts its QoS settings to the percentage of UAVs in each mission-mode and the underlying network conditions. As such, the system is adaptive against network congestion, network-topology changes and adversarial jamming. We provide quantitative results based on an SRC/BBN test-bed that emulates the Air Force Cooperative Attack scenario.

Multi-Layer QoS Management

There has been extensive research and development on QoS support within each layer in the ISO model for both wired and wireless networks. As an example, MANET routing software [1] can be extended to allow individual users to obtain QoS-aware paths. IP Differentiated Services (DiffServ) can provide service differentiation [2] for existing application software, while maintaining

This work was supported by the Air Force Research Laboratory under Contract F30602-00-C-0032.

compatibility with legacy IP routers. Forward Error Correction (FEC) techniques can enhance both network-layer and link-layer QoS in the presence of error-prone wireless links. Link-layer FEC improves hop-by-hop reliability. IP-layer FEC, such as the RMDP (Reliable Multicast data Distribution Protocol) [3] that uses “erasure coding”, can improve end-to-end delivery at the network layer.

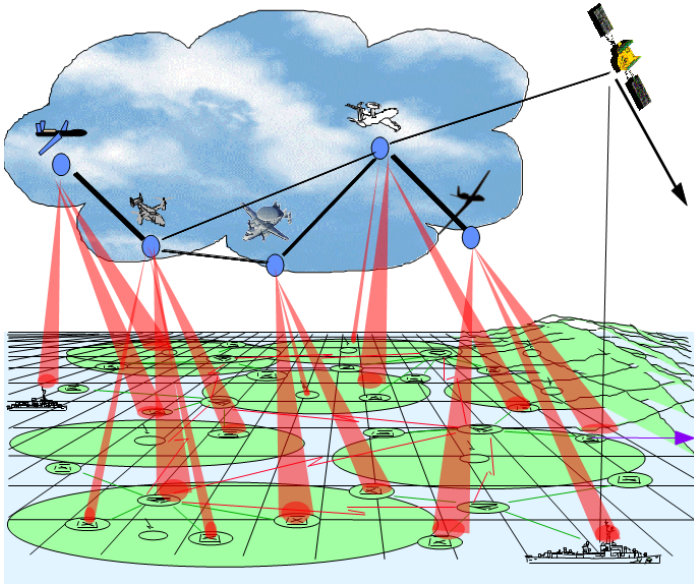


Figure 1. Joint Battle Infosphere Application Scenario

There has been less work however, on strategies for coordinating those disparate techniques – especially in the context of modern military information systems, such as those based on the JBI concept.

One of the core principles of the JBI is its publish/subscribe architecture that decouples information providers (publishers) from information consumers (subscribers) and both from the infrastructure that underlies the JBI. This provides challenges for supporting QoS in a JBI using only network-, MAC- and link-layer QoS management techniques. Not all connections to and through the JBI will be equal. Some will be through high-bandwidth connections, such as satellite communications or dedicated LANs, while others will be through highly constrained links. This means that some nodes will be able to handle more data than others, while some data sources will be able to produce more data than their connections, and the infrastructure underlying a specific JBI, can deliver. In the absence of application-layer and mission-layer prioritization and management of this data, the network layers and below will decide what gets through and when. Network-layer QoS management provides important support for prioritizing data flows. However, most application-layer and mission-centric

information – which is needed to distinguish the priority of data and to mediate the conflicting needs of applications within and between missions – may be lost at the network-layer and below. This motivates the need for “multi-layer” QoS management and enforcement that spans all of these protocol layers.

This coordination requires application-independent techniques that can allow both new and legacy applications to tailor their data flows to the current mission-state and the current network-state. Ideally, those techniques should use a component-based approach that provides a low “cost-of-entry” for users while maintaining backwards compatibility with their existing communications and data-processing systems.

Figure 2 shows several potential Application Layer and Network Layer techniques for QoS enforcement. The items shown in gray are discussed in detail in later sections. Multi-application rate shaping allows cooperating applications to adjust their behavior based on shared notions of their relative importance to the user. In IP network, call-admission control techniques such as “bandwidth brokers” are another proposed option. Finally, newer applications could use QoS-aware middleware to leverage both application-layer and network-layer QoS management techniques. (Note: legacy applications might be constrained to network-layer QoS management techniques, such as DiffServ. As such, they might only use the call admission control, packet marking, rate shaping, QoS routing and FEC technologies.)

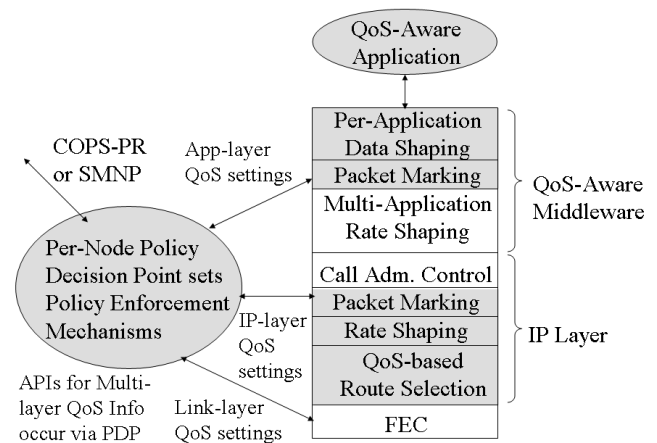


Figure 2. Multi-Layer QoS Enforcement Points

Mission-Based Network Management (MBNM)

An additional barrier to large-scale JBI deployment is the need for an overarching “Mission-Based Network Management” system that can coordinate user behavior

across all the echelons of a large JBI based on current mission-state and user-needs. In particular, for military applications, MBNM can adapt (and enforce) the QoS management policies based on operational phases (e.g., mission planning, mission deployment, and mission execution). This MBNM capability can be based on current Policy-Based Network Management (PBNM) tools such as the Common Open Policy Service (COPS) [4].

As an example, consider the Tactical Unmanned Aerial Vehicle (TUAV) scenario shown in Figure 3. In Ingress Mode, the TUAVs are moving from their launch point towards their search areas. During Target Search mode, they are searching for targets. During Target Identification (ID) mode, they have identified a target, and are awaiting permission for a strike. During Strike Mode, the other remaining TUAVs within a cluster loiter and publish image information through a JBI to the Air Operations Center (AOC). Based on the first TUAV’s perceived success/failure, the remaining TUAVs may either return to Target Search mode or re-enter the Target ID – Strike – Battle Damage Information (BDI) cluster of mission states. As shown in Figure 3 and Figure 4, this TUAV scenario also entails a ground-based unit that subscribes to imagery in its vicinity using the JBI.

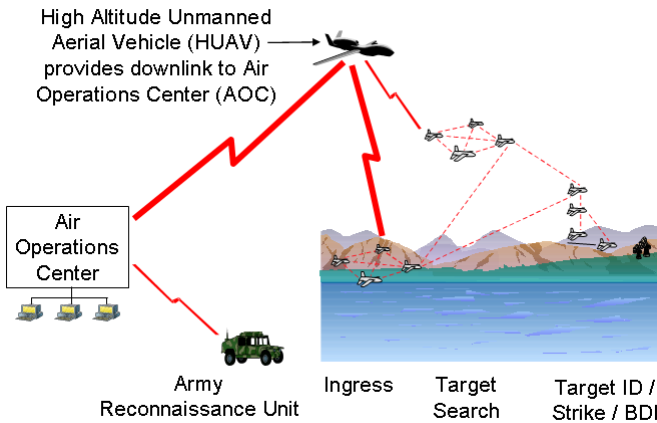


Figure 3. Mission-States for TUAV Application

Figure 4 shows the information flows associated with this exemplar mission scenario. If for example, the TUAVs can source more image data than the HUAV to AOC link can handle then this motivates the need for both multi-layer QoS management and mission-aware network management. These two techniques allow the TUAV’s Image Applications to adjust their sending rate based on the image consumer’s subjective QoS requirements, the underlying network congestion, and the relative importance of that TUAV’s current mission state. As an example, image data from TUAVs in Ingress Mode may have lower priority and require less resolution than image data from TUAVs in Target ID mode.

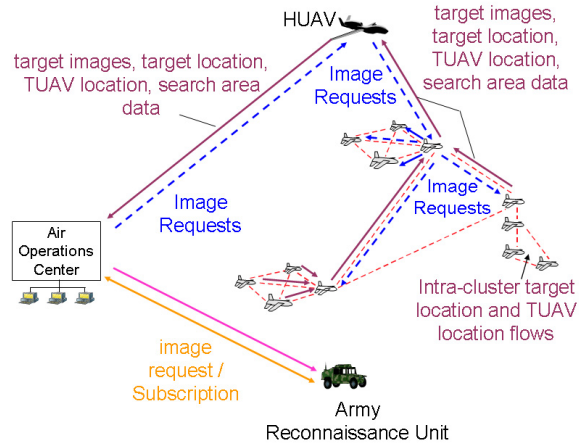


Figure 4. Information Flows for TUAV Mission Scenario

The remainder of this paper presents a standards-based architecture for:

- Multi-layer QoS management that coordinates both application-layer and network-layer QoS monitoring and enforcement techniques. This improves on existing network level QoS enforcement mechanisms by allowing lower-level network and MAC layer network management to be based upon higher-level mission priorities and requirements.
- “Mission-based” QoS management that provides adaptation and enforcement of the QoS management policies based on relevant operational phases, and the distributed coordination of user behavior based on mission-state and QoS-state. For the exemplar TUAV application shown above, those mission phases are Ingress, Target Search, Target ID, Strike, and Battle Damage Information (BDI). This improves on current state of the art in application management so that applications adapt and coordinate their behaviors to the resources available to them.

NETWORK LAYER POLICY DISTRIBUTION and QOS ARCHITECTURE

This section describes how the COPS protocols can be extended to provide mission-aware network management capabilities. It also outlines the network-layer QoS management techniques that were used in this research’s test-bed. The next section describes the application-layer QoS management techniques, and how they were combined with the network-layer QoS management

techniques in order to produce a “mission-aware, multi-layer QoS management” system.

Extensions of Common Open Policy Service – Provisioned (COPS-PR) to JBI and MANETs

This research used the policy distribution and management architecture shown in Figure 5. The items in boldface are part of the IETF’s standard policy-based network management architecture. The items in italics are the technical features that were added during this research.

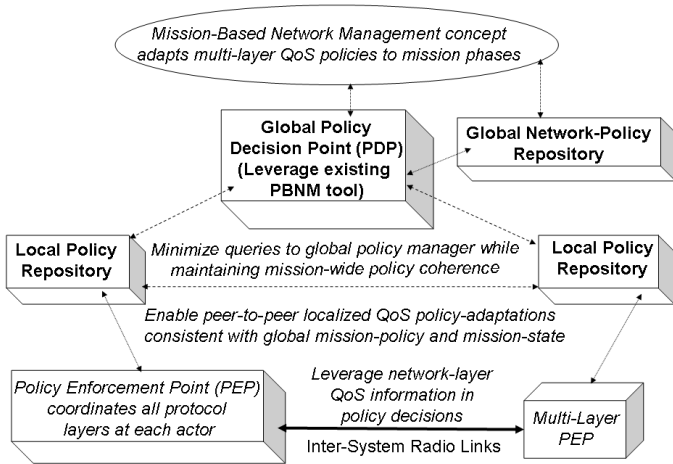


Figure 5. Policy Distribution Architecture

The Common Open Policy Service – Provisioned (COPS-PR) can be used as a baseline policy-messaging protocol for research on mission-based network management systems. COPS-PR [5] is a policy-based extension to traditional Simple Network Management Protocol (SNMP) based network-management approaches.

To continue the description of the military-centric exemplar scenario, during Ingress Mode a scenario generator emulated a TUAV being launched from a high-altitude platform. Just before “launch”, each TUAV downloaded its current mission-aware policies from the “Global PDP” located on that high-altitude platform to the “local PDP” located on that TUAV. The associated COPS-PR messaging is outlined in Figure 6.

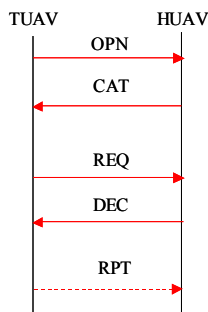


Figure 6. Policy Messaging within COPS-PR

For client registration, the Client Open (OPN) message identifies TUAV’s policy enforcement capabilities to the Global PDP. That entity then returns a Client Accept (CAT) message verifying that the Global PDP can supply those policies. The TUAV requests a policy download/update from the Global PDP with a Request (REQ) message. That entity responds with a Decision (DEC) message that supplies the requested policy information back to the TUAV. The optional Report (RPT) message then completes the initial policy-download. For the other mission-modes, each TUAV cluster leader sends a Report (RPT) message to the Global PDP upon a mission-state change. The Global PDP transmits Decision (DEC) messages that contain each cluster’s mission states to all TUAVs, which use that mission state, along with *locally sensed* application-layer and network-layer QoS metrics, to drive QoS adaptation on a per-node basis. This simple “Situational Awareness (SA)” system fuses Common Operational Picture data with traditional QoS metrics in order to improve network utilization and application-layer subjective QoS. For implementation purposes, the Global PDP can batch several received mission-state updates (from the local PDPs in the TUAV clusters) into one timer-driven DEC message.

This research used a subset of the IETF’s standard Policy Information Base (PIB) for IP Differentiated Services (DiffServ). The key elements were the Differentiated Services Code Point (DSCP) Marking Value, Priority, and Rate Shape Value. That existing “Framework” PIB [6] was then extended to include mission-state within the Policy Information Base (PIB).

The COPS-PR architecture requires a persistent Transmission Control Protocol (TCP) connection between the Global PDP and each UAV’s Local PDP. This client-server approach would be unworkable in large dynamic JBIs that include MANETs. As such, additional research is needed on a “peer-to-peer” version of a COPS-like protocol. A related research thread is experimenting with COPS-PR over reliable multicast rather than reliable unicast. A third issue is the positioning and number of “Global PDPs” for a deployed JBI that included MANETs. A large network will probably require several levels of PDPs between the individual actors and the rear-echelon command centers. A final issue is that an MBNM system for military applications should interface to existing command and control systems, in order to automatically learn various actors’ mission-states.

Network Layer Policy Enforcement Point (PEP)

A simple implementation for a network layer policy enforcement mechanism (see Figure 2) can use the Linux netfilter (*iptables*) and iproute2 traffic control (*tc*) utilities. Combined, these utilities provide the capability to mark IP packets with a Differentiated Services CodePoint (DSCP) value, perform priority-based QoS routing based on the DSCP value, and traffic shaping (rate limiting) on the traffic exiting the node's network interface. For mission-based network management, the *iptables* rules and traffic control setup can be implemented based on the contents of the PIB data received by each node's COPS-PR client from the COPS server and mission-specific policies stored at each node.

Network-layer packet marking may be preferable for legacy applications and COTS applications for which source-code is unavailable. However, application-layer packet marking may be preferable if source-code is available or if intelligent QoS-aware middleware is used. In that case, individual data-units within a single application flow can be marked differently. One simple example would be to give higher priority to a small "region of interest" within a larger image. The next section discusses the application-layer QoS management techniques in more detail.

ADAPTIVE IMAGE APPLICATION

In a JBI, many different applications will be publishing a wide variety of data. At any given time, some of this data will be vitally important to current operations, while other data will be less important. Still other data may be completely useless under the current circumstances, and may in fact be harmful due to its consumption of resources for its generation, transmission, and processing. While network-level QoS is capable of managing prioritized traffic, ideally the traffic should be prioritized (and possibly "shaped") by the application-layer first.

Ideally, all applications would know about the current mission and be able to respond accordingly. Unfortunately, modifying existing systems to be "mission-aware" can be: a) costly due to the often wide-ranging impact on the system code; and b) inflexible in the presence of mission-policy changes. One possible solution would be to have a separate manager assigning priorities to particular applications based on the current mission-state. However, this does not deal with the fact that applications can generate data with different levels of importance. It also does not scale in widely distributed, dynamic environments.

One solution to this data-management problem, illustrated in Figure 7, is the use of middleware for managing the

application QoS. This approach inserts data-management capabilities in the middleware (e.g., BBN's open-source Quality Objects (QuO) software [7]) between the applications and the network infrastructure. This data-management middleware receives information on the spatially- and time-variant network capacity and encapsulated notions of mission priorities. It then uses that information, in conjunction with interfaces to the network-management controls, in order to prioritize, filter and process the data (e.g., via shaping or compression). This helps ensure that:

- (a) the data most important to the mission is delivered with the highest level of QoS.
- (b) the available network resources are efficiently utilized.

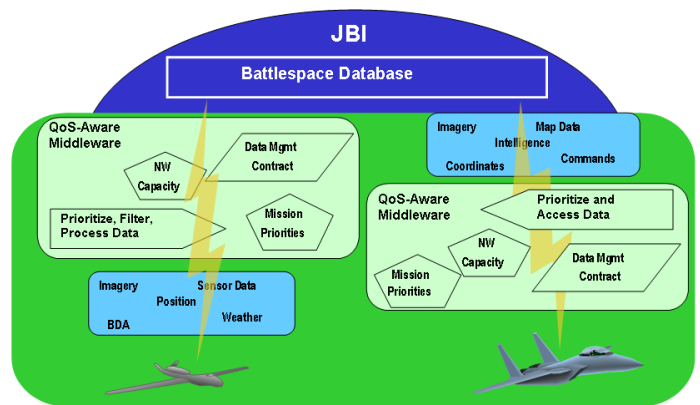


Figure 7. Middleware-based QoS data management ensures that mission-important data is delivered and that network resources are efficiently utilized.

Adaptive QoS management involves three stages: collecting the inputs to the policy decision making process, the policy decision making process itself, and performing the adaptations performed as a result of the policy decision. In this paper's example mission, the policy's goal is to transfer imagery from the TUAVs to the AOC given the current network-layer operating constraints and application-layer subjective/objective QoS requirements. As such, the adaptive QoS management process takes the following inputs into account:

- Environmental / Network Constraints
 - Available Bandwidth
- User / Application-Layer Requirements
 - Maximum/Minimum Image Scale
 - Image Quality
 - Deadline

QuO “contracts” encode policies for negotiated application-layer behavior based on these inputs and in response to (or anticipation of) different network conditions and user requirements. Encoding such policy in the middleware allows policies to be created and changed without re-writing application code. “System condition objects” are then used to interface to external sources of information such as mission state and external controls such as resource managers via middleware-based distributed object interfaces. System condition objects provide a consistent CORBA (Common Object Request Broker Architecture) interface for updating and retrieving QoS information. This makes them easy to use, yet powerful enough to trigger complex behaviors. For this research, the QoS-aware middleware interfaced to the network-layer QoS routing software using system condition objects in order to provide a seamless integrated data and resource management capability.

The following simple application-layer adaptations were used to meet the end-to-end application-layer QoS requirements:

- Still Image Manipulation
 - Image Scaling
 - Tiling
 - Tile Compression (lossless/lossy)
- Still Image Transmission
 - Bandwidth-based Pacing

These adaptations can be implemented as QuO “Delegates” that modify the application’s normal execution path in order to introduce new behaviors. Delegates provide interceptors between application interactions, to intercept operations on application objects in order to invoke adaptive behaviors and apply QoS management behaviors. The delegate can check the state of contracts to determine what adaptive policy to follow and prioritize its data, filter or process its data, or interface to network management controls according to the mission policy.

In this paper’s example TUAV mission, the policy contract is first evaluated when image transmission begins to determine what level of scaling is required to transmit the imagery (from the TUAV to the AOC) before the user’s deadline. If the contract determines that the bandwidth resources are insufficient to meet the deadline with the defined constraints (e.g., with the desired frame-rate and image resolution) then the imagery will not be sent and the request will generate an exception. Otherwise, the imagery is scaled to the appropriate level and transmission continues. The imagery is then broken into tiles to allow

finer-grained control, and the contract is evaluated again for each tile to determine the appropriate level of compression in order to meet the deadline. Again, if the contract determines that it is impossible to complete the imagery transmission by the deadline, the remaining tiles are discarded and the imagery is considered complete. The tiling order is chosen so as to send tiles for an “area of interest” first, so that if resources become scarce part of the way through the transmission, the most important tiles will have already been sent. If there are sufficient resources to continue, the contract determines the proper compression level for each tile and sends it at a rate appropriate for the current available bandwidth.

This prototype includes an “Image Adapter” that publishes imagery from the TUAV Imagery Application to the JBI and a simulated Army Unit that subscribed to imagery for its location with certain QoS requirements. The goal is to have the Image Adapter request the transfer of imagery from the appropriate TUAV with the Army Unit’s desired quality of service. The TUAV Imagery Application manages the imagery transmission to meet the requirements, if possible, and the results are published to the JBI where the Army Unit receives them through its subscription.

Unfortunately, the basic publish and subscribe operations don’t provide the communication channel necessary for a publisher to discover that there are subscribers interested in information that it can offer. To deal with this and similar issues, we added a “publish on demand” feature to the JBI platform. The publisher registers an “offer” to publish including a description of offered objects and a callback object that is invoked when a subscription matches the offer.

For a subscription to effectively match an offer, the publisher and subscriber need to agree on how to specify QoS for objects of interest. In the prototype, attributes relating to the data itself, such as size and quality, were included in the metadata and predicate, while information about the timeliness of the data delivery was provided via the subscriber’s sequence attributes.

Notice that the prototype uses QuO and JBI middleware to maintain the decoupling of the UAV imagery publishing from the army unit imagery consumption, even while it coordinates data and network QoS management to get end-to-end QoS for the image delivery.

SYSTEM TEST-BED and TEST RESULTS

A test-bed was implemented to demonstrate the feasibility of mission-aware multi-layer QoS management in a JBI. This section provides a brief outline of the techniques used to construct that test-bed, and the results gained from it.

Network Topology Emulation

The test-bed connects the computers into a single Ethernet hub or Wireless Ethernet LAN segment. It then performs the network topology changes necessary to simulate mobility (and the breaking and forming of network links) via IP Layer firewall rules [8]. In particular, a scenario generator distributes the firewall rules to the test computers (as shown in Figure 8) and then records relevant network traffic for later analysis. Those rules (which can be implemented via *iptables* in the Linux 2.4.x kernels) are used to configure each computer to drop packets from “non-neighbor” nodes. This approach allows emulated mobility in a mixed network of wired desktop PCs and wireless laptop PCs.

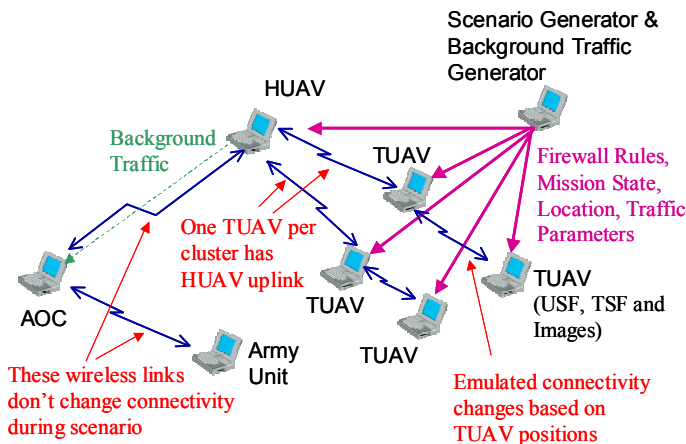


Figure 8. Lab Tests Use IP Firewall Rules to Enforce the Desired Connectivity Matrix

The Background Traffic-Generator emulates traffic from other TUAVs. The Scenario Generator generates mission-states, positions and traffic parameters for both emulated TUAVs (laptops) and virtual TUAVs. The emulated TUAVs run the MBNM and TUAV Image Application software. The virtual TUAVs provide additional network loading within the test-bed. The Scenario Generator also triggers the application flows associated with each mission-state. From a QoS standpoint, information flows associated with Target ID and Strike mode have priority over information flows associated with Ingress and Target Search mode. (Note: the total offered-load is chosen so that “Ethernet effects” did not appreciably impact the measured results.)

The emulated transmission-ranges for the TUAV-to-HUAV links and the HUAV-to-AOC link are user-configurable. The rate-shaping capability of the Linux Traffic Control (*tc*) utility is used to emulate the maximum data-rates on the various links.

The Scenario Generator typically enforces the following constraints of the scenario’s network connectivity. All communications between a TUAV cluster and the AOC occurs via the HUAV. Each TUAV cluster has a “cluster leader” that has an uplink to the High Altitude UAV (HUAV). All other traffic from a given cluster, is funneled to the AOC via the cluster leader’s uplink. If the cluster leader goes into Strike mode then the uplink was switched to a different node in that cluster. Figure 9 shows a typical connectivity pattern during an emulated scenario. (Note: the GUI shown in Figure 9 is a SRC-developed Java program that helps visualize MANETs during both lab-based emulations and open-air field demonstrations. The image is a pre-stored aerial photo of the Windy Hill Area in Atlanta, GA – which is where SRC’s HQ is located. That tool provides real-time data capture and playback from stored logs.)

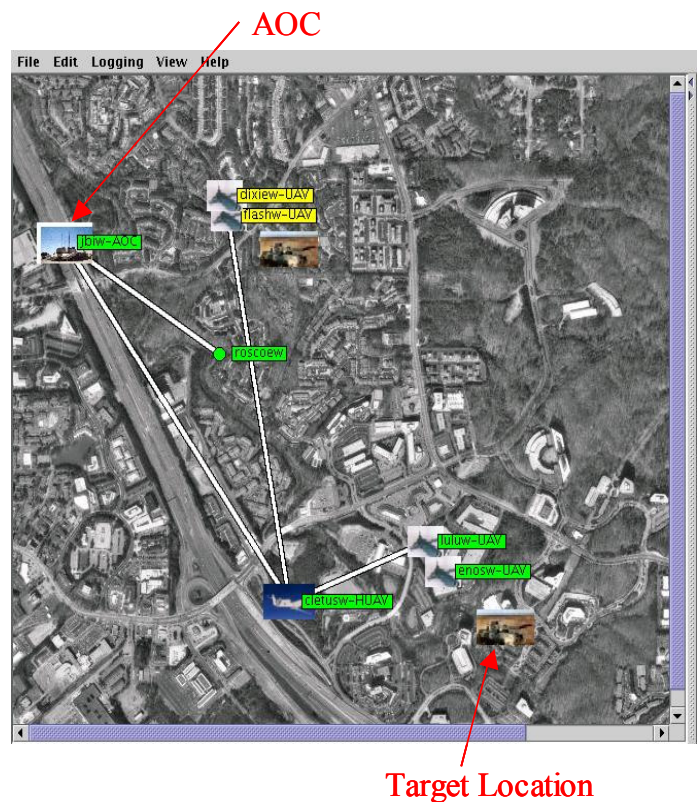


Figure 9. Typical Network Connectivity Pattern During TUAV Scenario

Mobility Model

During Ingress Mode, the TUAV moves from its launch point to a randomly chosen point within its designated Target Search Area. The TUAV’s motion within Target Search Mode is then either random motion or random-waypoint motion. During Target ID and BDI Mode, the TUAVs either continue their random motion or move towards the target location. During Strike Mode, the TUAV moves towards the target location.

QoS Routing Protocols for Mobile Ad hoc Networks (MANETs)

This project used an implementation of Zone Routing Protocol [9,10] that uses a simple bandwidth-based QoS metric representing the current available link bandwidth from a node to its one hop neighbor. The available link bandwidth is calculated by periodically sampling the total number of bytes transmitted and received from the network interface. The byte count is parsed from the Linux kernel maintained interface statistics file `/proc/net/dev`. Next, a kbits/sec used bandwidth value is computed that is based on the current sampled byte-count, the previously sampled byte-count, and the sampling interval. The calculated “used-bandwidth” is then subtracted from the maximum link bandwidth, a configurable parameter, to derive the “available link bandwidth”. This calculated metric is then mapped to a cost metric where the higher the available link bandwidth, the lower the routing-cost metric. (Note: in routing protocols, smaller QoS metrics indicate better paths. This is an analogy to lowest hop-count.) Finally, a MANET routing protocol was used (rather than OSPF) because of the mobility rates within the TUAV cloud.

Traffic Sources and Traffic Models

The test-bed uses three different traffic sources. They are: a) UAV State Frame and Target State Frame Generators; b) a UAV Image Application; and c) Virtual Node Traffic Generators. The UAV State Frames (USFs) and Target State Frames (TSFs) are generated on a user-configurable timer. The UAV State Frames contain information on each TUAV’s position and heading. The Target State Frames contain information on the Target’s classification, position and heading. Within the test-bed, a synthetic traffic load is generated by the virtual TUAVs. The emulated TUAVs generate actual image flows, and USF/TSF flows with dummy payloads.

Test-Bed Results

For the test-bed, each image (e.g., pre-stored aerial photo) is broken up into “tiles” as illustrated in Figure 10. The tile sending-rate is based on the Available Bandwidth metric supplied by the network-layer and the various application-layer constraints on image timeliness and image quality. The TUAV Image Application marks the DSCP in an outgoing IP packet based on the mission-state. (The MBNM Client supplied the mapping of mission-states to DSCPs to the TUAV Image Application.) The Network Layer PEP then provides differentiated queuing and rate-shaping based on DSCP markings of each IP packet. Finally, the TUAV Image Application applies compression algorithms and image scaling algorithms to the image data as necessary to meet the user’s timeliness

deadlines. The TUAV Image Application would abort an image transmission if the deadline has already passed or if it determines that meeting the deadline is impossible. The scaling options include full-scale, half-scale and quarter-scale. The compression options include loss-less (PNG format) and lossy (JPEG format).

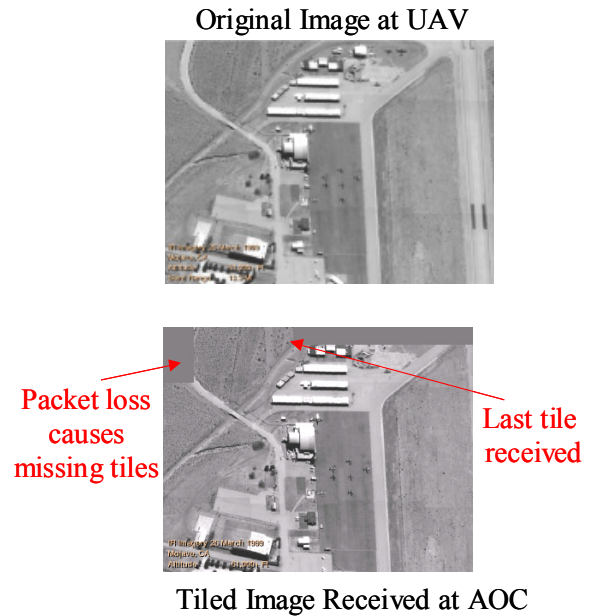


Figure 10. Tiled Images Emulate TUAV Image Stream

The TUAVs send tiled images automatically during Target ID and Strike Mode. Without QoS Management, excessive tile loss occurs as shown in Figure , since every UAV Image Application sends at its maximum rate. With QoS Management, the TUAV Image Apps dynamically learn the Available Bandwidth along their respective network paths to the AOC.

Received Image at AOC (w/o QoS Management)



Packet loss
causes
missing tiles



Received Image at AOC (with QoS Management)

Figure . Tile Loss Rate Showed Subjective Impact of Mission-Aware Multi-Layer QoS Management Techniques on End-to-End Application-Layer QoS

In that case, each TUAV Image Application sends at a slower rate, but this yields a lower tile-loss rate as shown in Figure . The QoS-managed system automatically trades off added latency (or a lower frame-rate) for less tile-loss. The TUAV Image Applications also take into account application-layer QoS metrics such as the timeliness deadline for each image. Based on the user requirements and the available bandwidth, the TUAV Image Applications then apply the “best” combination of image scaling (full-frame, half-frame or quarter-frame) and compression algorithm (lossless vs. lossy). Images from Strike Mode and Target ID Mode TUAVs get highest priority, and the largest bandwidth allocation in the Traffic Control Queue at each TUAV’s network-layer PEP. The USF/TSF info gets the lowest priority and lowest BW allocation.

CONCLUSIONS

This paper presented and analyzed an architecture for mission-aware multi-layer Quality of Service (QoS) management, within a Joint Battlespace Infosphere (JBI) context that leverages QoS-based routing and QoS-aware middleware. It illustrated the benefit of that cross-layer approach in a Tactical Unmanned Aerial Vehicle (TUAV) application such as “cooperative attack”. It also discussed the necessary extension to existing policy-based network management protocols, such as the Common Open Policy Service, that are required to further optimize network-layer utilization and application-layer QoS in a JBI.

REFERENCES

- [1] “IETF Working Group, Mobile Ad-hoc Networks”, www.ietf.org/html.charters/manet-charter.html
- [2] K. Nichols, S. Blake, F. Baker and D. Black, “Definition of the Differentiated Services Field (DS) in the IPv4 and IPv6 Headers”, RFC, 2474, Dec. 1998.
- [3] L. Rizzo and L. Vicisano, “RMDP: An FEC-based Reliable Multicast Protocol for Wireless Environments”, Protocol for Wireless Environments, Mobile Computing and Communications Review, Vol. 2, No. 2, April 1998.
- [4] Boyle, J., et al, “The COPS (Common Open Policy Service) Protocol”, RFC 2748, January 2000.
- [5] K. Chan, et al, “COPS Usage for Policy Provisioning (COPS-PR)”, RFC 3084, March 2001.
- [6] R. Sahita, et al, “Framework Policy Information Base”, RFC 3318, March 2003.
- [7] “Quality Objects Project”, <http://quo.bbn.com/>
- [8] “Ad hoc Protocol Evaluation testbed”, <http://apetestbed.sourceforge.net/>
- [9] P. Sholander, A. Yankopolus, P. Coccoli and S. Tabrizi, “Experimental Comparison of Hybrid and Proactive MANET Routing Protocols”, MILCOM 2002, October 2002.
- [10] Z. Haas and M. Pearlman, “Determining the Optimal Configuration for the Zone Routing Protocol”, IEEE JSAC, Special Issue on Ad-Hoc Networks, August 1999.