2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing

Dynamic Quality of Service Management for Multicast Tactical Communications

Matthew Gillen, Joseph P. Loyall
BBN Technologies
Cambridge, MA USA
e-mail: {mgillen, jloyall}@bbn.com

Joshua Sterling
US Air Force Research Laboratory
Rome, NY USA
e-mail: joshua.sterling@rl.af.mil

Abstract**—**Wireless networking is moving toward the adoption of IP protocols and away from the multitude of special-purpose tactical radios traditionally in the hands of emergency personnel, military personnel, and law enforcement. The adoption of standards, such as IP multicast, has facilitated this. IP multicast also enables recovering some of the advantages of the broadcast medium when using IP in tactical environments. However, the traditional Quality of Service (QoS) approaches for IP multicast fall short of satisfying the stringent QoS requirements in tactical environments, which typically have single-hop, line-of-sight connections. The reasons for this are (1) QoS in IP networks, frequently based on Differentiated Services, relies on routers to enforce the priorities which typically don't exist in tactical networks; and (2) QoS for tactical users needs to be enforced at the information level, not the packet level where the loss or delay of a single packet can invalidate an entire object of information. We present strategies for QoS management for IP multicast in tactical environments that provides information- and user-level QoS and addresses the specific challenges of tactical radios (such as the lack of reliable capacity information). We present our solutions in the context of a tactical information broker that provides beyond line-of-sight information management in a theater of operations.

Keywords- quality of service, multicast, information management, tactical systems, Cursor on Target

I. INTRODUCTION

The vision of ubiquitous connectivity and a global information grid is becoming a reality, with more and more previously disconnected users having handheld devices through which they can form ad hoc networks and through which they can reach back to richly-resourced systems and, therefore enterprise services, the Internet, and other wide-area networks. The commercial vision for this is obvious, with constant access to information in everyone's hands for social networking, entertainment, navigation, reference, and a host of other applications.

Beyond the commercial uses, there is a base of *tactical* users as well, which includes military personnel, first responder emergency personnel, law enforcement, etc., that have traditionally relied on voice-only radio communication, using a variety of specialized protocols that differ based on the service, affiliation, chain-of-command, and vendor. These communities of users are starting to become interconnected following the trend of wireless access to the commercial Internet-based community, despite their special needs for dedicated networks, special-purpose protocols, data formats and encodings, security and reliability guarantees, and isolation from users outside their specific community.

Two of the trends that have helped move along this path are the following:

- The adoption of IP-based communication over radio connections.
- The movement toward modern software engineering technologies, away from stove-piped, built-from-scratch systems.

Traditional tactical communications has been from radio transmitter to radio receiver using a broadcast and line-of-sight (LOS) medium. Traditional, legacy radios were designed to recognize specific waveforms so that only radios of a particular type could talk to one another, and any radio of that type within broadcast (i.e., LOS) range could receive a transmission.

Internet Protocol (IP) overlays, and especially IP multicast, has enabled a revolution in tactical communications and a move toward packet-based and routed communications across networks of tactical users. IP multicast is overlaid onto the underlying radio broadcast medium, enabling ad hoc networks, beyond-LOS relaying of information, and support for more information types and protocols.

Simultaneously, there is a move away from building stove-piped systems with specialized interfaces, to building systems on commodity operating systems, from standards-based services and components, and with standards based interfaces, such as Pub-Sub-Query [7], Cursor on Target [17], and Web Services [18].

These trends have to come with additional research and development addressing the specific performance, footprint, security, and reliability needs of tactical users. Specifically, although there has been significant research in Quality of Service (QoS) for multicast [1], [2], [5], [10], [14], it has primarily addressed packet differentiation, not the information-, application-, and client-level QoS necessary, where QoS policy must be applied to an entire information object, stream, or client-server interaction, and the tolerance for information loss or delay can vary dynamically over time and types of information.

In this paper, we describe an approach that we have developed for application-level QoS for IP communications in tactical deployments. We have developed our QoS management in a Pub-Sub-Query middleware layer based on our previous QoS research [12][13] and existing standards, and validated it in a series of live-flight experiments utilizing a US Air Force developed Pub-Sub-Query information broker

1555-0885/11 $26.00 © 2011 IEEE
DOI 10.1109/ISORC.2011.11

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **MAR 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE **Dynamic Quality of Service Management for Multicast Tactical Communications**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Raytheon BBN Technologies,Cambridge,MA,02138**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14th IEEE Computer Society Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC), Newport Beach, CA, March 28-31, 2011

14. ABSTRACT

Wireless networking is moving toward the adoption of IP protocols and away from the multitude of special-purpose tactical radios traditionally in the hands of emergency personnel, military personnel, and law enforcement. The adoption of standards such as IP multicast, has facilitated this. IP multicast also enables recovering some of the advantages of the broadcast medium when using IP in tactical environments. However, the traditional Quality of Service (QoS) approaches for IP multicast fall short of satisfying the stringent QoS requirements in tactical environments, which typically have single-hop, line-of-sight connections. The reasons for this are (1) QoS in IP networks, frequently based on Differentiated Services, relies on routers to enforce the priorities which typically don?t exist in tactical networks; and (2) QoS for tactical users needs to be enforced at the information level, not the packet level where the loss or delay of a single packet can invalidate an entire object of information. We present strategies for QoS management for IP multicast in tactical environments that provides information- and user-level QoS and addresses the specific challenges of tactical radios (such as the lack of reliable capacity information). We present our solutions in the context of a tactical information broker that provides beyond line-of-sight information management in a theater of operations.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

[9]. The experiments have shown increased performance over baselines without QoS management, the ability to interface with legacy military systems, and the ability to support the data formats and rates needed for tactical operations.

The rest of the paper is structured as follows. Section II describes the challenges associated with providing QoS management in tactical multicast environments. Section III describes our QoS approach for tactical multicast. Section IV describes a case study in which we instantiate our QoS management in a tactical information broker utilized for beyond-LOS communications and situational awareness in military operations. Section V describes some empirical results from our experimentation. Section VI describes related work in multicast protocols for ad hoc networks and QoS management. Finally Section VII presents some concluding remarks.

II. Challenges in Providing QoS Management in Multicast Communication

At the core of our software stack is an Information Management (IM) system that provides publish, subscribe, archive, and query services for tactical users, as shown in Figure 1. The live-flight demonstrations and exercises in which we have been involved (described in Section IV) have needed QoS management primarily for servicing subscriptions, for the following reasons:

- The tactical users who are most interested in the real-time distribution of sensor data get it “pushed” to them by registration of subscriptions.
- The tactical users receiving sensor data through subscriptions are the ones on the most disadvantaged links.
- Subscriptions typically are periodic in terms of how data gets delivered.
- The nature of subscriptions is such that new sensors may come on-line, producing additional data to be disseminated to the subscription. Therefore, dynamic QoS management to handle the varying load is critical.

In contrast, publish operations in our scenarios usually do not require as much QoS management. In some cases, the sensors producing the most data are co-located with the IM services (e.g., on air-borne platforms). In other scenarios, the sensor feeds come over high-bandwidth links (e.g., Common Data Link [6]). Similarly, in the scenarios that we have encountered, users perform queries before a mission, not in the middle of one. Servicing queries for tactical users is often a one-shot deal and the user is typically interested in all the results that satisfy a query. Therefore, managing QoS for subscriptions is the most difficult and interesting problem given our scenarios.

Subscriptions in tactical environments are challenging. For efficiency, it would be ideal to transmit each item of information once for all recipients, mimicking the advantages of broadcast in tactical radios, instead of once for each recipient. The typical way to achieve that in IP networks is IP multicast. IP-multicast was originally intended to optimize and provide scalability for video delivery over the Internet, keeping the bandwidth requirements on the video provider constant with respect to the number of subscribers. IP multicast use in our scenarios with tactical radio networks is a little different, in that it is mostly used within a single LAN (i.e., there are no routers involved).

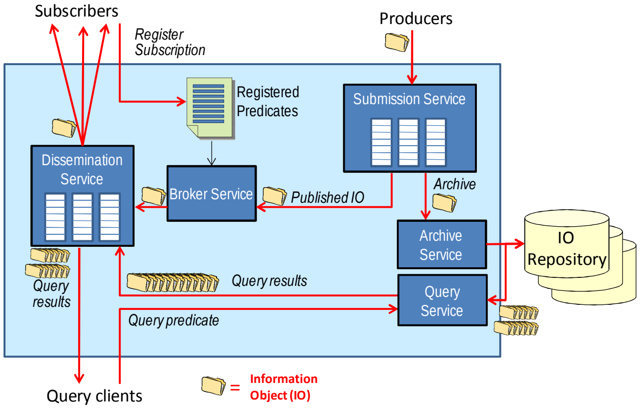

Figure 1. Core Information Management Services

Tactical IP-based radios (i.e., radios that are not based on 802.11 standards) present interesting challenges. On the one hand, there is an underlying broadcast medium that the radios use to communicate with each other. On the other hand, the IP layer puts a point-to-point overlay on the broadcast medium. IP multicast provides a way to get back the broadcast properties of the medium, and do it in a way that is supported natively by most operating systems’ network stack.

However, IP multicast presents specific challenges of its own when trying to use it as a substrate for subscriptions. We chose to implement the server as a static set of subscriptions whose endpoints (the receiver of the subscription ‘hits’) were multicast groups. This has the advantage of broadcast, where each item of information is only sent once, regardless of how many receivers there are. The downside of this approach is that IP multicast does not provide application-level visibility into who has joined which group (i.e., who is interested in receiving messages).

Some multicast implementations are called ‘sparse-mode’ multicast. These implementations will not forward a multicast packet from the local host unless at least one other node has joined the packet’s destination multicast group. This helps save bandwidth, but it makes it impossible for a sender to know exactly how much bandwidth is being used (the usual technique of measuring how many bytes are written to a socket does not work in this situation).

Other multicast implementations are called ‘dense-mode’, and generally write anything out to the media layer, regardless of whether there are any registered consumers for the destination multicast group.

These variations in implementation make it difficult to design a general purpose, application-level QoS management solution.

Previous work in integrating QoS and multicast (e.g., AQoSM [5]) have focused on providing optimizations for routers, and alter the implementation of multicast within a router. This does not work for our purposes for two reasons. First, in the typical IM-based tactical scenario, there are no routers between the information source (the IM system) and the consumers (tactical users) who are typically connected via a single hop, LOS tactical link. Second, many of these schemes assume a DiffServ configuration that matches the prioritization goals. Even if a given radio supports DiffServ, tactical radio configurations usually need to be approved well in advance of working out details for a particular mission. This means that it is unlikely that one could change the DiffServ configuration on the radios to accommodate a late-breaking mission requirement or dynamically unfolding situations.

Another challenge to using multicast is that it is UDP based. This means that the transport-layer protocol will not regulate itself to the capacity of the end-to-end link (like TCP would). To avoid uncontrolled and unpredictable packet loss (which can invalidate larger items of information), it is critical that the sender self-regulates the amount of data pushed out to the radio, to match it to the bandwidth available. To do this effectively, the sender needs real-time updates regarding the current capacity of the network, information that is not readily available in tactical networks.

Our approach, which manages QoS at the application layer, avoids the need to reconfigure radios and provides a solution that is agnostic to many of the differences in IP radio implementation. However, it depends on overcoming three challenges:

1. Determining how much bandwidth is actually being used;
2. Determining who is subscribed to what (so that system-wide prioritization may be implemented); and
3. Determining the current capacity of the radio network.

In the next section, we describe how we addressed these challenges in an actual distributed airborne tactical environment, and the basis it establishes for developing more comprehensive solutions to the challenges above.

III. Design and Development of QoS Services for Multicast-Based Tactical Communication

The context in which we were addressing QoS for multicast-based tactical communication includes a server-side IM system on an embedded airborne platform, with client-side information publishers and consumers. While technically the fact that the IM server is airborne is not important, what is important are the challenges and constraints implied by this context:

- The clients and server are mobile and connected to one another through LOS, single hop links.
- Information publishers and consumers are not connected to one another and, in fact, might not even be able to establish a connection because of distance, obstacles, or incompatible equipment.
- All information is routed through the server, and consumers share bandwidth to and from the server.

Furthermore, the client-side and server-side are legacy systems. The client-side system has the ability to ‘subscribe’ to multicast groups. The client discovers the available groups by listening for service advertisements sent to a well-known multicast group. The IM system on the server-side is also a legacy system, shown in Figure 2, including an Information Broker (InfoBroker) that supports point-to-point subscriptions, but not multicast. The IM system does not have the ability to allocate bandwidth on a per-subscription basis.

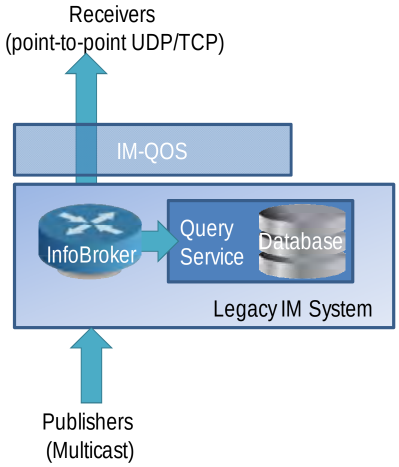

Figure 2. The legacy server-side IM system with an added QoS management component

A. *A Solution for QoS Management in the Multicast-based Tactical Environment*

Our goals for providing QoS management in this domain were to provide as much control and flexibility to the operator of the IM system as possible, while making the behavior of the system predictable and understandable.

Our first step toward solving the QoS management problem for this domain focused on the server-side. This was done for practical reasons—the client-side had a large installed user-base, whereas the server-side had no installed user-base and was therefore more amenable to modification—as well as technical—the server-side solution is more straightforward and less prone to implementation error than the client-side solution described in Section IV.A.

We added an IM-QoS component to the server-side legacy IM system, as shown in Figure 2. The IM-QoS component makes a subscription in the InfoBroker for each ‘service’ the operator wants clients to see as a unique offering. The IM-QoS has a configuration file that describes each ‘service’ in a tuple, consisting of

- The subscription filter that is used in the InfoBroker
- The name of the service to be used in the announcements
- The destination multicast group for events that are ‘hits’ on the subscription filter

The subscription filters are generic, and there is no requirement that the various services offered need to be disjoint in terms of the types of events that would be delivered to subscribers of those services (although in many cases it makes sense to configure the system that way).

The configuration of the IM-QoS component also allows the specification of a set of QoS modes. Each service can be in one mode at a time. A mode is defined as set of filters, where each filter has a set of properties associated with it, specifically

- Importance: the relative priority of events matching this filter to other events

- Replace by publisher ID: allows 'replacement' on outgoing queue on a per-publisher basis.

For example, a mode might have two filters, one that matches all imagery and sets them to high priority, and one that matches all location-updates, and sets them to low priority. A different mode might set imagery to low priority and location-updates to high priority and turn on replacement for location-updates.

Each service can have a bandwidth limit associated with it as well. This is an upper limit on how much data will be used by that service. This limit can be expressed as either an absolute number, or as a percentage of a dynamically updated "available capacity" (all services must use the same way of specifying the bandwidth limit). We implemented the option to use an absolute number because there is often no reliable way in radio networks to determine capacity at a given moment. An absolute value is sometimes preferable from a network planning point of view as well, since a particular data flow can be estimated a priori to require a certain amount of data per second. The 'percentage of capacity' option for specifying bandwidth limits is intended for making the best use of the network in situations with dynamically varying network capacity. The 'absolute' limit would potentially waste available network capacity if the capacity went above the sum of the absolute limits.

These features enable the IM system to provide the following predictable QoS to receiving clients:

- Differentiated delivery of high priority information.
- Management of queue growth for real-time information by sending the "newest" information.
- Aggregate sharing of available bandwidth by information flows.

The bandwidth limit allows us to maintain good network behavior even when using transport protocols that do not regulate themselves (e.g., UDP) or using self-regulating protocols in ways that do not allow the self-regulation to function (e.g., creating large numbers of individual TCP connections that each carry a small amount of data before being closed).

B. *Building to a Comprehensive QoS Management Solution*

One of the salient features of the scenarios we consider is that actors are coming into and out of the radio network over time. This is a major reason why the client-side side system listens for multicast service announcements to determine what is in the network at any given time.

At the same time, one of the drawbacks of our 'first step solution' is that the service definitions are statically configured on the server side. The clients must choose subscriptions based on the best approximation of their needs on each server that they encounter. It would be better if clients could specify their information needs in terms of generic subscriptions, and not necessarily tie a subscription request to a given server. A solution that we have begun implementing is to have clients make service announcements of their own, with user provided subscription requests embedded in the announcement. Servers in the area would listen for these announcements, and register them as standard subscriptions with the local Information Broker.

An issue with this approach is how to maintain the benefits of IP-multicast when there are multiple clients. A straightforward approach would be to have every registered subscription have a multicast endpoint. That opens up several other issues, some of which are user-interface issues for the clients, but some are more fundamental. For instance, assume a client creates a subscription for a given geographic area and another client comes in later with a request for a different geographic area. To make it interesting, assume these two geographic areas intersect, but neither is a complete subset of the other. It would be possible for the second client to join the first client's subscription group, and create a new subscription group with just the 'new' area that only interests the second client.

Such an approach might seem feasible for geographic filters. But what about more arbitrary metadata filters (e.g., a filter that incorporates 'type', 'source id', and or time)? Or consider differences in extrinsic properties, such as the 'replaceable' flag. It becomes exceedingly difficult to cover generic filters with clever combination schemes.

There is also the issue of partitioned networks, which can happen regularly in radio networks from LOS issues. In the example above, if the two clients do not have LOS to the same server, but have LOS to one another, the second client might register a subscription that is purely a difference from the first but only receive a subset of what he is interested in, because the server cannot see the first client's request.

This situation is exacerbated because the IM-QoS component does not know which clients are subscribed to which IP-multicast groups. This prevents both potential bandwidth savings of not publishing to groups that do not have subscribers, but also prevents prioritization decisions from being based on *who* is interested in the data.

A solution is to share the multicast subscription information at a higher level. Client-announcements would include information about the multicast groups that the client has joined. This enables better prioritization decisions and allows the system's behavior to better match commander's intent.

IV. A Case Study in Deploying on Tactical Information Management Systems

We have instantiated our multicast QoS management solution in an advanced tactical information management system, Marti[1]. Marti has the promise of enabling ubiquitous information access to tactical users in deployed environments. Marti includes the following capabilities not available in today's deployed tactical systems:

- Situational awareness in dynamic, rapidly changing situations.
- Rapid deployment of communications and information management in service of tactical operations.
- Beyond LOS communications using tactical assets.
- Reachback to users at the tactical edge through federation of IM services.
- Access through multiple interfaces, such as FalconView

[1] Named after Radio Marti, a Miami, Florida based radio broadcaster that transmits Spanish radio broadcasts to Cuba.

and C2PC, and using standardized information formats, such as Cursor on Target [17].

Marti is modular and service-based and consists of the following core components and services (shown in Figure 3):

- An information broker for matching published information to subscribers based on type, characteristics, and content.
- A *Query Service* to service requests for previously published information stored in the onboard database.
- *QoS Managed Information Management (IM-QoS)*, which disseminates information to subscribers, prioritizing important information, managing shared and constrained bandwidth, and adapting outgoing data rate and size to the available bandwidth.
- An *Image Chipper* that improves overall QoS by replacing large image payloads in messages with thumbnails, reducing the size of messages and increasing the rate that can be delivered through highly constrained bandwidth.
- A *Web Service* interface that supports requests for imagery through HTTP requests. This provides an interface through which users can request full images corresponding to delivered thumbnails.

In ongoing sets of live-flight experiments, Marti has been hosted on multiple platforms, including

- A high-altitude (up to 85,000 feet) balloon serving as a surrogate *High Altitude Long Enduring (HALE)* unmanned platform.
- Sensor pods, such as the LITENING Pod [11], attached to existing aircraft.

For IM services, including the information broker and query service, we utilize *Fawkes*, a tactical SOA-based information management service infrastructure developed by the US Air Force. Fawkes utilizes *PostGIS*, a PostgreSQL database that supports the CoT information format.

Marti includes client- and server-side QoS management with the following features:

- Bandwidth allocation.
- Rate limiting to fit within bandwidth limits.
- Prioritization of information based on subscriber needs.
- Image chipping to reduce the size of large images and support image retrieval on demand.
- Replacement of enqueued CoT messages with newer ones, supporting the tradeoffs of client preferences for timeliness or completeness of information delivery.

A. Client-side QoS Management

Marti includes a proxy that is co-located with an information publisher (e.g., a UAV sensor) and that provides a QoS layer between the publisher and the CoT Router. The proxy optimizes the quality of data delivery by making appropriate trade-offs between data rate and quality. It can re-shape data to reduce size (with potential reduction in quality) and/or adjust the rate of publication.

The proxy receives raw data from the publisher via a socket (both TCP and UDP protocols are supported) and modifies that data according to available resources and mission requirements, and then forwards it via a socket to the CoT Router. If sufficient outgoing bandwidth is available, the proxy simply passes data through unmodified. When bandwidth is insufficient to deliver the raw data at the desired rate, the proxy can re-shape that data to make the best use of the outgoing bandwidth. The proxy queues the modified data before sending, to facilitate more reliable delivery when the outgoing connection is intermittent and to control the rate of data sending.

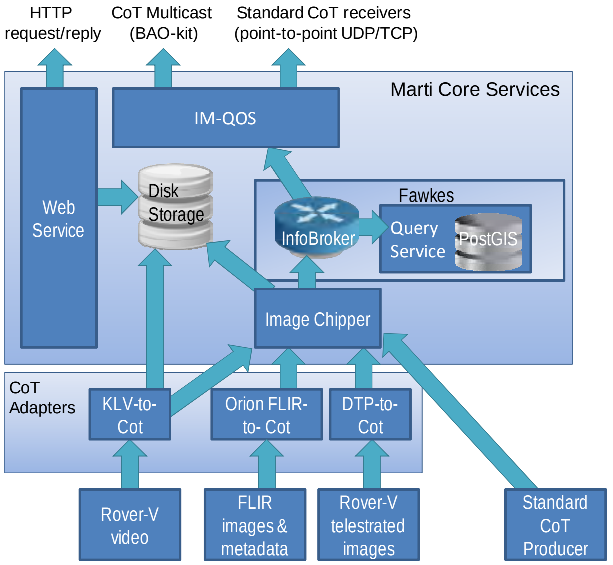

Figure 3. Marti has a modular service-based architecture and design, which supports multiple client applications providing and consuming information based on the Cursor on Target (CoT) standard.

One can command the proxy at runtime to favor data quality over rate, or vice versa. The proxy listens for incoming CoT events, which contain commands that control its behavior. The proxy supports information shaping including cropping, scaling, JPEG compression, and rate.

The proxy can monitor the signal strength of Marti's radio network via Simple Network Monitoring Protocol (SNMP). It uses this data to automatically determine available bandwidth, which is used to determine the QoS settings to enforce. If SNMP monitoring is disabled, the proxy can be configured for outgoing bandwidth via a CoT event.

B. Server-Side QoS Management

The IM-QoS component shown in Figure 33 encapsulates the server side QoS management. This functionality is similar to that on the client-side. It prioritizes, controls the rate of, and shapes data destined for subscribers based on the available bandwidth and the number and relative importance of the subscribers sharing the bandwidth.

C. Interface to Client-Side Applications

Marti's service-based design has enabled it to work with many existing tactical interfaces and equipment with which users are familiar. This is typically a huge problem for tactical environments, because there are many legacy systems, highly trained users, and severe weight and size constraints. Any new system that adds to the equipment that a soldier, firefighter, or EMT needs to carry or requires him to undergo new training to use, will face a resistance to adoption.

In contrast, Marti interfaces to several client-side applications in use by tactical warfighters, enabling them to interact in ways that they previously could not, providing access to information that was previously unavailable, and increasing the applications' utility and the users' situation awareness with high levels of QoS. The tactical applications that Marti currently interfaces with include FalconView [8], the Battlefield Air Operations (BAO) Kit [16], the Command and Control Personal Computer (C2PC) [3], and the Tactical Ground Reporting System (TIGR) [4].

D. Marti Information Formats

Marti uses the *Cursor on Target* standard [17], which has been developed to make interoperability between a variety of systems possible. Because Marti uses CoT as its native language, it easily integrates with and supports the large number of systems that already support CoT, either as producers or consumers. However, there are still many existing applications, such as legacy and stove-piped systems, that use specialized or proprietary messages.

As shown in Figure 3, Marti includes an *Adapter Layer* for supporting alternative information formats. Marti currently includes adapters that translate a variety of specialized and proprietary messages into CoT including the following:

- Capture still-frames from streaming MPEG-2 video with KLV-encoded metadata, and translate them into valid and accurate CoT messages.
- Turn data from a proprietary packet format used for telestrated images from a Rover unmanned aerial vehicle [15] into CoT events, so that users with the *FalconView* application can receive them.
- Capture still-frames and text metadata files from onboard *Forward Looking Infrared Radar (FLIR)* sensors and *fuse* them into CoT messages.
- Input MPEG-2 video with embedded metadata and extract still frames and the associated metadata, and generate a CoT message.

Using adapters instead of adding native support for other message types in the Marti core services allows Marti to leverage CoT tools, while keeping the core Marti services streamlined, modular, and extensible.

V. Experimental Results

QoS management is a key capability for an IM system that supports anonymous publish/subscribe. The point of using an IM system is to foster interoperability and opportunistic information dissemination. Put other way, IM allows less a priori planning of tactical networks, while getting greater benefit from them.

We have crafted a scenario that exemplifies some of the issues and demonstrates how QoS management allows the critical information for the mission to be delivered to clients, while still supporting the opportunistic delivery of other information that may be relevant to ongoing missions, but may not be critical. We compare the QoS managed version to a baseline (non-QoS managed) version of the scenario, and show how the QoS managed version improves the situation for ground users.

Our scenario starts with one airborne IM system feeding two independent ground units. We use two multicast groups, one for the precise participant location and identification (PPLI) information, and one used to send all the imagery. The initial configuration can fit all the published data for both multicast groups within available network capacity (220kbps). This situation could have occurred because of a priori network planning. There are four distinct imagery streams, with three active at the start of the scenario. One of these three imagery streams is over an area of interest (AOI), and thus is higher priority. The baseline system can handle the first part of the scenario well.

At *time=s* we introduce a new sensor (publisher) to the system. This could represent a soldier-cam that got turned on halfway through a mission or perhaps a UAV enters the tactical area to provide additional coverage. The new sensor will begin publishing a new imagery stream (the fourth stream becomes active). There are two possibilities for how this new data gets to the IM. Either a completely separate radio network is used or the same radio network that the tactical users are already using. For the second case, we are dealing with TDMA radio networks and each radio essentially has dedicated transmit time. Therefore, in either case we aren't concerned about the additional bandwidth from publisher to the IM system (since that bandwidth is unusable by any other actor in the network anyway). However, with the extra data now coming out of the IM system, we now have an overload of the IM's transmit time (i.e., the IM to clients link).

Table 1 shows the effects that the network infrastructure can have on our traffic, and how the QoS-management helps to mitigate those effects. We are running these experiments on a LAN and simulating the restricted bandwidth using the Token Bucket Filter (tbf) mechanism of the linux kernel.

Table 1: Mean latency of imagery traffic in milliseconds; standard deviation in parentheses

tbf limit (bytes)	Baseline-Important	Baseline-Other	Managed-Important	Managed-Other
20000	787 (112)	638 (94)	304 (149)	616 (397)
40000	1296 (451)	1459 (241)	320 (183)	685 (263)

Besides allowing us to set the maximum throughput, it also has a setting that controls how many bytes will be queued at the kernel layer. This queue depth setting is analogous to what might be provided internally with an IP radio. We ran both the baseline scenario and the QoS-managed scenario using two different settings for the queue depth. Table 1 shows the difference in the overload part of the scenario (after S). The QoS management maintains low latency of the important traffic under load. Furthermore, in the baseline, the effect of the load is dramatic: doubling the *tbf* buffer

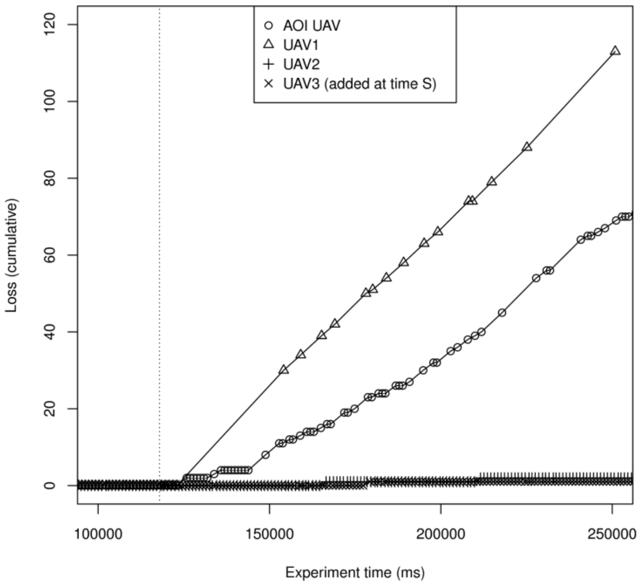

Figure 4: Baseline imagery loss

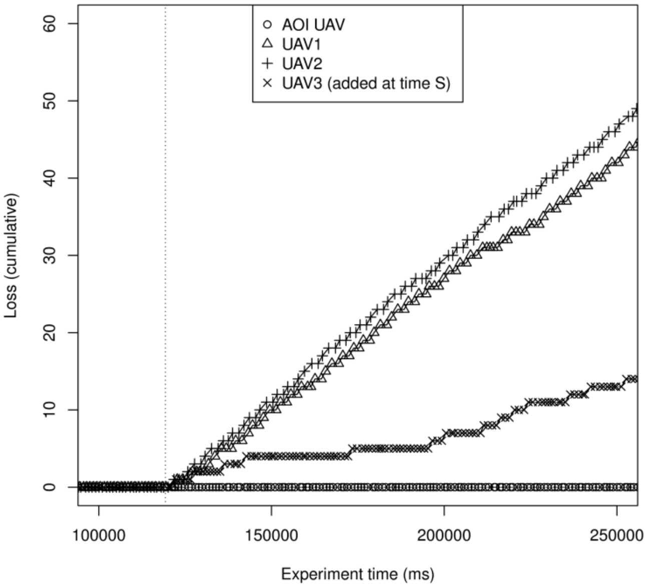

Figure 5: QoS-Managed imagery loss

nearly doubles the baseline latency. The QoS-managed version maintains a similar low latency regardless of the *tbf* buffer size.

For the loss graphs, we will be referencing the baseline and QoS-managed runs that used a *tbf* buffer of 20k.

As you can see in Figure 4, the baseline system performs predictably before time *s*, when the new sensor joins. After time *s* (indicated by the vertical line), the outbound network link from the IM becomes overloaded and the loss rate of all four imagery sources go up, with random and uncontrolled information loss. Figure 4 shows that in this run, the AOI UAV (which happens to be the most important) and UAV1 incur most of the lost packets.

Before explaining the results from the QoS-managed runs, it is important to note a few things about the QoS policy we used. Section III.A describes how each multicast group (service) can be assigned an independent bandwidth limit. We configured each group to have slightly more bandwidth than needed to handle the traffic for each group before time *s*. We do not change those allocations during the course of this experiment, even though our software does allow dynamic update of those values.

Within a multicast group, we have prioritization schemes. We also use a replace-per-publisher policy for the non AOI imagery streams. This will cause loss if there is any queuing, but it is more controlled than the baseline loss (we are choosing to lose old images from the less important feeds). For the imagery multicast group, we prioritize images over the AOI (i.e., from the *AOI UAV*) higher than other images.

The results for the QoS-managed run are shown in Figure 5. The early part of the scenario (before time *s*) looks similar to the baseline. After time *s*, several interesting things happen. First, The PPLI is unaffected by the additional load. This is because the PPLI messages are so small that the small amount of slack we had in the PPLI group's bandwidth allocation is sufficient to absorb additional traffic. But it is also important to note that the overload with respect to the imagery traffic does not affect the PPLI group.

Meanwhile, for the imagery group, the images of the AOI come through with no loss and latency similar to what was observed before time *s*. The non-AOI images begin to experience loss though, since the QoS-management is limiting the bandwidth usage and starts queuing. Since the non-AOI images have a replace-per-publisher policy, the queuing means that they will start to be dropped as newer events from the same publisher come in.

While these experiments were conducted in a lab, it is important to consider some aspects of the intended deployment environment when comparing our approach to another, such as DiffServ. DiffServ could potentially enforce some of the same prioritization (though there would still need to be an application-layer component deciding which DiffServ code-point to use for which messages). However, as mentioned in section II, tactical radios often do not have a DiffServ implementation, and even if they did, the configuration for the radios can be locked down far in advance of a mission, making DiffServ impractical.

Secondly, there are some policies that can be enforced only at the application layer, and not at the packet-layer. The 'replace-by-publisher-id' policy is a good example of something that cannot be implemented at the packet layer, since it requires deep inspection of the event.

VI. Related Work

Many multicast approaches are based on multicast trees, in which routing tables map out trees of routes to groups of nodes. Multicast transmits packets along a tree matching the group of nodes to which the packet is to be sent. The packet only is duplicated when a branch in the tree is reached. QoS management approaches in these multicast implementations target minimizing the span of trees, the cost of maintaining the trees, and the routing cost to reduce end-to-end delay, variation in the delay, and bandwidth constraints [14].

Aggregated QoS Multicast (AQoSM) [5] uses aggregated multicast to provide QoS management using DiffServ. AQoSM relies on a strategy to reduce the state explosion of maintaining multicast trees that serve multiple groups. AQoSM is targeted at QoS multicast provisioning in a single DiffServ domain over a backbone, routed network. In contrast, our approach targets tactical networks of individual radio links, where each link is a LOS connection, bandwidth declines with distance, and nodes at the endpoints of links are tactical platforms (handheld devices, ground or airborne vehicles) rather than routers, and DiffServ is not well supported, if at all.

Li et al describe an approach that exploits multicast trees, but utilizes them at the application layer in the form of an overlay multicast network [10]. Li shows that an application-level multicast tree approach in which each node makes decisions selfishly (to work with its neighbors to maximize its own QoS) can result in globally optimal QoS.

Other approaches to QoS management target mobile networks, most frequently ad hoc networks. The Mesh-evolving Ad hoc QoS Multicasting (MAQM) protocol uses a reservation based approach, in which each node in an ad hoc network determines the bandwidth available within a neighborhood around the node at session initiation [1]. QoS is enforced per-packet by dropping packets when QoS delivered is not what is expected. Our experience is that maintaining the bandwidth needed for a reservation is problematic in tactical networks with moving nodes, some of which are aircraft moving at high rates of speed. Estimates of available bandwidth as needed by this approach are also difficult and such estimates are not reliable. Furthermore, enforcing QoS at the packet layer can result in losing individual packets of an element of information, invalidating the entire information.

VII. Concluding Remarks

We have described an approach to application- and information-level QoS management that works with IP multicast and supports QoS in tactical environments better than current approaches to QoS for IP multicast. Our approach is a foundation for developing new QoS capabilities for tactical networks that work with information types, not packets; that compensates for some of the limitations of IP multicast in tactical environments; and works with line-of-sight, unreliable radio communication. We have instantiated our initial prototype in several tactical platforms as part of a tactical information broker that provides situation awareness and information exchange in a theater of operations. The tactical information broker, Marti, has been integrated with legacy and deployed tactical systems and has been demonstrated and evaluated in multiple flight exercises. The work described herein is ongoing, but shows promise in revolutionizing QoS management for IP multicast in tactical deployments.

Acknowledgment

This research has been sponsored by the U.S. Air Force Research Laboratory.

References

[1] K. Bür, C. Ersoy, “Ad Hoc Quality of Service Multicast Routing,” *Computer Communications*, 29 (2005), pp. 136=148.

[2] K. Bür, C. Ersoy, “Performance Evaluation of a Mesh-Evolving Quality-of-Service-Aware Multicast Routing Protocol for Mobile Ad Hoc Networks,” *Performance Evaluation* 66 (2009), pp. 701-721.

[3] Command and Control Personal Computer (C2PC). GlobalSecurity.org. http://www.globalsecurity.org/intell/library/reports/2001/compendium/c2pc.htm.

[4] A. Corrin. “TIGR keeps troops in the field safer.” *Defense Systems*, March, 2010

[5] J. Cui, L. Lao, M. Faloutsos, M.Gerla, “AQoSM: Scalable QoS Multicast Provisioning in Diff-Serv Networks” Computer Networks by Elsevier, Vol 50/1 pp 80-105, 2005.

[6] R.G. Dewey, J. Bishop, “Common Data Link from Space – Preliminary Lessons from the TacSat-2 Demonstration Program,” *Military Communications Conference*, October 17-20, 2005.

[7] P. Eugster, P. Felber, R. Guerraoui, A-M. Kermarrec., “The Many Faces of Publish/Subscribe,” *ACM Computing Surveys*, Vol. 35, No. 2, June 2003, pp. 114-131.

[8] FalconView Mapping Software Goes Open Source. Dr. Dobb’s Journal. August 18, 2009.

[9] R. Grant, C. Combs, J. Hanna, B. Lipa, and J. Reilly, "Phoenix: SOA Based Information Manage-ment Services," Proceedings of the 2009 SPIE Defense Transformation and Net-Centric Systems Conference, Orlando, Fl, April 2009.

[10] D. Li, J. Wu, Y. Cui, J. Liu, “QoS-Aware Streaming in Overlay Multicast Considering the Selfishness in Construction Action,” *26th IEEE International Conference on Computer Communications*, May 6-12, 2007.

[11] LITENING, Advanced Airborne Targeting and Navigation Pod. Federation of American Scientists Military Analysis Network. October 28, 1999, http://www.fas.org/man/dod-101/sys/smart/litening.htm.

[12] J. Loyall, M. Carvalho, A. Martignoni III, D. Schmidt, A. Sinclair, M. Gillen, J. Edmondson, L. Bunch, D. Corman, “QoS Enabled Dissemination of Managed Information Objects in a Publish-Subscribe-Query Information Broker,” SPIE Conference on Defense Transformation and Net-Centric Systems, April 13-17, 2009.

[13] J. Loyall, A. Sinclair, M. Gillen, M. Carvalho, L. Bunch, A. Martignoni III, M. Marcon. “Quality Of Service In Us Air Force Information Management Systems,” MILCOM, October 18-21, 2009.

[14] I. Mandoiu, A. Olshevsky, A. Zelikovsky, “QoS Multimedia Multicast Routing,” *Handbook of Approximation Algorithms and Metaheuristics*, Chapman & Hall/CRC, May 15, 2007.

[15] A. Patton, “ROVER Adds Extra Set of Eyes to Sky,” *Space War*, August 3, 2006. http://www.spacewar.com/reports/ROVER_Adds_Extra_Set_Of_Eyes_To_Sky_999.html.

[16] F. Pushies. *U.S. Air Force Special Ops (Military Power)*. Zenith Press, June 15, 2007, pp. 116-117.

[17] D. Robbins. Unmanned Aircraft Operational Integration Using MITRE's Cursor on Target. The Edge, MITRE, Volume 10, Number 2, Summer 2007.

[18] W3C, Web Services Architecture, February 11, 2004. http://www.w3.org/TR/ws-arch/.