NAVAL POSTGRADUATE SCHOOL
Monterey, California

Using Multiple Collaborative Agents for Adaptive Quality of Service Management in C4ISR Networks

Student author: LT Raymond Rivera, USN
Naval Postgraduate School
1 University Circle, Code 39
Monterey, California 93943
(831) 656-2772
rarivera@nps.navy.mil

Thesis Advisor: Dr. Alex Bordetsky
Information Systems Academic Group
Naval Postgraduate School
1 University Circle, Code 32
Monterey, California 93943
(831) 656-2264
abordets@nps.navy.mil

6th International C2 Research and Technology Symposium
United States Naval Academy
Annapolis, Maryland
June 2001

Abstract

This research explores the potential of agent technology for adaptive Quality of Service (QoS) management of C4ISR networks. With the growing emphasis on information superiority, any timesavings or additional utilization of resources enabled by effective network management become increasingly important. Intelligent agents are ideal for assessing information, adapting to dynamic conditions, and predicting future network conditions. In the kernel of the proposed multiple agent system (MAS) testbed are agent shared memory and majority rule architectures for agent conflict resolution. The case based reasoning (CBR) technique provides the foundation for building the agents' shared memory of QoS management solutions and allows the individual agents to share their associations of feedback controls in response to application and user QoS profiles. Based on the Telecommunications Management Network (TMN) functionality, we use this agent architecture to effectively translate the warfighter's service layer application requirements across the network. The fundamental frameworks of Service Level Management (SLM) and Policy Based Management (PBM) serve as cornerstones in effectively gathering specific application requirements. Finally, we apply these techniques to an actual C4ISR application at the Pacific Region Network Operating Center (PRNOC) for ongoing research study and agent implementation feasibility studies at the Naval Postgraduate School.

I. INTRODUCTION: C4ISR IN THE 21ST CENTURY

C4ISR networks of the future are increasingly reliant on fast, efficient information exchange over wide distances. In the 21st century, information superiority is the key to battlespace dominance. C4ISR networks are the enablers to this goal and central in the Navy's movement towards Network Centric Warfare. At a minimum, C4ISR networks must be capable of providing voice, video, and data capabilities to the warfighter. At the same time, the information exchange must be accurate, timely, and secure in order to be useful. These factors make the effective management of C4ISR networks paramount. As the growth of information technology increases, so does the need for coordination and maintenance.

The evolution of C4ISR networks and their management systems over the years has resulted in a variety of network management issues. Although all C4ISR networks are required to follow the same basic guidelines and interoperability standards under the Joint Technical Architecture (JTA) and Defense Information Infrastructure Common Operating Environment (DII-COE), there are many different considerations that must be reconciled. These include: diverse services, networks, and technologies; multiple vendor equipment; loosely organized management applications; multiple management protocols; and multiple data representations [1]. C4ISR networks must be capable of adapting end-to-end resources and Quality of Service (QoS) across heterogeneous, and oftentimes, mobile networks.

In general, management of these networks occurs at Network Operations Centers (NOC's). NOC's utilize standard network monitoring approaches like Simple Network Management Protocol (SNMP) and Common Management Information Protocol (CMIP) to monitor, test, and evaluate network parameters including traffic patterns, bandwidth utilization, network response times, and e-mail response times. Unfortunately, with increasing requirements for fast information exchange, these techniques need improvement and adaptive management capability.

Adaptive management capability of C4ISR networks could be achieved through the usage of multiple collaborative, intelligent agents to overcome the nominal deficiencies in C4ISR network management. Although agent technology is relatively new, it has already demonstrated exciting potential in a variety of applications that lend themselves to this research. Basic agent characteristics of *autonomy*, *adaptability*, *scalability*, and *co-operability* allow the sharing of information over the entire span of the network. Intelligent agents assess information, adapt to existing conditions, predict future network conditions, and advise on anticipated future conditions. With multiple, collaborative agents, knowledge and expertise can be shared, eliminating the need to store all knowledge locally. In the context of a dynamic environment with unique application profiles, this framework is ideally suited for translating the warfighter's service level requirements. The end result is a more efficient, responsive, and potent C4ISR network.

In the kernel of the proposed multiple agent adaptive management testbed are agent shared memory and majority rule architectures for agent conflict resolution. The case-based reasoning (CBR) technique will be used as the foundation for building the agents' shared memory of QoS management solutions. It allows the individual agents to share their associations of feedback controls in response to application and user QoS profiles.

The committee type multi-participant group decision support technique will be adopted for resolving the conflicts among multiple agents in allocating the networking resources in response to the conflicting QoS requirements. The conflict resolution architecture is composed of an artificial neural network (ANN) with two hidden layers. Each node in the second hidden layer represents the committee solution for QoS resource allocation that the multiple agent system (MAS) learned while managing the C4ISR task and adapting to the conflicting QoS requirements.

In accordance with the Telecommunications Management Network (TMN) functionality, the agent architecture effectively translates the warfighter's service layer application requirements across the network. The fundamental frameworks of Service Level Management (SLM) and Policy Based Management (PBM) are used to effectively gather the specific application requirements. From these requirements, the multiple agent testbed becomes the enabling framework for the intelligent adaptive capability of collaborative work.

Using these building blocks for our research, we investigate an actual C4I application at the Pacific Region Network Operating Center (PRNOC) and use it for ongoing modeling and simulation research at the Naval Postgraduate School. In this instance, we are investigating the adaptive allocation of bandwidth under dynamic conditions via multiple collaborative agents.

A. Organization of Remaining Sections

Above, we introduced the basic context of this research, that is, adaptive QoS management of C4ISR networks in the 21st century. In Section II, we investigate the tenets behind agent technology and introduce the proposed agent architecture for this research. Section III is a review of adaptive QoS management. Section IV is a continuing discussion of our proposed architecture, and the application of these concepts in an actual C4I application. Section V contains our concluding notes and introduces ongoing work at the Naval Postgraduate School.

II. AGENT TECHNOLOGY

Agent based technology is an interdisciplinary area of research that started receiving special attention from the research community in the early 1990's. This technology holds exciting potential for the artificial intelligence (AI) and computer science communities because of its ability to reach a broad range of applications across many industries. To reach this potential, there are also many challenging problems including security, resource consumption, complexity, and the degree of trust in agents to do exactly what is desired. While these challenges are real, they are not enough to dampen the spread of the agent paradigm. Researchers are continually developing innovative new approaches and applications for agent technology.

From DoD's perspective, agent technology is expected to help reduce time spent manipulating stovepipe command and control (C2) systems, make it easier to assemble future systems, improve interoperability, reduce system complexity, and help solve data blizzard and information starvation problems [2]. Agent applications range from robotics to information retrieval to e-commerce to network management and telecommunications. Based on this versatility, it is easy to envision agent technology being applied in the area of adaptive QoS management.

A. What is an intelligent agent?

In general, intelligent software agents are a relatively new class of software that act on behalf of the user to find and filter information, negotiate for services, easily automate complex tasks, or collaborate with other software agents to solve complex problems. The main idea behind software agents is *delegation*, whereby the user delegates a task to the agent. In turn, agents act *autonomously* to perform the task on behalf of the user. In order to facilitate task accomplishment, *communication* is an important interface between user-to-agent and agent-to-agent. Finally, the agents must be able to *monitor* the state of their environment and make the decisions necessary to complete their tasks. [3]

When working with agent technology, the first order of business is effectively localizing the meaning of the term "agent," for there are literally hundreds of definitions and contexts. The term agent is highly overused and can mean different things to different applications. For instance, in network management, there are SNMP and CMIP agents, but these are really nothing more than servers providing data to their clients. On the other hand, there are expert systems with huge knowledge bases, which are also considered agents because of their intelligent behavior. This work focuses on the latter type of agent that intelligently makes decisions. Ultimately, these agents interface with the SNMP/CMIP agent functionality only as the abstraction of higher service level requirements to lower network level requirements.

In general, the following basic definition of agent applies to this research: "A computational entity that acts on behalf of others; is *autonomous*, *adaptive*, and *intelligent*; and exhibits the ability to learn and cooperate (*collaborate*) " [1]. More advanced agents may also have other attributes, such as *mobility* (allowing migration from host to host) and *personality* (manifesting some human qualities such as cooperation, caution, and greed). These additional characteristics can be explored as possible enhancements or alternatives to the research.

B. Agent Topology

As for the classification of agents, the range of methods to develop a standard topology is highly varied. One prevalent method of classifying agents is in terms of dimensions [4]. Certainly agents cannot only be described in just two or three ways because of the variability of the term and the need to accurately distinguish one agent from the next. Following, agents can first be classified by their *mobility*, i.e., by their ability to move around some network. Thus, they may be classified as *static* or *mobile*. Second, agents can be classified by the presence of a symbolic reasoning model, as either *deliberative* or *reactive*. Deliberative agents engage in planning and negotiation with other agents to achieve goals, while reactive agents respond to the present state of the environment in which they are a part. Third, agents can be classified by the exhibition of ideal and primary attributes such as *autonomy, learning*, and *cooperation* to derive the following four types of agents: *collaborative, collaborative learning, interface,* and *truly smart* agents (Figure 1). Fourth, agents may be classified according to their roles such as *information* or *Internet* agents. Fifth, agents can be classified as *hybrid* if they combine two or more agent philosophies in a single agent. Lastly, agents may exhibit any of a wide range of secondary attributes. In sum, just as the means of defining agents is diverse, so are the methods of classifying them.

Error! Unknown switch argument.

Figure 1. Topology based on primary attribute dimension. [4].

C. Why Multiple?

There are many reasons why a multi-agent approach is more advantageous than a single agent approach. First of all, the management of C4ISR networks is too large a problem for a single centralized agent. There are resource limitations and robustness concerns in only using a single agent. Decentralization takes away the possibility of a single point of failure. Moreover, dividing functionality among agents provides modularity, flexibility, modifiability, and extensibility [5]. Second, multiple agents allow for the interconnection of multiple existing legacy systems, which can be especially helpful in DoD. By building an agent wrapper around such systems, they can be incorporated into an agent society. Third, multiple agents improve scalability due to the organizational structure of the agents, which allows them to dynamically change to reflect the dynamic environment. Fourth, multiple agents provide solutions for inherently distributed problems by drawing from distributed information sources and distributing the expertise. For these reasons, multi-agent systems are more prevalent than single agent systems. [4]

D. Why Collaborative?

The collaborative behavior criterion for intelligent agents coincides with social ability. By *collaborative*, the usage of a multiple agent system is implied. Collaborative agents work in concert with other agents to achieve a common goal. The rationale for having collaborative agent systems is a specification of the goal of distributed artificial intelligence (DAI). It may be stated as: "creating a system that interconnects separately developed collaborative agents, thus enabling the ensemble to function beyond the capabilities of any of its members" [4]. The criterion of "collaborative" goes hand in hand with "multiple" in that it dictates teamwork among the agents. Agents cannot be collaborative without other agents to collaborate with. In other words, the union of the two characteristics is integral to the accomplishment of the factors listed above.

E. Why Adaptive?

An agent is considered *adaptive* if it is capable of responding to other agents and/or its environment to some degree. At a minimum, the agent must be able to react to a certain stimulus. For this research, adaptive also means the ability to reason, learn, and evolve. These agents are deliberative and can change their behavior based on experience and a dynamic environment. Learning techniques include artificial neural networks, Bayesian rules, credit assignments, classifier rules, and case based reasoning. Adaptive agents can be *passive*, whereby they respond to environmental changes without attempting to change the environment; or *active*, whereby they exert some influence on the environment to improve their ability to adapt.

Unfortunately, by providing agents with the capability to adapt, there is also a possibility of inducing undesirable side effects – particularly in situations where global system behavior may be significantly affected by a minor local change [6]. An adaptive agent must be able to adapt to unforeseen conditions, have a reasonable amount of behavioral assurance, and be able to respond in a timely manner. When developing adaptive agents, one must consider the tradeoff between verification of proper agent coordination and speed. If the agents cannot act in a fast enough manner, this obviously defeats the purpose of having them. Despite this conundrum, the characteristic of adaptability remains integrally important in allowing the agents the ability to respond and thrive in dynamic environments.

F. Case Based Reasoning

In the kernel of the proposed intelligent support architecture is the layered model of case memory. Case memory is useful in that it supports the discovery of pertinent collaborators, the retrieval of information pertinent to collaboration, and the creation of conventions among individuals by utilizing the CBR technique for indexing, capturing, and retrieving collaborative objects [7].

As a source of comparison, the logic behind CBR usage is similar to the usage of case law in the legal domain. In this domain, case studies are used as a point of reference. Lawyers and judges examine pre-existing case law to determine applicability to current cases at hand. Of course, not every new case is exactly like an old one, but the advantages of being able to apply prior work and experience to a new situation are clear. Not having to "reinvent the wheel" every time alleviates the amount of work to be done, while simultaneously giving higher credence to the ultimately decided outcome of the case.

The general architecture for CBR illustrates the evolutionary nature of the case library. In the *retrieve* stage, case law is injected into the process as an initial step in determining similarity with the current input. Next, in the *adaptation* stage, the system attempts to reconcile case memory with the new situation. *Execution* follows and the case library is updated with the new method in the *organization* phase. In this manner, the knowledge base is continually updated. [8]

G. Agent Communication

Communication is the backbone to any agent system because it allows agents to share information and thereby determine the overall behavior and organization of the system. Agent communication is accomplished with three components: *ontology*, *content language*, and *agent communication language* [1]. *Ontology* is a collection of terms and rules that define, govern and localize a certain domain. The *content language* is used for information encoding through statements about the domain, which combine terms from the corresponding ontology into meaningful sentences. The *agent communication language* (ACL) provides formalism for exchanging messages.

Currently, agent communication is one of the most important areas for standardization. The Object Management Group (OMG) is one agency attempting to ensure the variety of communication languages is kept at a minimum. Messages must have a well-defined semantics that is computational and visible. Therefore, ACL's are required for interoperability. ACL's must have formal semantics so that different implementations preserve the essential features. Possible implementations include:

- Knowledge Query Manipulation Language (KQML).
- Foundation for Intelligent Physical Agents (FIPA) ACL.
- Knowledge Interchange Format (KIF).
- XML-based

There are two standards regarding agent-based systems: FIPA ACL and OMG's Mobile Agent System Interoperability Facilities (MASIF). The interactive nature of multi-agent systems drives the need to support interoperability between agents from various sources. Moreover, the development of such a standard is necessary for the successful utilization of agent technology in an open environment.

H. Agent Architecture

In practice, the collaborative multiple agent architecture will be used in conjunction with network operations management teams decision support relationships. Therefore, we consider the perspective collaborative multiple agent structures using the multi-participant information processing and networking paradigm [7]. In accordance with this paradigm, decision-making relationships can take place locally or span across vertical and horizontal organizational boundaries. Standard network computing topologies can be applied to derive the three basic models of *group, team*, and *committee*. [7].

In the *group* model (Figure 2), the structure of information flows is a mesh network. It links multiple decision-makers in a way that allows complete interaction among them.

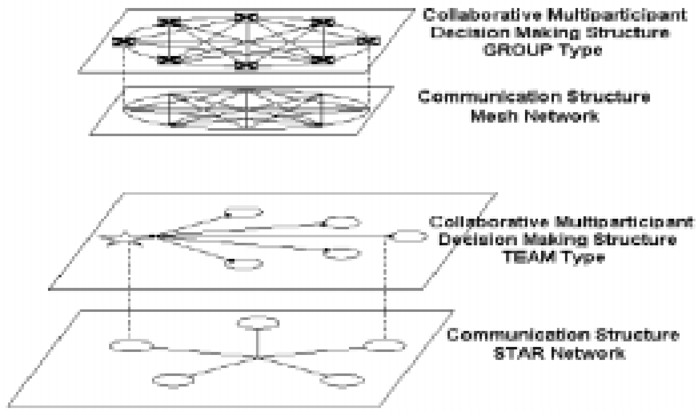

Figure 2. Group and Team type multi-participant structures. [7]

The *team* model represents a more centralized pattern of a single decision-maker with no participant interaction. Several local area and wide area communication topologies could satisfy the team structure support requirements. The primary topology is generally star and fits local and interdepartmental relationships. Also, bus and ring provide chain and circle type relationships to the team members.

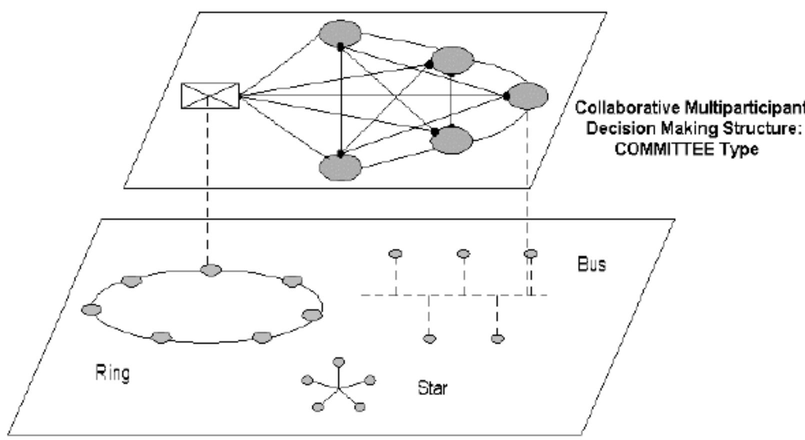

Figure 3. Committee structure. [7]

The third basic model is *committee* (Figure 3) and is composed of multiple levels. This model combines a single decision-maker on the first level with the complete participant interaction on the next. In turn, this allows collective behavior that is based on the different types of majority rules or consensus protocols. On the second level, a combination of star and ring topologies could be used to support local and interdepartmental committee structures.

To summarize, group multi-participant structures may not be the most appropriate prototype for the multiple agent adaptation since it relies on the mesh topology and does not separate facilitator (coordinator) from the other members. Unlike it, team topology naturally allocates a role for the decision-maker (facilitator), but lacks cooperative relationships among the members, which is critical in the joint knowledge discovery process. For these reasons, the committee model represents the best compromise between the group and team multi-participant structures. In other words, it allows a facilitator (coordinator) role, while at the same time compensating for the lack of participants' interaction that is typical for the team structure.

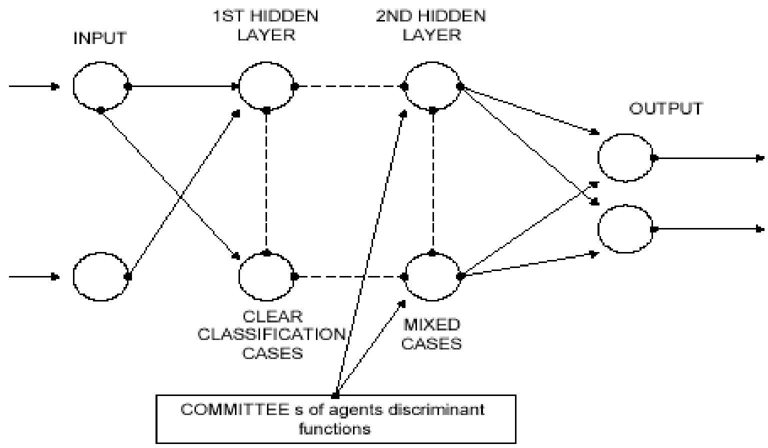

Figure 4. Representing the agents committees by Artificial Neural Network. [Ref. 7]

With respect to adaptive QoS, the agents committees will be implemented in a four-layer artificial neural network as shown in Figure 4. The layers consist of an input layer, first hidden layer, second hidden layer, and output. The first hidden layer of agents resolves relatively easy cases to allow for network bandwidth adaptation without any contradiction. The second hidden layer resolves more challenging cases. In this layer, the selection criteria for the committee of constraints may vary. When considering factors that are all considered equal, the selection criterion is a simple majority rule. The learning process will compare the new problem with the set of developed (learned) empirical constraints that represent the network layer bandwidth adaptation experience (case memory).

III. FUNDAMENTALS OF ADAPTIVE QOS MANAGEMENT

In accordance with user profiles, intelligent agents adapt to a dynamic environment by utilizing network resources and channels to translate the user's desires across the network. The agents bridge the interface between the user's service level requirements and the network management requirements in the TMN framework. We follow a systems level analysis methodology to develop SLM techniques in capturing application requirements between users and service providers.

A. Service Level Management (SLM)

Multiple collaborative agents are ideally suited to match the service level requirements of the warfighter in a transparent manner. Service Level Management (SLM) refers to "the process of negotiation, service level agreement (SLA) articulation, checks and balances, and reviews between the supplier (NOC) and consumer (warfighter) with respect to the services and service levels that support the consumer's business practices" [9]. In other words, SLM provides a formal method for optimizing the C4ISR network; that is, by best meshing the desires of the warfighter with the capabilities of the network service provider (NOC).

Learning and understanding the needs of the user (warfighter) is the first step in SLM, which is not necessarily as easy as it can seem. The warfighter and network manager have a different language when discussing requirements. Moreover, the two camps have different perspectives in how to map the well being of elements in the infrastructure into the well being of the services. The differences can be summarized as follows:

- Parameters that are easy to understand and measure for network specialists do not translate well into parameters that are easily understood by ordinary customers.
- Parameters that are easily understood by customers are not easy for network specialists to measure.

This disparity is known as the "Semantic Disparity Problem." Overcoming it is generally recognized as the crux of SLM. [10]

B. Gathering Requirements

To develop application requirements, we follow a systems level analysis methodology because of its compatibility with SLM. By understanding the network in terms of levels, we can better distinguish the specific QoS needs of the user and understand the inter-relationships of the various network components with respect to QoS. Requirements add to each other, such that application requirements add to user requirements, host requirements add to application requirements, and all add to network requirements. As a result, requirements filter down from user to application to host, resulting in a service request that is a set of service requirements, or service levels, to the network that correspond to different levels of the TMN layer architecture. This results in a service offering that is end-to-end, consisting of service requirements that are configured in each element (e.g., router, bridge, circuit, etc). [11]

C. Quality of Service (QoS)

The challenge of network management is to consistently deliver high levels of performance to the user. This has become increasingly difficult due to higher bandwidth requirements for applications and the unpredictable nature of application deployment. As a result, QoS can fluctuate from day to day. At a NOC, this can be due to ships deploying, contingency operations, environmental considerations, or system degradation.

In broad terms, the QoS of a wide area network (WAN) is a measure of how well it does its job, i.e., how quickly and reliably it transfers various kinds of data, including voice, video, and data, from source to destination. Since there are so many kinds of communications traffic, there is more than one set of criteria to satisfy. Technically, QoS refers to a combination of various factors that collectively describe system performance networks. In general, the following are usually recognized as highly important [12]:

1. Availability

Ideally, a network is available 100 percent of the time, but this is obviously not always the case. Even so high a figure as 99.8 percent translates to about one and half down hours per month, which can be definite concern for the tactical warfighter [12].

2. Throughput

Throughput is the effective data transfer rate measured in bits per second. Throughput is not synonymous with bandwidth, which is merely the size of the pipe. In contrast, throughput takes into account such factors as number of users, bit overhead for identification or other purposes, and line degradation.

3. Packet loss

Network devices, such as switches or buffers, sometimes have to hold data packets in buffered queues due to congestion. If the link remains congested for too long, the buffered queues overflow resulting in packet loss. In turn, the lost packets must be re-transmitted resulting in a longer total transmission time.

4. Latency

Latency is delay introduced in application traffic flowing across a network path due to queuing, processing, or congestion. Other sources of delay include propagation, transmission, routing, and satellite propagation. From an application service perspective, optimizing the total end-to-end delay is more important than individual sources of delay.

5. Jitter

Jitter is the distortion of the inter-packet arrival times compared to the inter-packet times of the original transmission (i.e. delay variance). Causes include variations in queue length, variations in the processing time needed to reorder packets that arrived out of order due to different paths, and variations in the processing time needed to reassemble packets that were segmented by the source before being transmitted [12]. Jitter is particularly demanding to multi-media applications.

Applications differ in the way they use bandwidth and their QoS requirements. For example, a long transfer file needs a high throughput and low packet loss, but is not very sensitive to delay and jitter. Live videoconferencing also needs high throughput, but is sensitive to both delay and jitter. The unpredictable mix of applications running on a dynamic network and the conflicts that occur due to simultaneous application requirements induces QoS problems. This is the fundamental dilemma for QoS resource management and the driving impetus behind using intelligent agents. "Throwing bandwidth at the problem" is not sufficient in itself to guarantee that specific applications will perform adequately under all traffic conditions. The bandwidth must be intelligently managed to prioritize application requirements and business priorities.

D. Policy Based Management (PBM)

Meeting QoS requirements under dynamic conditions can be tied to Policy Based Management (PBM). Policy based management is defined as "the combination of rules and services where rules define the criteria for resource access and usage" [13]. Instead of getting involved in the details of queuing mechanisms and configuring routers and switches, PBM allows the network manager to simply define a policy that might say, "give my SAP application guaranteed bandwidth and the highest priority."

As in SLM, PBM is accomplished via the Service Level Agreement (SLA). Ideal in concept, but difficult in reality, SLA's help the service provider and user to work together to establish specific expectations. The SLA's help translate the service layer requirements into the network management layer requirements, i.e. meet the SLM paradigm. PBM is a cornerstone for programming the agents.

E. Telecommunications Management Network (TMN)

First introduced in the mid-1980's, TMN has become the globally accepted framework for the management of telecommunications networks. For the most part, it is described in International Telecommunications Union – Telecommunication Standardization Sector (ITU-T) and other standards. The functional architecture of TMN is termed the logical layered architecture. It essentially categorizes the OSI management functionality layers as shown in Figure 6.

The use of the term layer recognizes an implicit support hierarchy among the functionality. However, the architecture does not allow communications between non-adjacent layers. Higher-level layers are viewed as having a higher level of information abstraction compared to lower layers. From this perspective, network management functionality is viewed as more vendor-independent than element management, while service management functionality is viewed as more technology independent than network management. [14]

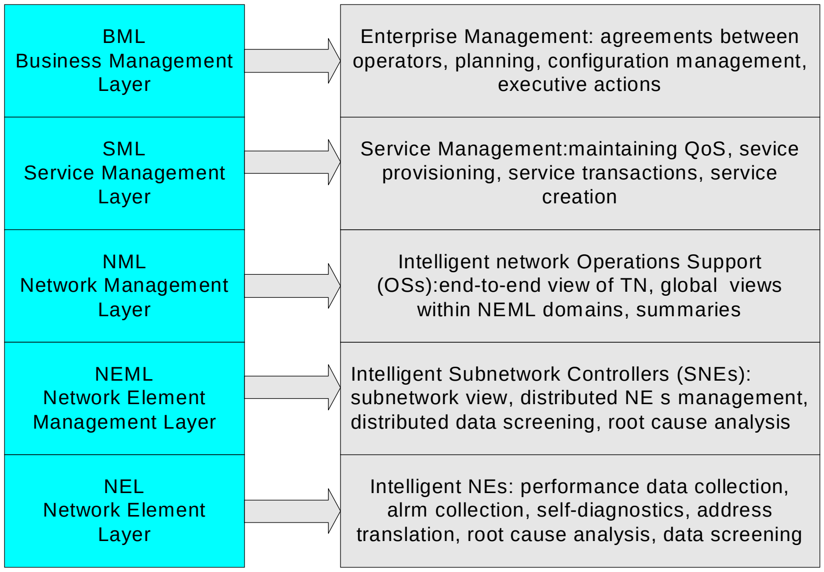

Figure 6. Telecommunications Management Network. [7]

IV. A PROPOSED MODEL FOR ADAPTIVE QOS MANAGEMENT

A. Layers of Feedback Control: Individual Agent Adaptation

Having established the underlying principles behind adaptive QoS management, we continue our development of the proposed agent framework from Section II. First, we tie the Telecommunications Management Network (TMN)/ Service Level Management functionality to the fundamental concept of system coordination to address the problems of agent adaptation. By doing so, this allows the identification of critical relationships through associated feedback controls [7]. From this perspective, the process of adaptive control and coordination in a multi-agent architecture can be based on the idea of mapping feedback control relationships into an agent's shared awareness memory, where feedback controls are delivered via agents-facilitators. In turn, this functionality is expanded into the agents' integration with case memory.

Unfortunately, real-time applications such as audio/video conferencing and shared application control have strict requirements in terms of delay and bandwidth as discussed earlier. While asynchronous applications need only to adapt naturally via changes in response time, real-time applications must reduce the quality of the data stream to meet reduced bandwidth needs. Adding to this, when multiple applications run simultaneously, lower-priority applications may be required to adapt to lower bandwidth usage or even be switched off entirely to free up bandwidth for higher priority applications.

Two layers of feedback control *Call Preparation Control (CPC)* and *Connection Control (CC)* can be considered to support multiple applications. Call preparation Control integrates feedback gathered from previous conferencing sessions to make informed decisions regarding connection setup and bandwidth tradeoff in future sessions. Its adaptation is long-term and mainly associated with the allocation of resources for the entire length of a multimedia call. Connection Control reflects ongoing performance measurement and adaptation throughout the length of the call. Its adaptation is short-term, such as may be required during a single call. The requirements of both layers of feedback control are summarized below [7].

1. **Call Preparation Control requirements**
   - A call must establish, modify, and execute voice, video, and multi-media application sharing communication between multiple users.
   - A call must involve coordination between parties to satisfy response time, bandwidth, and other QoS requirements.
   - A call contains relationships between user profiles, media, and system resources that may be dynamically modified during a call.
   - Each user can request resources individually.
   - A call will allow negotiations between different sites for system resources.
2. **Connection Control requirements**
   - Provided QoS parameters must be supervised.
   - Flow control, congestion control, routing, reservation, and re-negotiation of resources must be provided for.
   - Connections are modified and released.

B. Call Preparation Adaptation: Service Layer Feedback Controls

The proposed agent architecture can now be fully represented by the following components: (1) case based reasoning memory, (2) agents-facilitators, and (3) collaborative feedback controls. The layers of case memory are structured according to the following feedback control relationship for a web conferencing service:

SLM event (t) = **{U(t), X(t), P(t), I(t)}**,

where:
SLM event = Service Level Management event; **U(t)** is a set of *user input controls* (desktop conference calls, links to knowledge sources); **X(t)** is a set of *SLM process state variables* (QoS restraints such as response time and bandwidth); **P(t)** is a set of *service process outputs*; and **I(t)** describes the *environmental impact* to the service management process.

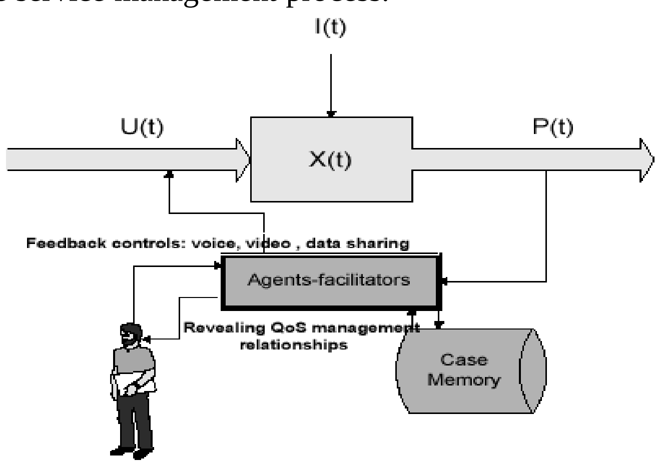

Figure 7. Feedback control model for individual agent adaptation. [7]

The memory architecture of agents-facilitators is layered and divided into *bridge* or *router agents* operating with different combinations of feedback control layers. Objects such as individual profiles of collaborators, QoS indices for multimedia streams and timely events, problem solving task profiles, and other collaboration objects form different layers of case frame representation. The other segments, for example, collaborator profiles, are mainly populated during adjustment interactions with agent-facilitators. The layers are structured according to the feedback concept for multimedia multi-point conferencing awareness. Under this concept, user access to case memory is provided by agent-facilitators, which enable collaborators to communicate via desktop video conferencing and shared applications at different levels of bridges, routers, and gateways, depending on which segments of case memory are involved. [7]

The router agent plays a major role in providing feedback controls and adaptation in service management. First of all, it provides user memory transactions by capturing the necessary information to support personal, document, and task profiles. Second, it helps locate appropriate human sources of knowledge and manage desktop video conferencing calls to selected experts. Lastly, it provides training and capturing of QoS management knowledge in case memory. Figure 7 illustrates the feedback control association of service process outputs and SLM process state variables with user input controls into the case memory. [7]

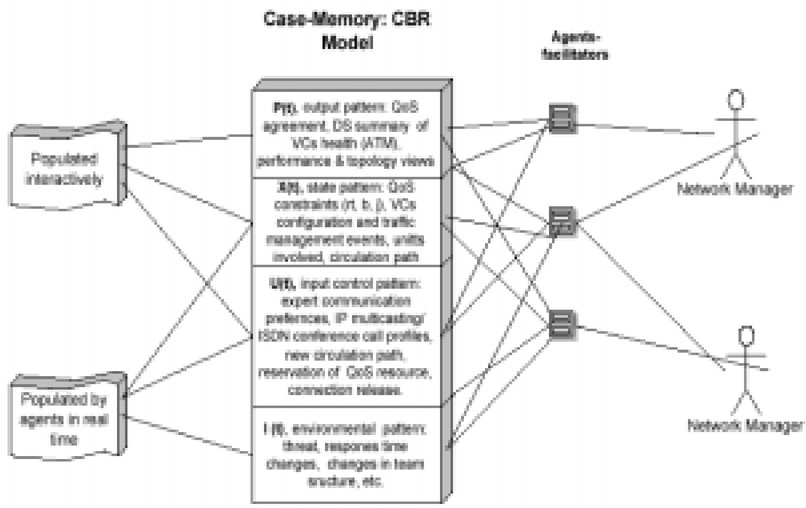

Figure 8. Feedback Control Association. [7]

Figure 8 represents the knowledge retrieval model, in which each interface between layers from the bottom-up is an association based on the underlying levels. The content profiles and user response time requirements are captured in real time and populate the lower segment of the case memory stack. The agents capture the sequence of application calls (content profile) with corresponding time stamps and convert them into response time and bandwidth requirements that populate the QoS segment of a case memory frame. [7]

In general, the QoS constraints associated with a specific SLM event are comprised of boundaries that define preferred bandwidth for voice, video, white board, and application sharing. According to such a profile, each conferencing node has associated voice, video, whiteboard, and/or application sharing delivery trees. Switching among these delivery trees helps to satisfy otherwise infeasible response time requirements. The rules for switching delivery trees can vary based on the system, such as operational heuristics, for example. Each SLM event has a corresponding set of rules that is associated with the QoS segment of the agents' case memory. The Router Agent reads the QoS segment of the feedback control association {service process outputs, **P(t)**; SLM process state variables, **X(t)**; user input controls, **U(t)**} from the case memory and coordinates the delivery tree switching (i.e. bandwidth allocation solutions) with the other agents-facilitators. When coordination is done, the agents transfer the coordinated solution to the network layer connection control.

C. *Connection Control Adaptation: NE Layer Adaptation*

In general the Connection Control requirements include:

- Supervising QoS parameters,
- Providing flow control, congestion control, routing, reservation and renegotiation of services,
- Modifying and releasing connections, and
- Notifying applications to allow them to adapt.

As opposed to Call Preparation Control, in which decisions are made *before* the call is made, Connection Control is done on an ongoing basis throughout the duration of the call. Feedback regarding network conditions is continuously collected and processed to allow the applications in use to adapt. Being that the most dynamic network resource is allocated channel bandwidth, this becomes the targeted area for network layer feedback controls.

D. *Integration of Service Management and Network Management Layer Adaptation*

Based on the example from the previous section, there are several scenarios where Service Management Layer (Call Preparation Control) adaptation output would be useful:

- Specifying the initial number of multicast groups to which to subscribe.
- Specifying the number of consecutive report intervals, which should trigger adaptation.
- Specifying the levels of loss, which are significant, both for indicating congestion in the network and the absence of congestion.

Since most multipoint conferences consist of many components including audio, video, and shared application control, it is necessary to balance the bandwidth needs of each individual tool. Accordingly, video streams may be constrained to black and white images in favor of high quality audio or lower priority streams may be shut off entirely in favor of higher priority streams.

IV. DEVELOPING AN AGENT APPLICATION

We apply this framework as a possible bandwidth allocation solution for the Pacific Region Network Operating Center (PRNOC) in Wahiawa, Hawaii. Bandwidth for the ships is generally constrained by satellite and component limitations for each ship. In general, the Fleet NOC's are the fleet portals to the Defense Information Service Network (DISN), both unclassified (NIPRNET) and classified (SIPRNET) and provide firewall, mail store and forward, web caching and other network services. Primarily, the NOC is interested in ADNS RF for bandwidth management. Unmanaged, all service connections, whether critical or not, compete evenly for the available bandwidth.

The Automated Digital Networking System (ADNS) is a unique system in which the basic backbone is the classified network (GENSER). To obtain UNCLAS or TACINTEL, encryption devices are used to encrypt the information to get to the baseline GENSER and decrypt at the other end. The bandwidth is shared among the three classifications (as opposed to separately allocated bandwidths). Certain minimum bandwidths can be guaranteed for each level. However, bandwidth management cannot be solely considered from one level, but in totality of all three levels. Thus, if in the future there is an increase in regular GENSER traffic, adjustments to bandwidth allocation may need to be made to both the GENSER and UNCLAS sides. In this problem, there are numerous levels of QoS management that can be intelligently managed. As part of our initial study, we investigate the intelligent bandwidth of one portion of this pipe (UNCLAS), within which are numerous applications that compete for bandwidth. By optimizing the utilization of UNCLAS, we can limit the amount of additional bandwidth that may interfere with the GENSER and TACINTEL.

At present, PRNOC uses Packeteer's Packetshaper as its network management tool. It develops policies based on operational heuristics and best guesses derived from historical patterns. It is here that we investigate using an agent framework. Intelligent agents are a natural fit to work hand-in-hand with the Packetshaper to automate the process; that is, the agents implement policy by dynamically allocating bandwidth, while the Packetshaper actually carries it out. Various agent committees can be implemented with different voting patterns depending on the situation and priorities of the ship, i.e., wartime, peace, deployment. In this manner, the most appropriate applications have the highest priorities in accordance with the situation. [15]

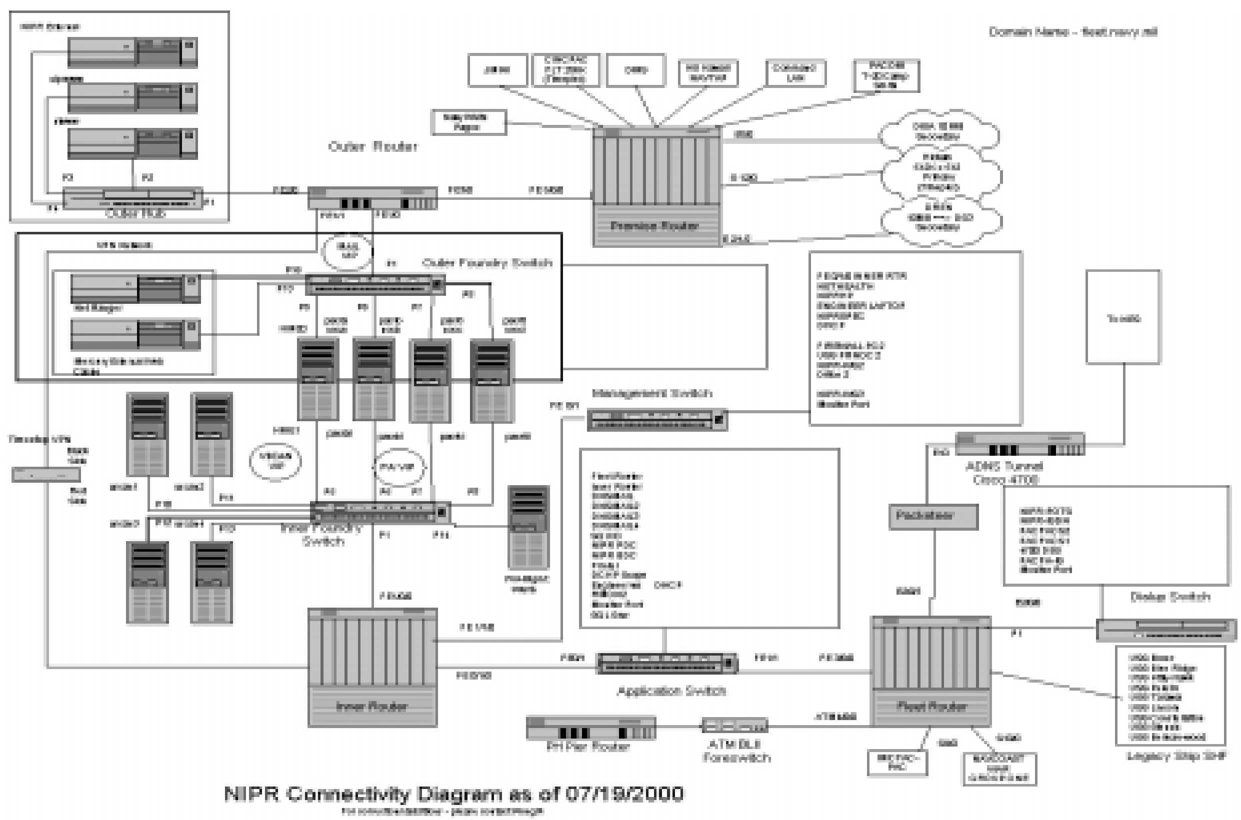

Figure 9. PRNOC NIPRNET configuration and Packeteer placement.

V. CONCLUSION

Agent technology represents a potential breakthrough in the adaptive QoS management of C4ISR networks. With the spread of the agent paradigm, new developments continuously aid research in our particular area of research. Utilizing the agent paradigm at PRNOC is just one example of how intelligent agents can aid in dynamic environments. Moreover, the proposed agent framework is just one way to approach this problem and many others like the ZEUS agent framework [16], IMPACT [17], RETSINA [18], and others could provide alternative methods. Finally, the agent decision cycle can be modeled with the Extend modeling program from Imagine that, Inc. With the growing emphasis on information superiority in the 21st century, agent technology for adaptive QoS management can be a tremendous aid to our C4ISR capability.

References:

[1] Bieszczad, A., Biswas, P., Buga, W., Malek, M., Tan, H. "Management of Heterogeneous Networks with Intelligent Agents." Lucent Technologies, Inc., Bell Labs Technical Journal, October-December 1999, pp. 109-135.

[2] Manola, F., Thompson, C. "Characterizing the Agent Grid." Object Services and Consulting, Inc., June 1999.

[3] Agent Builder, An Integrated Toolkit for Constructing Intelligent Software Agents. Reticular Systems, Inc., 1999. http: //www.agentbuilder.com/

[4] Nwana, H. "Software Agents: An Overview." Knowledge Engineering Review, Vol. 11, No. 3, pp1-40, September 1996.

[5] Agent Technology Green Paper, Version 0.92. Object Management Group, 25 April 2000.

[6] Gordon, D. "APT Agents: Agents That Are Adaptive, Predictable and Timely." In the Proceedings of the First Goddard Workshop on Formal Approaches to Agent-Based Systems (FAABS 2000).

[7] Bordetsky, A. "Adaptive QoS Management via Multiple Collaborative Agents." Agent Technology for Communication Infrastructures, Chapter 4. Hayzelden, A. and Bourne, R. Wiley: 2000.

[8] Lewis, L. Managing Computer Networks: A Case Based Reasoning Approach. Artech House, 1995.

[9] Lewis, L. "Spectrum Service Level Management Definition, Offerings, and Strategy." Cabletron Systems, March 1998.

[10] Lewis, L. White Paper: Service Level Management with Spectrum. Aprisma, December 2000.

[11] McCabe, J. Practical Computer Network Analysis and Design. Morgan Kaufmann Publishers, Inc. San Francisco: 1998.

[12] Dutta-Roy, A. "The Cost of Quality in Internet-style Networks." IEEE Spectrum, September 2000, pp. 57-62.

[13] Vicente, J., Cartmill, H., Maxson, G., Siegel, S., Fenger, R. "Managing Enhanced Network Services: A Pragmatic View of Policy-Based Management. Intel Technology Journal, Q1 2000.

[14] Sidor, D. "TMN Standards: Satisfying Today's Needs While Preparing for Tomorrow." IEEE Communications Magazine, March 1998, pp. 54-64.

[15] Stephenson, B., Lee, W. Concept of Operations for Fleet NOC Application of Packeteer Packetshapers. Pacific Region Network Operations Center, Wahiawa, Hawaii.

[16] The Zeus Agent Building Toolkit. Zeus Technical Manual, British Telecommunications Labs, September 1999.

[17] Bigham, J., Cuthbert, L., Hayzelden, A., Luo, Z. "Multi-Agent System for Network Resource Management." Department of Electronic Engineering, Queen Mary and Westfield College, University of London, 1999.

[18] http://www.cs.cmu.edu/~softagents/retsina/