2006-09

Systematic Adaptation to Network Resource Constraints in Coalition C2 Environments

Clement, Michael R.; Bordetsky, Alex

11TH ICCRTS
COALITION COMMAND AND CONTROL IN THE NETWORKED ERA

Systematic Adaptation to Network Resource Constraints in Coalition C2 Environments

Author: Michael R. Clement (STUDENT)
Advisor: Dr. Alex Bordetsky

Michael R. Clement
Naval Postgraduate School
589 Dyer Road, Room 200A, Monterey, CA 93943 USA
(831) 656-1804
mrclemen@nps.edu

Abstract

Tactical environments exhibit the scarcest communication resources of any military setting. However, emerging C2 systems utilize data services demanding significantly higher bandwidth and lower latency from tactical networks than ever before. The heterogeneous nature of C2 systems and data services within coalition environments raises the level of complexity even further. This paper takes a systems approach to adaptive network resource management, focusing on techniques that enable application networks to regulate themselves according to available connectivity and cooperative prioritization of data services. Services that would fail under connectivity constraints in traditional networks are able in this model to degrade gracefully while maintaining usable levels of functionality. Schemas for both mandated and cooperative data prioritization are discussed. The example of the USSOCOM-NPS Field Experimentation Program is used to illustrate the range of potential applications of this technique to coalition C2 operations.

I. Introduction

In the tactical environment, data networks are limited by the constraints of wireless communication and portability, yet the demand for real-time multimedia data services is ever-growing. With the emergence of blue force tracking, network-based unmanned air vehicle control, and video and voice teleconferencing into the field, even reasonably capable networks have become taxed to deliver information in a timely and reliable manner. Certain classes of networks are in a further constrained state: wireless networks have to address connectivity and user mobility issues; mesh networks utilize the same links to provide end-node connectivity to some users and a backhaul to other users, doubly taxing some links. A schema is necessary to optimize the flow of information across the tactical network; however, a blanket policy is not sufficient in a dynamic, mobile environment where network resources come and go, and the data requirements change over time.

Existing methods for network resource management, collectively known as Quality of Service (QoS) technology, generally use a fixed set of traffic classifiers that specify end-to-end or per-hop behavior for individual packets [1]. These classifiers provide some distinction between types of traffic, offering a degree of prioritization. However, they fall down at the point of articulating end-user requirements or adapting to changes in available network resources. Some QoS technologies assign static resource reservations for the lifetime of the traffic flow and most are unable to adapt to changing network resources. Nearly all methods rely on a small set of classifiers or network parameters that define traffic requirements. In a complex coalition forces environment, the number of users and types of services quickly become too large for static classifiers to adequately describe and require a way to dynamically assess and adjust the network in the absence of an overarching static prioritization schema.

Proposed approaches for adaptive network resource management, also referred to here as adaptive networking, aim to address these issues by focusing on techniques for dynamic resource negotiation and traffic adaptation [2][3][4][5][6][7][8][9]. Rather than relying on a static set of classifiers, adaptive networks base their behavior on general statements of end-user requirements combined with the overall network state. Adaptive networks provide mechanisms for negotiating between the application and the network for available resources, and dynamically adjusting service levels as available network resources change over time.

However, some important problems remain unaddressed. Requests for data services are still treated atomically; that is, each application traffic flow is handled as its own negotiation for and reservation of network resources, inhibiting similar data streams from being intelligently combined or multicast. Even with a resource management layer, the underlying network is still based on the traditional routed network model, meaning that intelligent utilization of alternate data paths is limited. A model is needed that can articulate each user's service requirements in terms of every layer of the system, from service to application to logical and physical network, and intelligently adapt both the network of application traffic flows and the physical and logical network itself to optimize the flow of information between all users.

This paper examines adaptive networking, then presents a systematic model of adaptive networks, and finally builds a framework in which to study and develop this model. Section 2 surveys a number of existing adaptive networking techniques. In section 3, the adaptive network is discussed as a system, and the concept of relationships between layers is introduced within the context of existing work on QoS requirement translation. Section 4 outlines a campaign of experimentation for further developing these concepts and presents three scenarios as hypothetical initial experiments.

II. Review of Existing Work

The first part of this paper reviews existing work on adaptive networking and in order to develop an understanding of how it extends the traditional QoS model. Badrinath et al. contrast adaptive networks with traditional QoS models in that adaptive networks go beyond a static policy to achieve dynamic resource negotiation between services [2]. Moreover, whereas traditional QoS prioritizes traffic by packet classifiers or routing rules, an adaptive network may enable the services themselves to participate in the resource negotiation process. This means that the application endpoints of a service may be actively involved in the negotiation of services, and may themselves adapt to resource constraints. Their paper presents six adaptive networking systems, three of which are discussed here: Odyssey, developed at Carnegie Mellon; and Conductor and Smiley, developed at UCLA. Each of these systems takes a different approach, and their comparison makes a good study for adaptive network design.

Odyssey is an application-aware environment for resource negotiation. Each application must negotiate for network resources from a central manager; requests are specified within a tolerance range. If at any time the network cannot support the requested service level within the tolerance limits, the manager notifies the application which in turn can adjust its requirements by lowering the quality or quantity of network traffic consumed or produced. Conductor by contrast is application-transparent, utilizing adaptive agents to transform the data in-stream in order to accommodate the network state. This approach requires the adaptive agents to understand the format of the transmitted data and be able to alter it without affecting the state of the application using it. Smiley uses a two-fold human-in-the-loop approach, first by providing interactive feedback to the user about the network resource consequences of performing a particular action before it occurs, and second by predicting the user's near-future data requirements and intelligently pre-fetching data where possible. In these examples, Badrinath et al. capture the essence of adaptive networks simply: by modifying data flows in different ways at different points in the network, the network as a whole can achieve better performance from the collective end users' perspectives [2].

Poellabauer et al. present another application-aware mechanism called Q-Fabric [3]. Their approach combines application level adaptation to available resources with network level adaptation to provide the best possible stability of each data flow. Individual nodes manage their respective resources while working cooperatively to negotiate paths of transmission between one another. However, there is no overarching manager for all nodes collectively. Utilizing an anonymous publish-subscribe model for control communication, each node modifies its behavior based on local resource states plus the feedback it receives from other nodes.

Lu and Bharghavan describe an approach that uses feedback loops between applications and the network, where either may initiate (re)negotiation for resources at any time [4]. As the application runs and the network changes dynamically, the network informs the application if it cannot support that application's requirements, while the application informs the network of its resource needs as they change over time. When an application requests resources that conflict with current resource allocations, the network adjusts every allocation within specified tolerances to achieve the best compromise for all flows.

The next two reviewed papers present novel approaches to resource adaptation. Tung and Kleinrock present a method that adjusts the network iteratively, every iteration rewarding or penalizing each node based on a probability derived from the number of nodes that produce a desirable result [5]. A reward acts as a positive feedback loop, encouraging the node to maintain its state; a penalty acts as negative feedback, causing the node to change its behavior in some way. Since the reward probability function is based on the network state, each node adjusts independently yet the network as a whole adjusts cooperatively to achieve better overall performance and thus more rewards.

Another novel approach is presented by Tennenhouse and Wetherall, wherein every network node becomes programmable [6]. In their model, rather than the network as a cloud that connects endpoints, the network itself is a medium for computation and storage. One potential application is to mimic the traffic adapting method of Conductor [2], where each intermediate node is capable of altering an application’s traffic en route. Another application is data aggregation, where intermediate nodes gather and reduce information from several sources before sending a summary message on toward the destination. Both applications derive from the ability to reprogram each network node on demand.

Helmy discusses large-scale multicasting and introduces the notion of popularity-based adaptive multicast infrastructure [7]. His model augments the single multicast rendezvous point that aggregates all queries and messages. As group membership increases, this system increases the multicast management infrastructure proportionally, and distributes that infrastructure proportionally to group popularity by network region. This approach is unique in the review, using an adaptation scheme based on the popularity of each data flow.

Bordetsky et al. discuss two approaches in [7, 8]. The first paper focuses on a human-in-the-loop approach similar to Smiley [2], wherein each node is equipped with network monitoring capabilities and the operator adjusts physical position (in the case of wireless networks) or application usage to adapt to current network conditions [8]. The second paper [9] describes an application-aware model based on the Real-Time Protocol (RTP). Their model is multicast-oriented, making the receiver primarily responsible for adaptation. The sender provides multiple data streams of varying quality, so the receiver can determine the maximum feasible bandwidth to consume and automatically select the best usable data stream. This combines the application-aware approach of Odyssey [2] with a variation of the adaptive renegotiation shown in [4]. This approach is interesting in that the receiver chooses stream quality on a completely selfish basis, yet the network as a whole adapts to its own constraints.

III. A Systems Approach to Adaptive Networking

Existing approaches to adaptive networking are able to dynamically adjust each flow based on network conditions, which enables applications to continue to function even in a degraded state. This is an improvement over traditional networks, in which an application that is deprived of required network resources will often stall to buffer or crash altogether. Most reviewed techniques provide some form of cooperative adaptation and create feedback loops between layers of the network as well as the user. However, there are a number of higher level adaptations that have yet to be uncovered by existing methods.

In nearly all the adaptive networking mechanisms presented above, the individual application traffic flow is handled independently during resource negotiation, real-time performance evaluation, and traffic adaptation. The notable exceptions are the two novel techniques that vary significantly in approach, and the multicast approach of Helmy [7], which adapts networking resources based on groups of users of a particular data resource. The latter is demonstrative of the power of multicast: it scales efficiently to a large number of users who require the same data resource. A limitation to this solution is that the users must be accessing the exact same resource; if for example two users access the same video stream, but one wants higher image resolution than the other, then two separate streams must be sent. None of the surveyed adaptive network mechanisms are capable of recognizing and adapting to this case, by sending the highest quality stream along the common data path, then branching and sending appropriate streams to each end user. The existing model of resource negotiation based solely on network performance parameters is insufficient for articulating high-level service requirements; however, a multilayer systematic model for adaptive networks may enable the design of a system that could identify this case and perform en-route adaptation similar to Conductor [2].

There has been some work on multi-layer models for mapping higher level user requirements such as video quality, frame rate, and resolution, to network requirements such as bandwidth, latency, and jitter. Siller and Woods propose a three-layer model that relates network parameters such as delay, packet loss, and jitter; application parameters such as resolution, frame rate, and color; and a version of user service requirements they refer to as Quality of Experience (QoE) [10]. They define QoE as user perception of quality of the presentation of the underlying layers. Bauer and Patrick present another model for QoE as an extension to the OSI network model [11]. Above the application layer, they define three human factors layers that start with the display technology that connects the human to the computer, and progressing through human cognition and finally to the overall human need that drives the information process. Quo and Pattinson discuss a four-layer model of user, application, system, and network [12], and use the five types of QoS parameters from deMeer and Eberlein [13] to categorize parameters at different layers. These categories account for network, application, and user parameters, but also add cost-oriented parameters that describe the cost of both the information itself and of moving information across the network. Finally, Nahrstedt and Smith build hierarchies of parameters, defining the relationships between specific parameters at several layers [14].

This paper defines a four-layer model that encompasses the layers most relevant within the author's experiences in the USSOCOM-NPS Field Experimentation Program at the Naval Postgraduate School [8][15]. This four-layer model spans from the physical network up to the network of data services, and is partially aligned with the OSI model of networking layers [16]. The first layer is the physical network comprised of nodes connected by point to point and shared media links. This layer includes switches, routers, endpoint devices, and any other device connected to a medium. The second layer is the logical network, which consists of all routable nodes, including clients, servers, routers, and the logical links between them. The third layer is the set of application traffic flows that traverse the network; at this layer intermediate nodes become transparent, leaving a network of data paths between endpoints. If the network was circuit-switched, or a virtual circuit-switching framework such as Multi-Protocol Label Switching (MPLS) [17] was used, then this layer could be expressed as the set of all active or potential circuits. The fourth and top layer is the set of data services to which individual traffic flows belong. At this layer, the network of individual clients and servers is abstracted away and networks of services form between service providers and consumers. It is at this layer that user-level requirements become meaningful, and through this model can translate into parameters useful to network adaptation.

This model provides a conceptual understanding of the layered network, but in order to utilize this model to translate user-level requirements into application and network requirements, it is necessary to define quantitative and deterministic relationships between these layers. The International Telecommunications Union published ITU-T Recommendation G.1010, which categorizes common types of services and specifies reasonable ranges for network parameters such as packet loss and delay for each type [18]. Liu et al present a model for breaking a distributed service into its component processes, then using a statistical model based on the M/M/1-Queue [19] to translate process flow parameters into network delay parameters [20]. DaSilva also demonstrates a model for determining quantitative relationships between upper layers and lower layers of the network, but his focus is on data loss [21]. He discusses the complex relationship between packet loss and datalink frame loss, and presents formulae for relating the two. Nahrstedt and Smith extend their hierarchies of parameters discussed above and offer several examples of quantitative QoS parameter translation from the application layer to the network layer [14]. The formulae in these papers are important for establishing quantitative and deterministic relationships between network parameters at different layers; this is what enables a Quality of Service model that can articulate user service requirements at higher layers to exist. It also enables the adaptive mechanism to assess performance at higher levels based on the lower level network performance, which in turn enables responsive adaptation based on higher level requirements.

In the following section, a number of scenarios for experimentation are presented. In each of these, the key higher level adaptive attributes are discussed, and a notional hierarchy of service and network parameters is proposed. This is intended to serve as a framework for the development of real experiments for exploring these concepts further and developing usable adaptive mechanisms.

IV. Campaign of Experimentation

The campaign of experimentation is a way of structuring a series of related experiments, from the initial discovery phase, through highly-structured hypothesis testing, and finally into demonstrations of well-developed technology [22]. The experiments proposed here lie within the first two stages, blending the not yet hypothesized study of certain adaptive characteristics with specific tests of the effectiveness of other techniques. These phases focus more on the ability to control each variable in the experimental networks, less on the fidelity to real-world situations. Once these and similar experiments take place and provide feedback into the design of new adaptive mechanisms, additional experiments may be used to phase into a real-world scenario with more distinct elements interacting, and eventually toward the demonstration of these techniques in a tactical exercise.

The inspiration for the scenarios is the Tactical Network Technology (TNT) experimentation series, part of the USSOCOM-NPS Field Experimentation Program, which has a heavy emphasis on the data and tactical components stressed in these scenarios. It is envisioned that these experiments could be performed within the context of future TNT experiments.

Experiment A: User-Tunable Quality of Experience

This experiment envisions a user with specific operational requirements, requiring some form of streaming multimedia data; streaming video is chosen in this case. For example, one user's requirements might be stated as follows: "I need to see video at a resolution high enough to identify a moving vehicle by make and model from a low-flying UAV. This video must be presented to me within five seconds of real-time so that I can provide feedback to operators on the ground. The video stream must be steady enough for me to track the motion of the vehicle. I will be viewing this video in a remote command center five kilometers away."

From this summary, several parameters can be extracted. First, at the application or service layer, there is an overall maximum delay of five seconds from the time the video is received optically by the UAV camera to the time it is shown to the viewer. This not only includes video processing and transmission delays, but buffering delays, which are proportional to the size of the video buffer. It might make sense to minimize the buffer to reduce the overall delay, but the buffer provides a guard against the effects of network jitter and packet loss. Within that buffer window, there is an opportunity for late video frames to arrive or lost frames to be resent. Larger buffers mean more delay but also provide the possibility of a steadier video stream in an unreliable network.

Second, the network layer must support the throughput required for high-resolution video. This means that the choice of medium for transmitting the data is a concern. The medium must be low-latency and should be highly reliable, so that the network loss and jitter are low, in turn allowing the buffer size to be reduced. Finally, at the physical layer, the data path is at least partially wireless if connecting UAVs to a command center located multiple kilometers away. Physical placement of wireless nodes impacts signal strength, which affects both reliability and link speed, which in turn affects the required buffer size and the overall throughput available.

Using the multicriteria analysis method as described by Statnikov et al. [23], these parameters can be categorized into design and criteria variables, which can then be used for evaluation of various configurations. Design variables serve as inputs to the model, describing adjustable parameters within specified constraints. Criteria variables are the desired outputs, which should either be minimized or maximized toward optimal values. Although this approach defines a flat set of parameters rather than the layered model discussed earlier, it provides a convenient means for denoting the inputs and outputs for evaluation. The design and criteria variables for this experiment are shown below.

Design Variable	**Description**	**Units**
Digitizing Delay	Length of time from optical capture of a video frame to its transmission on the network	Milliseconds
Network Delay	Length of time from transmission of video frame from camera to reception at viewing station	Milliseconds
Buffer Size	Size of video reception buffer at viewing station	Kilobytes
Display Delay	Length of time from video frame leaving video buffer to physical display on screen	Milliseconds
Network Jitter	Variance from average delay that a single network packet might encounter in transmission	Milliseconds
Packet Loss	Percentage of transmitted packets that do not reach the viewing station	Percent

Throughput	Rate of video flow across the network	Bits/second
Link Distance	Physical distance between endpoints of the wireless network link	Kilometers
Signal Strength	Strength of wireless network signal; related to link distance, and affects throughput, network delay, and packet loss	dB
Resolution	Video resolution in terms of pixels; function of throughput and reliability	Pixel count

Criteria Variable	**Description**	**Max/Min**
Overall Delay	Delay from optical capture at camera to presentation at viewing station; function of all delays plus buffer size	Min
Video Jitter	Instability in video presentation; function of network jitter and packet loss	Min
Video Framerate	The maximum possible number of frames per second given network conditions	Max
Viewer Distance	Distance of viewing station at command center from UAV or UAV control station	Max

The primary goal of this experiment is to develop an understanding of the relationship between the user's requirements and the corresponding optimal network parameters. A high degree of control should be used here, only allowing one parameter to be changed at a time so that its effects on performance can be captured; however, the choice of parameter change should be determined by the "user." For the sake of this experiment, the "user" should be informed by or automated according to a standard of what constitutes acceptable quality. Some of the criteria such as overall delay are already quantified; other criteria, such as sufficient resolution to identify the make and model of the vehicle, require constraining certain environmental parameters such as altitude of the UAV and optical resolution of the camera, and quantification such as minimal pixel height and width for vehicle identification. This paper assumes that those quantities and relationships are supplied from appropriate existing operational research.

The network design for this experiment should be simple; a single long-range wireless link should suffice to connect the command center to either the UAV itself or to a UAV ground station at the remote site. To reduce the need for complex wireless analysis, it should be assumed that line of sight exists for the wireless link, and that there are no significant sources of interference. However, the wireless amplification may be such that a five kilometer link is at the upper bound of the link’s capability, so that link distance may be a constraint. The choice of video encoding should be such that there exists a linear relationship between resolution and required throughput, and the maximum frame rate should always be send per the current network constraints. All other stability and performance constraining issues should be minimized; the UAV should be flown in such a pattern that the video quality is always optimal to the ground station, and its flying altitude should be within the necessary range to allow vehicular identification per its optics.

The user will have a set of controls or the ability to request changes in design variables where possible. Parameters such as the requested resolution, distance between the ground station and command center, and buffer size will be naturally changeable. The use of traffic alteration devices in the data path could make network jitter, delay, and packet loss malleable as desired by the experimenter. Whereas the goal of the user is both technically to adjust the design variables to optimize the criteria variables and operationally to succeed at identifying the vehicle, the goal of the experimenter is to create an environment in which it is impossible to completely satisfy all requirements, thus forcing the user to make intelligent compromises.

Experiment B: Intelligent multiplexing and multicasting of similar data streams

This experiment focuses on the ability of an adaptive network to recognize cases where multiple users request the same or similar data resources, and intelligently adapt the transmission of those resources in a network-efficient way. An example was proposed earlier: suppose two users want to view the same video stream. The first user wants to use this video stream to perform High-Value Target (HVT) identification, similar to that in Experiment A. The second user wants motion video from a UAV in order to track its location. User A requires high resolution for identification, but may not require a high frame-rate. User B does not require high resolution to identify major ground features as the UAV flies, but requires a higher frame-rate in order to maintain a sense of the direction of motion of the aircraft.

Suppose that all network links are constrained for throughput, and neither user has a link that supports high resolution and high frame-rate video. In the traditional model, the video source will have to send two copies of the video, each suiting the requirements of the destined user. However, a well-designed adaptive network could recognize this case, and assess the efficiency of sending high resolution and high frame-rate video along the common path of both users, then utilizing an adaptive agent to split the streams into appropriate qualities for each user at the point of divergence. Some priority-based QoS approaches can accomplish this through clever encoding of streams, where low-quality video packets are interwoven with packets containing the delta for high-quality video [1]; however, this approach is limited when the user-level requirements extend in multiple dimensions.

This experiment should test an adaptive mechanism that attempts to evaluate and adapt to such cases. The adaptive mechanism must be able to predict network performance based on each user’s higher level requirements. Parameters such as resolution and frame-rate have correlations with throughput. Although the CPU speed of the adaptive agent is not directly a network parameter, it affects the viability of transcoding the input video stream into multiple output streams. Relevant design and criteria variables are shown below.

Design Variable	**Description**	**Units**
Common Data Path Throughput	The throughput of the data path common between the Video Source and both User A and User B; also the path between the Video Source and the Adaptive Agent	Bits/second
User A Data Path Throughput	The unique data path from the Adaptive Agent to User A	Bits/second
User B Data Path Throughput	The unique data path from the Adaptive Agent to User B	Bits/second
Adaptive Node CPU Speed	The processing speed of the adaptive agent, related to its capability to transcode video	Megahertz
User A Resolution	The requested resolution for User A	Pixels
User B Resolution	The requested resolution for User B	Pixels
User A Frame Rate	The requested frame-rate for User A	Frames/second
User B Frame Rate	The requested frame-rate for User B	Frames/second

Criteria Variable	**Description**	**Max/Min**
Common Path Utilization	The utilization of bandwidth along the common path	Min
Path A Loss	Packet loss along User A's unique path	Min
Path B Loss	Packet loss along User B's unique path	Min

The goal of this experiment is to build an understanding of thresholds in network and adaptive agent performance. Specifically, the experimenter should ask how similar two data requests must be before combining and subsequently splitting them becomes more efficient than sending separate streams. In this experiment, the only user is the experimenter, who adjusts each of the design variables as desired, and monitors the behavior of the adaptive mechanism to determine at what threshold it assesses that one case is more efficient than another case. If the adaptive mechanism is controllable, or multiple adaptive mechanisms are available to test, then a better comparison of the tradeoffs between approaches may be obtained.

Like the previous experiment, this will be highly controlled so that the relationship between parameters and their effects on the network can be monitored. This experiment will have lower fidelity per the definition in [22] than Experiment A, since in a real tactical network there will be numerous data streams being evaluated and adapted simultaneously. The network setup should again be simple, using one physical and logical link between the video source and the intermediate node, and single links from the intermediate node to each user.

Experiment C: Utilization of non-optimal network paths for load balancing

The final proposed experiment investigates a network with several alternate paths, with a focus on utilization of alternate or sub-optimal paths in order to balance the traffic load. In a network consisting of multiple network technologies, there may be several paths for sending data between any two endpoints. The traditional routed network model generally assesses and chooses the most optimal path based on one or more metrics, such as link speed, number of routing hops, path latency, or path reliability. In the general case, this is a good approach; the chosen path for application traffic is most likely to be highly reliable and robust, whereas questionable or less capable links are used less heavily.

However, in a resource-constrained environment, there may be value in utilizing these non-optimal paths for traffic with less-restrictive delivery requirements. Lower throughput links may be chosen for low data-rate traffic; an under-utilized path consisting of more hops could be used to balance the load on highly-utilized links. Existing technologies including MPLS [17] and the PPP Multilink Protocol [24] provide a foundation for this technique; however, the multilink protocol is designed for end-to-end link sharing rather than network load balancing, and MPLS focuses on building virtual circuits rather than dynamic adaptation throughout the network. The important facet is the ability of the adaptive mechanism to evaluate these alternate paths and to intelligently select those paths for application traffic with less-restrictive delivery requirements.

This particular experiment does not require a complex mapping of user level requirements onto network parameters, though it may be constructed with that in mind. The focus here is on creating a complex network of many nodes and links, each link having different throughput, latency, and reliability parameters. Then there must be several applications, each with different service requirements, which the adaptive mechanism can evaluate and match against the available network resources to appropriately distribute the traffic load. Some of the relevant design and criteria variables are shown below.

Design Variable	**Description**	**Units**
Link Throughput	The throughput of the link, for each link	Bits/second
Link Latency	The latency of the link, for each link	Milliseconds
Link Loss	The packet loss along the link, for each link	Percent
App Throughput	The required flow for the application, for each application	Bits/second
App Latency	The maximum latency for the application, for each application	Milliseconds
App Loss	The maximum allowable packet loss for the application, for each application	Percent

Criteria Variable	**Description**	**Max/Min**
Links Overloaded	The number of data links in the network that are operating at or over their rated maximum throughput	Min

Notice that in this case the networked nature of the experiment requires each segment of the network to be characterized individually, which means that the number of design and criteria variables changes depending on the size and topology of the network. Also note that the design variables do not include a representation of the data path or network location of each application endpoint. This is left to the experimenter to define.

The experimenter's goal this time is to study the process of assessing alternate paths and selecting paths for various data streams. Since this adaptation is heavily dependent on the topology, it is difficult to quantify the results in terms of benefits to using this adaptive approach. However, this technique differs from the traditional routed network model wherein the logical network converges to a tree with a single unique path between any two endpoints. Therefore one main focus must be on developing and evaluating algorithms capable of determining several alternate paths and monitoring each path's performance state in real time.

Conclusion

This paper presented the current state of the art for adaptive networking, and highlighted its benefits over traditional Quality of Service networking models. It then extended the concepts of adaptive networking by proposing a systems approach that unifies network performance requirements with user service requirements. This provided a means to explore several new adaptive networking ideas that have yet to be implemented, some of which were discussed in the proposed campaign of experimentation. Future work includes constructing and performing the outlined experiments, and developing and implementing adaptive architectures that provide mechanisms to adapt networks and services in novel and efficient ways.

References

[1] Hagen, Silvia. IPv6 Essentials. O'Reilly, 2002.

[2] Badrinath et al. "A conceptual framework for network and client adaptation." Mobile Networks and Applications. May 2000: 221-231.

[3] Poellabauer et al. "Cooperative Run-time Management of Adaptive Applications and Distributed Resources." ACM Multimedia '02. December 1-6, 2002: 402-411.

[4] Lu, Songwu, Vaduvur Bhargavan. "Adaptive Resource Management Algorithms for Indoor Mobile Computing Environments." ACM SIGCOMM Computer Communication Review. August 1996: Volume 26 Issue 4.

[5] Tung, Brian, Leonard Kleinrock. "Using Finite State Automata to Produce Self-Organization and Self-Control." IEEE Transactions on Parallel and Distributed Systems. April 1996: Volume 7 Number 4, 439-448.

[6] Tennenhouse, David L., David J. Wetherall. "Towards an Active Network Architecture." Proceedings of the DARPA Active Networks Conference and Exposition. 29-30 May 2002: 2-15.

[7] Helmy, Ahmed. "Architectural Framework for Large-Scale Multicast in Mobile Ad Hoc Networks." IEEE International Conference on Communications. 28 April-2 May 2002: Volume 4, 2036-2042.

[8] Bordetsky et al. "Network Aware Tactical Collaborative Environments." 9th International Command and Control Research and Technology Symposium. 14-16 September 2004.

[9] Bordetsky et al. "Adaptive Management of QoS Requirements for Wireless Multimedia Communications." Information Technology and Management. January 2003: Volume 4, Number 1, 9-32.

[10] Siller, Mario and John Woods. "Improving Quality of Experience for Multimedia Services by QoS Arbitration on a QoE Framework." Proceedings of the 13th Packed Video Workshop. 2003

[11] Bauer, B., & Patrick, A.S. (2004). *A Human Factors Extension to the Seven-Layer OSI Reference Model.* Retrieved 6 Mar 2006. <http://www.andrewpatrick.ca/OSI/10layer.html>

[12] Guo, Xinping and Colin Pattinson. "Quality of Service Requirements for Multimedia Communications." 19 Jun 1997. Retrieved 6 Mar 2006. <http://www.hiraeth.com/conf/web97/papers/guo.html>

[13] deMeer, J. and A. Eberlein (2001). Toward an Advanced QoS Architecture, in Sloan, A. and D. Lawrence (eds) *Multimedia Internet Broadcasting*, Springer Verlag, ISBN 1-85233-283-2

[14] Nahrstedt, K., and J. Smith. “A service kernel for multimedia endstations.” Proceedings of the Second International Workshop on Multimedia: Advanced Teleservices and High-Speed Communication Architectures. 1994: 8-22.

[15] Bordetsky et al. Center for Network Innovation and Experimentation (CENETIX). Retrieved 27 Jan 2006. <http://131.120.176.50/cenetix/cenetix.asp>

[16] ISO/IEC 7498-1. International Standards Organization. “Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.” Nov 1994. Retrieved 10 Apr 2006. <http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip>

[17] Fineberg, Victoria. “QoS Support in MPLS Networks.” MPLS/Frame Relay Alliance White Paper. May 2003. Retrieved 6 Apr 2006. <http://www.mfaforum.org/tech/MPLSQOSWPMay2003.pdf>

[18] ITU-T G.1010. Telecommunication Standardization Sector of ITU. “Series G: Transmission systems and media, digital systems and networks: End-user multimedia QoS categories.” Nov 2001. Retrieved 14 Mar 2006. <http://ftp.tiaonline.org/tr-30/tr303/Public/0312 Lake Buena Vista/G1010 - 11-01.doc>

[19] “M/M/1 Queueing System.” EventHelix.com. Retrieved 18 Feb 2006. <http://www.eventhelix.com/RealtimeMantra/CongestionControl/m_m_1_queue.htm>

[20] Liu et al. “Mapping Distributed Application SLA to Network QoS parameters.” 10th International Conference on Telecommunications. 2003

[21] DaSilva, Luiz A. “QoS Mapping along the Protocol Stack: Discussion and Preliminary Results.” IEEE International Conference on Communications. 2000: 713-717

[22] Alberts, D. and R. Hayes. Campaigns of Experimentation: Pathways to Innovation and Transformation. CCRP Publications, 2005.

[23] Statnikov, et al. “Multicriteria analysis of real-life engineering optimization problems: statement and solution.” Nonlinear Analysis. 2005: Number 63, e685-e696.

[24] Sklower, et al. The PPP Multilink Protocol. Retrieved 6 Apr 2006. <http://www.rfc-editor.org/rfc/rfc1717.txt>