Adaptation of Collaborative Applications for Network Quality Variation

T Andrew Au, Cindy Tran

Defence Science and Technology Organisation, Australia

DSTO Fern Hill, Department of Defence,
Canberra, ACT 2600, Australia
Email: andrew.au@dsto.defence.gov.au
Phone: +61 2 62566117
Fax: +61 2 62566180

Adaptation of Collaborative Applications for Network Quality Variation

T Andrew Au, Cindy Tran
Defence Science and Technology Organisation, Australia
DSTO Fern Hill, Department of Defence,
Canberra, ACT 2600, Australia
Email: andrew.au@dsto.defence.gov.au

Abstract

The ability to access information on demand is a critical capability in the battlefield. Deployed military headquarters are often located in a challenging tactical environment, whereby major disruptions to collaborative work lead to an impairment in the operational capability. These challenges in turn stem mainly from volatile connectivity and sporadic network mobility, which are a feature of the unpredictable nature of the battlefield. A network management approach based entirely on resource reservation is especially difficult in this environment since disturbances and unanticipated events often occur with network-based applications. This paper explores the concept of network awareness in support of continued operation in an unpredictable battlefield environment. In particular, we propose a platform-independent event-delivery framework to facilitate application adaptation, and demonstrate the advantages of this approach in supporting the proactive management of applications.

1. Introduction

The rapid development of information and communication technology has brought a new level of collaboration and communication to the battlefield [1]. An essential part of the fabric of military operations is the information systems in the headquarters, providing the critical capability to collect, process, and distribute relevant data to remote locations, thus allowing the military force to gain dominant battlefield situation awareness. However, deployed headquarters are often located in a tactical environment that is generally characterised by volatile network connectivity due to mobility demands and hostile attacks [2]. The wireless communications links often suffer from signal attenuation due to Doppler effects and spatial fading, as well as vulnerability to weather, electromagnetic interference, interception and jamming [3]. The consequences of these phenomena are poor use of bandwidth, high latency (delay) and high error rate. These extremely unfavourable conditions degrade the capability to offer high-quality services to network applications. In particular, the viability of collaborative applications necessitates network quality of service.

In the following we describe the conventional approaches to QoS and place them in the context of the characteristics of battlefield networks supporting a collaborative environment. This informs later analysis of the inadequacy of conventional QoS approaches, and the description of a new approach. The strength of this method is demonstrated by its implementation for the enhancement of a particular collaborative application.

1.1 Quality of Service

Quality of Service (QoS) is generally regarded as an end-to-end network application requirement. It imposes bounds and limits on communications requirements such as end-to-end delay, data rate, error rate, and their variances in order to guarantee the performance and stability of the application. The problem of QoS has been studied from different perspectives and at different protocol layers, especially at the logical link layer [4, 5]. The current data loss recovery schemes such as Go-Back-N Automatic Repeat Request (ARQ) or Selective Repeat ARQ can solve the problem of data loss at the data link layer [6] at the expense of greater variations in delay. Nevertheless, the timing requirement of real-time traffic can hardly be satisfied in these circumstances. Further, military threats such as intrusion, jamming, or physical destruction are highly synergistic. Also, the effects on a network of increased traffic resulting from a sudden increase in the tempo of combat could lead to rapid degradation of network performance.

QoS support enables a network to provide a certain level of services that are appropriate for different applications or users. At the network layer, Integrated Services (IntServ) is a service framework [7] to enforce QoS within which network resources are explicitly managed by means of resource reservation and admission control for individual data flows. Typically associated with this approach is the Resource ReSerVation Protocol (RSVP) [8], a signalling protocol that applications could request end-to-end per-flow QoS from the network, and indicate QoS requirements and capabilities to peer applications. Differentiated Services is a complementary approach [9] designed for large-scale networks that uses a simple method of classifying individual Internet Protocol (IP) data packets requiring similar QoS into a limited number of service categories based on packet marking and traffic prioritisation.

Management of tactical battlefield networks presents a multitude of challenges due to the unpredictable nature of the links. Network layer QoS mechanisms are designed to deal mainly with congestion loss and queuing effects rather than packet loss due to link errors. A dynamic environment, with potentially moving nodes connected by wireless links, presents additional challenges to providing QoS support to such a network [6]. Thus, the network dynamics can be attributed to variable link characteristics and node movement. Moreover, node movement, including the movement of end systems and routers as intermediate systems, causes network topology changes. This has a major impact on the viability of static QoS mechanisms based purely on resource reservation.

Another source of end-to-end network instability common to both dynamic and fixed environments is variable demand on network resources by end-system applications. The conventional response to variable application demand is based on resource reservation and admission control. Rather than providing an unacceptable QoS to all users, it is generally better to deny service to some users in order to grant a reasonable service to others [6]. In order to appropriately perform admission control, allocate resources, and perform other functions necessary for providing QoS, the network layer needs updated information from the link layer on the effective data rate of each interface, as well as possibly other parameters regarding the transmission quality. Further, periodic mission re-assignments can occur and may also result in frequent traffic pattern shift [2]. It may induce synchronisation of massive traffic demand, which can rapidly bring the network to a standstill.

It is clear that a system that can adapt to the dynamic network conditions is extremely important, preferably without human intervention, or, with the least amount of human intervention possible. Such a system, if feasible, will be of utmost benefit to the effectiveness of a collaborative environment.

1.2 Collaborative Environment

A collaborative environment is designed to provide integrated audio, video, document, data and application sharing. It is the capability of sharing and collaborating on various forms of electronic information and data that makes collaborative environments valuable and attractive. Collaborative environments provide comprehensive mechanisms that remove many of the potential obstacles of collaboration by offering a variety of interactive and visual tools to create, capture, present, and communicate information in the most effective way possible. Often, commercial collaborative tools are developed for moderately capable networks where bandwidth is not a substantial problem. A more challenging scenario is that of coordination between deployable headquarters in which the collaborative tool is critical during the conduct of initial planning of the deployment.

Collaboration and conferencing are not the same activities, in that collaboration need not be synchronous. Email is a simple collaboration tool that allows users to asynchronously exchange information, whereas conferencing involves all participants at the same time who are geographically separate. A key aspect of collaborative work is establishing common ground by sharing information through any given communication channel or media so that the task can be accomplished. Olson and Olson [10] argued that the quality of group work via both video and audio is not significantly different from similar group working with audio only. US Navy examined two independent variables (voice and a collaborative tool) relating to the performance measure of a situation assessment task that required collaboration among players [11]. The lack of significant differences on the quality and consistency scores indicates that voice, the collaborative tool, or both are broadly similar in information effectiveness. Each experimental condition provided adequate means to transmit tactical information to the commander.

Based on these findings, exploiting any functional communication channels for information sharing seems to be a feasible approach to maintaining collaboration among members in the face of network transport problems. For instance, a video conferencing session can be gracefully downgraded to a voice connection if the required bandwidth is not sufficient, or simply to an email conversation to keep the collaboration going. Another advanced solution is to place a format conversion server in the network at a strategic point to offer media conversion (eg, voice to email).

1.3 Contribution of this Paper

This paper explores the concept of network awareness for maintaining distance interaction in a collaborative environment subject to battlefield conditions. In many cases, an end-user has no control over the network resources, but can monitor the network QoS via appropriate middleware. The end-user can therefore adapt the application to provide the best performance possible while the network quality is varying or deteriorating.

Partial failure is a central reality of distributed computing, where one component can fail while the others continue [12]. Further, large-scale networks are often limited in full end-to-end QoS capability due to the diversity of networking technologies. It is likely that these networks have somewhat piecemeal QoS capability. To understand the effect of such partial support of resource reservation, the degraded performance of a QoS-enabled application under different network conditions is demonstrated in Section 2. This is done by emulating the varying degree of resource reservation coverage between a source and a destination. We conclude that the conventional resource reservation approach to enhancing application QoS does not offer a significant advantage in the context of a realistic military network.

Section 3 describes an architecture for exporting network awareness using an event-based approach, suitable for changing network conditions. This enables an application to dynamically adjust the functionality and resource usage. Through cognisance of the network capability, applications are able to proactively perform their functions with greater effectiveness. Policies can also be applied to the application domain, to adapt to changing environmental conditions for optimal performance.

Section 4 demonstrates the advantages of this architecture through the implementation of network awareness capability to enhance Microsoft NetMeeting. With user-defined policies, the enduring performance reveals the benefits of the adaptive QoS approach in a dynamic environment.

Thus, a new architecture to support the proactive management of application is described and its advantages demonstrated. Section 5 concludes the paper.

2. Effect of Partial Resource Reservation

QoS requires the cooperation of all logical layers in a network, from the application layer down to physical media and of all network elements. In particular, guarantee service assumes that every router supports resource reservation. Unfortunately, support for QoS is not always available on every element of a large-scale network. Those nodes not supporting resource reservation may simply ignore QoS requests, or even reroute packets. An example is the Internet where support for resource reservation on every network element is simply not feasible. Even in a less complex but large-scale network, it is difficult to ensure full support of resource guarantees in every network device. Besides, such a network is extremely vulnerable to mismatch of configurations [13], which could cause serious outages or performance degradation. In large-scale networks, an approach based entirely on resource reservation cannot guarantee end-to-end quality service, leaving the distributed applications in limbo.

Figure 1 illustrates an experiment to demonstrate the effect of partial QoS support in a Windows 2000 network. The QoS Admission Control Service (ACS) provides a control point for bandwidth requests for QoS users, and determines if the necessary network resources are currently available. Various network conditions were simulated using the

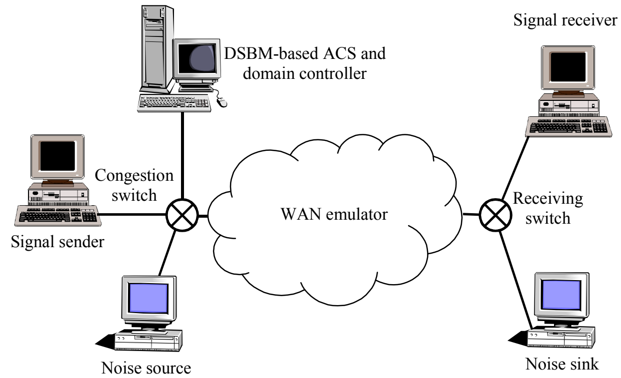

Figure 1: The experimental environment

Cloud WAN Emulator [14]. We used *qtcp*[1] to measure end-to-end network service quality between the signal sender and the signal receiver. The noise generated across the network competes with the *qtcp* session for network resources, creating an overload of 50%.

In the experiment, resource reservation is only effective in the local networks at both the source and destination ends. The simulated link is characterised by the degree of packet loss from 0% to 10% to cover the range expected in the Internet [15] where the packet loss can be attributed to congestion, bit errors, and deliberate discard. Packet discard due to corrupted data is more likely on wireless and satellite links because of poorer link quality. Figure 2 plots the 1-hop throughput versus packet loss, with and without QoS reservation. Extrapolating from this 1-hop measurement, Figure 2 also shows the throughputs for 2-hop and 3-hop links.

The throughputs are gradually reduced as the congestion of the simulated link increases. The number of hops has a more significant effect on the throughput of the QoS-enabled flow than that of the best-effort flow. Under serious congestion, the benefit of QoS reservation diminishes as the QoS-enabled flow goes along more congested hops. Even though best-effort traffic is constantly worse off under severe congestion, QoS-enabled flows are not particularly well treated. This experiment demonstrates that the performance difference between QoS-enabled and best-effort flows is less significant if QoS reservation is not recognised by most of the hops. Given that few routers in the Internet allow QoS reservation, it is by and large a best-effort network over which QoS-enabled applications are not privileged to receive priority treatment in the face of congestion.

3. An Architecture for Exporting Network Awareness in Support of Application Adaptation

In a volatile environment, an application must be aware of current resource

[1] Qtcp measures end-to-end network integrity and service quality for QoS verification. Qtcp sends a sequence of test packets through a network and reports on the queuing delay experienced by each packet.

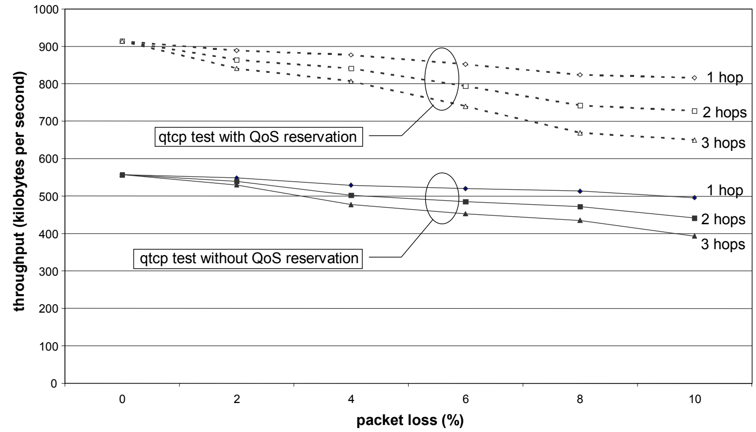

Figure 2: qtcp measurement with and without QoS reservation

availability, so as to dynamically upgrade or gracefully degrade in response. An architecture is required for exporting awareness of the computing and communication environment to an application, usually in terms of asynchronous events. To be network aware, an application must be notified of changes in the network state and the associated information. For instance, on detecting a current network change, an application may need to estimate the network parameters for self-adjustment like typical latency and bandwidth. Such parameters are especially useful with intermittent network connectivity of deployed headquarters. Further, network awareness can be found useful when an application is first activated. The latest network state can help select the appropriate modules for an optimal configuration.

Network detection and adaptation strategies can be constructed directly at the application level, but such ad hoc implementation restricts its reusability by other applications. It also violates separation of concern between functional and environment-specific aspects, and leads to unwieldy and custom code. A powerful means to achieve application portability is to encapsulate platform peculiarities in middleware runtime support, which lies between the application and system resources. Integration of an adaptation mechanism with the middleware brings the awareness and control closer to the application and at the same time, functional aspects of the application are kept separate from the environmental peculiarities.

3.1 The Event Delivery Framework

Figure 3 illustrates a loosely coupled mechanism where a change in the network state is modelled as an asynchronous event. Information is advertised and delivered to interested parties who subscribe to particular events, without prior knowledge of their identities. An event object encapsulates information related to the occurrence of a network change. Upon publication of an event, the event system notifies any subscribers who have registered to receive notification of that event. Each application can implement appropriate adaptive behaviour based on the occurrence of specific events. Examples are events associated with the battery status and the network connectivity in mobile computers [16].

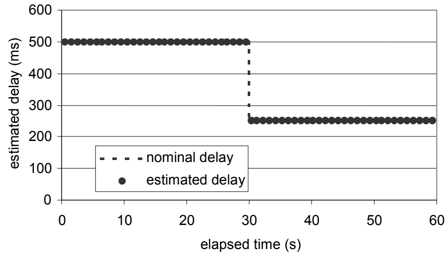

Figure 4: Detecting decreased delay

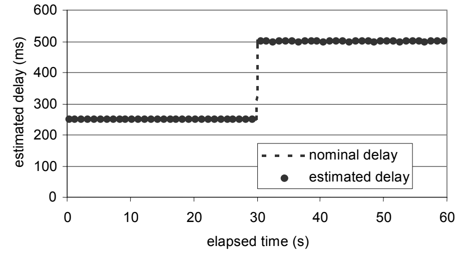

Figure 5: Detecting increased delay

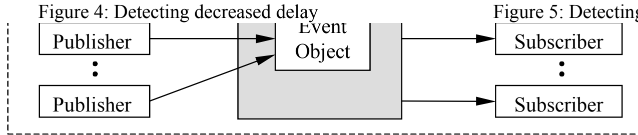

Figure 3: Architecture for network awareness

3.2 Adaptive Applications

Network awareness facilitates the development of adaptive applications. By definition, an adaptive application is able to operate at different operating points during its lifetime, each having different resource requirements. Ultimately the functional constraints have to make decisions on which service-level is functionally possible. Performance sensitivities of an application determine what adaptation algorithm is most appropriate. For example, video streams can be degraded through reduction of different parameters [17]: the frame rate, the quality of individual frames, the size of individual frames or colour components. Whenever a received event indicates that resource conditions have changed, the QoS Manager is able to dynamically reconfigure the application functionalities to match available network resources according to user-defined policies for adaptation [18].

3.3 The Network Monitor

The network monitor performs certain tests on the quality of network service. It consists of individual software modules that can conduct measurement of network conditions, record statistics such as latency, available bandwidth, and round trip time (RTT). Actual available bandwidth or throughput is frequently used as a realistic estimate of the real network performance that one can achieve. It is typically measured by timing the transfer of a large bulk of data. Poor throughput indicates network congestion but it depends also on the configuration of the transport protocol, such as TCP window size.

There are numerous useful network-monitoring tools [19] to determine link state, functionality and capacity. In our experiment, we selected *ping* and *iperf* but these components can be replaced without affecting the rest of the architecture. *Ping* can measure point-to-point RTT and estimate packet loss. Assuming symmetric paths, the one-way point-to-point latency is half of the RTT. *Iperf* [20] calculates available bandwidth using the amount of bytes received and the time it takes to receive all the packets. We configure *iperf* to measure the TCP throughput as the available bandwidth, and to produce the datagram loss of UDP, which is taken as the measure of packet loss.

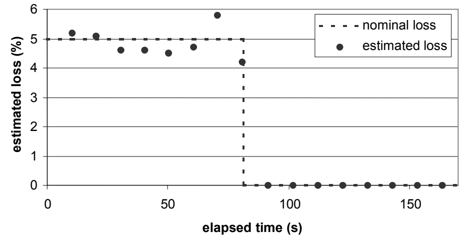

Figure 8: Detecting decreased packet loss

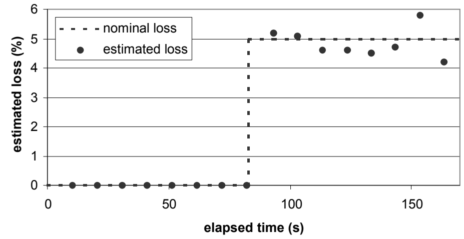

Figure 9: Detecting increased packet loss

3.4 Estimation of Network Quality

The network monitor must be able to faithfully react to true changes in the network. To determine its accuracy, we created a testbed in which we can emulate changing link characteristics. We subjected it to various network conditions simulated using the NIST Net Emulation Package [21]. Similar to the control-theoretic technique of impulse response analysis [22], we synthetically generate two waveforms for each parameter of interest. The first waveform is called step-down that instantaneously decreases an initial value halfway through the trace. The second is called step-up that increases the value.

The results for step-down and step-up waveforms of the RTT are shown in Figure 4 and Figure 5, respectively. Since all estimates observed by *ping* match accurately with the nominal values, it confirms that increases and decreases in delay are easily detectable. The results for step-down and step-up waveforms of the available bandwidth are shown in Figure 6 and Figure 7, respectively. At no load, *iperf* correctly measures the available bandwidth with some discrepancy possibly attributed to overheads for Windows 2000 network. When an artificial load of 51kbytes/s is introduced, *iperf* is able to observe changes in available bandwidth. It confirms that increases and decreases in available bandwidth are detectable. The results for step-down and step-up waveforms of the packet loss are shown in Figure 8 and Figure 9, respectively, at 1 Mbyte of UDP test messages. This estimate is considered reasonably adequate in detecting increases and decreases in packet loss. The test requires a series of UDP datagrams and its accuracy depends on the message length. A better estimate results at the expense of measurement overheads and turnaround time.

3.5 Implementation Details

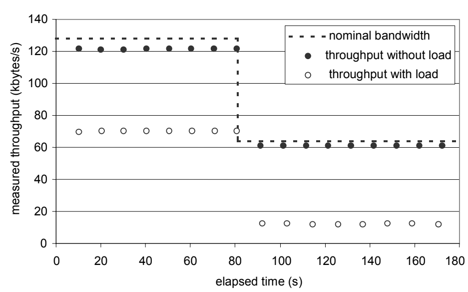

Figure 6: Detecting decreased throughput

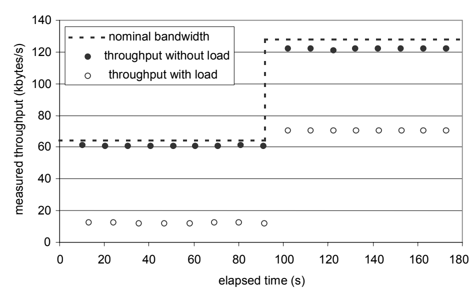

Figure 7: Detecting increased throughput

The event system is implemented based on COM+ Event service where the publishers and subscribers are registered components. A publisher fires an event by calling a method on the event object. COM+ Event service then notifies the subscribers. Persistent subscription to the COM+ Event service is required since it can withstand system reboots and is not bounded by the lifetime of the subscriber application. Each persistent subscriber resides in the COM+ database and is built as a dynamic link library containing a class that implements the interface of the event (See Figure 10). When new estimates are available from the network monitor, the publisher fires an event to the COM+ service in the server computer. The COM+ service then delivers the event to each subscription through its proxy. Once an event arrives at a client computer, the COM+ service loads and instantiates the subscriber, then the event is delivered to it. This event is recorded into the registry in the client computer for future reference and becomes available for use should an application require. If an application is already active, the event is also passed to the QoS Manager (See Figure 3).

4. Enhancing NetMeeting with Network Awareness

Collaborative environment is an important application in the battlefield. The knowledge of network conditions allows the users to exploit the use of resources rather than aggravate the network problem with aggressive retries. For demonstration, we chose to enhance Microsoft NetMeeting with the capability of network awareness. Depending on the prevailing network conditions, users are advised to appropriately alter the functionalities of NetMeeting for optimal performance. The network monitor component on the server computer implements *ping* and *iperf* to retrieve the network conditions of a destination. *Ping* returns the RTT, while *iperf* calculates the available bandwidth based on TCP. *Iperf* also estimates the loss using the feature of UDP. These measurements are collected as an event to be fired off by the publisher on the Event Service. The subscribers receive the event regularly and are informed of any changes. These results are written into the registry and a message is displayed to advise the user. According to the network conditions and the QoS policies, the user is advised to take adaptive actions if required. Figure 11 shows the user interface containing four entities: a ListView to display network conditions, an ActiveX Control NetMeeting, a suggestion box to recommend possible actions, and a log to record all messages and actions.

4.1 QoS Manager

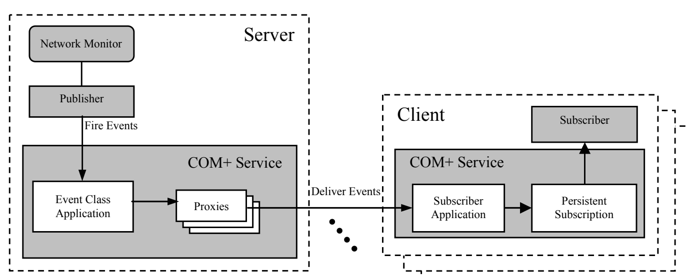

Figure 10: Implementation of multiple persistent subscriptions

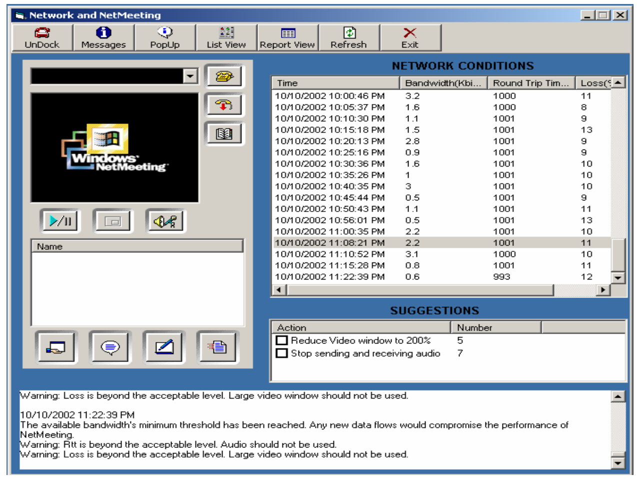

Figure 11: User interface to NetMeeting for network awareness

The QoS Manager receives information about network conditions (available bandwidth, delay, and packet loss), and in consultation with QoS policies, advises NetMeeting users to take particular actions. In the face of extreme network constraints, it helps achieve robust operation through adaptive self-configuration in an ongoing dynamic balance.

Enhancing an application with adaptation may also compromise its user-friendliness. Hence, we implemented the adaptive decisions in a semi-automatic fashion so that the user is always consulted before an action is taken. Messages about network conditions are displayed, providing warnings to users and suggesting an optimised configuration.

The drawback of a dynamic application is information overload to users. It may become another burden if the user interface is not well designed. A better approach is using an intelligent agent (such as Microsoft Office Assistant), who is able to assist the user about all the network applications within the operating environment. The agent can be set to operate so that it meshes with the way the user works. Users will automatically get suggestions on how to configure the applications for optimal performance within the network constraints.

4.2 QoS Policies

The QoS Policies capture the rules for network adaptation that dictate optimal modes of operation for any given environmental conditions and user requirements. These are versatile user preferences for usage monitoring and application configuration. We implemented a set of lightweight policies to allow self-configuration when NetMeeting is started, and to allow automatic reconfiguration when NetMeeting is already running.

The measure of available bandwidth reveals the current network usage, indicating a degree of possible congestion. The rules for bandwidth adaptation were set according to reference measurements [23]. Users are able to predict whether it is feasible to start NetMeeting or to add a new information flow (voice, video, whiteboard, chat, application sharing, or file transfer). When NetMeeting is first activated, the latest network state is retrieved from the registry that is used to advise the user of the optimal configuration of NetMeeting. When a new information flow is initiated in a bandwidth-constrained environment, the user is advised of the possibility of network congestion.

From the QoS point of view, timing parameters of interactive applications are sometimes more important than the data loss ratio. When the round-trip delay is more than 50 ms, echo becomes a problem without echo cancellation [24]. When the one-way delay is greater than 250 ms, talker overlap is made worse [24]. One-way delay of more than 400 ms is unacceptable for voice [25], whereas video is generally regarded as a supplementary medium in video conferencing. Our policy enables voice only if the RTT is less than 800 ms. Packet losses greater than 10% for audio and video data are generally intolerable in multimedia applications unless the encoding scheme implemented provides extraordinary robustness. Our loss policy enables voice only if loss is less than 10%. With these policies, the enhanced NetMeeting demonstrated its enduring performance compared to the standard version, revealing the benefits of the adaptive QoS approach in a dynamic environment.

5. Conclusions

To facilitate adaptation in an unpredictable battlefield environment, we have proposed a platform-independent event delivery framework so that collaborative applications and their users can be made aware of changing network conditions. As middleware, it allows concerns about the application functionality and adaptation to be kept conceptually separate from concerns about network awareness. For example, a video conferencing session can be gracefully downgraded to a voice connection if the required bandwidth is not sufficient, or simply to a message conversation to keep the collaboration going.

In the application domain, policy-based management can optimally adapt applications to changing environmental conditions. According to user preferences and application specific policies, adaptive applications are able to adjust their behaviours appropriately through cognisance of the communications capability, and perform their intended functions with greater effectiveness. We chose to enhance NetMeeting to demonstrate the viability and salient features of the proposed design. We have evaluated our implementation in an emulated network environment. Depending on the prevailing network conditions, users are advised to appropriately configure NetMeeting for optimal performance. The results have shown that this approach can be an important component in graceful performance degradation and in maintaining network-based system availability.

In summary, we have described and shown promising results on the use of proactive adaptation by the end-users in response to poor and variable network quality. With adaptive applications, users with little or no formal training can control their own use of available resources. Knowing the prevailing network conditions, users can help prevent further network deterioration. For better usability, an intelligent agent can be configured to handle all system information and to provide only timely advice as necessary to the users.

References

[1] B. T. Robinson, "Who Goes There?" IEEE Spectrum, October 2003.
[2] L. Kant, C.-H. Zhu, D.K. Hsing, and M. Lee, "Design of an Adaptive Configuration Management Architecture for Tactical Battlefield Networks," Proceedings of IEEE 2000 Conference on Military Communications, MILCOM 2000.
[3] H. Zheng, S. Wang, and J. A. Copeland, "QOS Aware Mobile Video Communications," Proceedings of IEEE 1999 Conference on Military Communications, MILCOM 1999.
[4] A. Ohta, M. Yoshioka, and T. Sugiyama, "PRIME ARQ: A Novel ARQ Scheme for High-Speed Wireless ATM – Design, implementation and performance evaluation," Proceedings of IEEE 1998 Vehicular Technology Conference, VTC 1998, pp. 1128-1134.
[5] O. Nakamura, H. Matsuki, T. Oono, and T. Tanaka, "Pre-Repeat Selective-Repeat ARQ in Fading Channel," Proceedings of IEEE 1998 International Conference on Universal Personal Communications, 5-9 October 1998, Florence, Italy, Vol. 2, pp. 1253-1257.
[6] M. Mirhakkak, N. Schult, D. Thomson, "A New Approach for Providing Quality-of-Service in a Dynamic Network Environment," Proceedings of IEEE 2000 Conference on Military Communications, MILCOM 2000.
[7] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture," IETF, RFC 1633, June 1994.
[8] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," IETF, RFC 2205, September 1997.
[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," IETF, RFC 2475, December 1998.
[10] G.M. Olson, J.S. Olson, "Technology Support for Collaborative Workgroups," in Coordination Theory and Collaboration Technology, edited by G.M. Olson, T.W. Malone, and J.B. Smith, Lawrence Erlbaum Associates, 2001.
[11] "Review and Recommendations for Development and Experimentation with Collaborative Environments," TTCP C3I TP-10 Report, The Technical Cooperation Program, US Air Force Research Laboratory, Jan 2000.
[12] J. Waldo, G. Wyant, A. Wollrath, S. Kendall, "A Note on Distributed Computing," TR-94-29, Sun Microsystems Laboratories, Nov 1994.
[13] L. Kant, D. Hsing, and T.-H. Wu, "Self-Configuration Management System Design for Tactical Battlefield Networks," Proceedings of IEEE 1998 Conference on Military Communications, MILCOM 1998.
[14] The Cloud WAN Emulator User Guide, Shunra Software Ltd, 2000.
[15] Internet Traffic Report, available at http://internettrafficreport.com/main.htm
[16] G. Welling, and B.R. Badrinath, "An Architecture for Exporting Environment Awareness to Mobile Computing Applications," IEEE Transactions on Software Engineering, Vol. 24, No. 5, May 1998, pp. 391-400.
[17] B. Noble, "System Support for Mobile, Adaptive Applications," IEEE Personal Communications, Feb 2000, pp. 44-49.
[18] V. Witana, M. Fry, M. Antoniades, "A Software Framework for Application Level QoS Management," 1999 Seventh International Workshop on Quality of Service, IWQoS'99, IEEE, 1999, pp. 52-61.
[19] Network Monitoring Tools, available at http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html
[20] Iperf 1.7.0 – The TCP/UDP Bandwidth Measurement Tool, available at http://dast.nlanr.net/Projects/Iperf/
[21] NIST Net, available at http://snad.ncsl.nist.gov/itg/nistnet/
[22] B. Noble, "System Support for Mobile, Adaptive Applications," IEEE Personal Communications, Feb 2000, pp. 44-49.
[23] Microsoft Windows NetMeeting 3 Resource Kit, Microsoft Corporation, 2001.

[24] P. Mehta, S. Udani, "Voice over IP: Sounding good on the Internet," IEEE Potentials, October/November 2001, pp.36-40.
[25] ITU-T Recommendation G.114: Transmission Impairments; Transmission Delay.