

Ticket Servers for Network Traffic Prioritization

Cory C. Beard

Computer Science/Electrical Engineering

University of Missouri-Kansas City

5100 Rockhill Road, Kansas City, MO 64110

beardc@umkc.edu

(816) 235-1550

Fax: (816) 235-1260

and

Victor S. Frost

Electrical Engineering and Computer Science

University of Kansas

Lawrence, KS 66045

frost@eecs.ukans.edu

Ticket Servers for Network Traffic Prioritization

Cory C. Beard and Victor S. Frost

Abstract-For broadband packet networks to be widely useful to society, they must dynamically recognize some network flows, like those that deal with disaster response, military operations, or emergencies as having greater importance than others. This paper proposes an architecture of geographically distributed ticket servers that issue importance tickets that indicate the priority that a flow should be given in the current dynamic network context. Any type of user or flow can be given priority, depending the user needs and the context. User agents contact ticket servers using an agent communication language, then a ticket server intelligent agent determines how valuable of a ticket to issue. Use of ticket servers and agent communication enables quick adaptation to dynamic context changes and provides user feedback so that high priority communication activities can be conducted effectively.

Keywords-Priority, ticket servers, resource management, emergency management

1. INTRODUCTION

Modern broadband data networks are being designed to integrate all types of multimedia traffic. More importantly, however, they are designed to integrate and support the activities of all types of *users*. As networks become more and more useful to society, new types of users and user applications emerge, and people will increasingly rely upon these networks to be available and reliable.

The user type of particular interest in this paper is the National Security/Emergency Preparedness (NS/EP) user. Recent terrorist events in the United States on September 11, 2001, have shown that

telecommunication networks provide tremendous value to society in response to disasters. However, these events have also shown what is common with disaster response -- that tremendous stress is placed on these networks. In New York City, most notably the stress was on wireless networks and the public switched telephone network (PSTN). The stress came both from damaged facilities and network demand up to 400% more than normal [1].

Of particular interest for this paper are multimedia communications and applications over the public Internet. Aside from isolated problems with web sites of news organizations, the Internet performed admirably after the events of September 11 [2]. But it is anticipated that as convergence of voice and data services occurs over the Internet in the near future the same problems experienced by telephone and wireless networks will increasingly be seen on the public Internet.

NS/EP users currently use the public Internet for outreach, information sharing, and electronic mail. However, use of the public Internet for mission-critical activities is currently modest [3, 4]. Instead of using the public Internet for critical functions, the NS/EP community depends more on specialized PSTN services, like the Government Emergency Telecommunication Services (GETS) [4], and on dedicated TCP/IP networks. The reliability and security of the public Internet is considered inadequate for mission-critical functions, even though several applications have been identified that could make valuable use of the public Internet [5], such as coordination of response teams, applications to assess extent of damage, and medical information and image exchange.

This paper specifically addresses the issue of network resource availability for NS/EP network communications. It is common in NS/EP contexts for the availability of network resources to be severely restricted in the aftermath of a major event [6] [7] [8] [9]. In some cases, demand on traditional telecommunications networks has reached five times normal levels during the first day of an event [10]. Much of the traffic demand, however, is for lower priority uses, such as people outside a disaster area

calling to see the status of loved ones [11]. Also, user behavior can hamper disaster response. For example, with the Alfred P. Murrah Federal Building Bombing in Oklahoma City in 1995, out-of-town media called emergency operations centers and demanded to stay on the line until they could get information ([9], p. 356). Overloaded circuits also made it very difficult for off-duty emergency workers to call the emergency command center when they received a page to come help with the emergency.

Serving priority users in NS/EP situations involves providing a so-called "emergency lane" for priority traffic [12] where important resource requests are given priority access to network resources. This can be realized in many ways, for example through using dedicated resources, policies to limit resource usage by low priority users [13], or preemption of low priority users when no free capacity is available. No such mechanisms are currently implemented in the public Internet [3], however, although requirements for these mechanisms have been defined in Recommendations by the ITU for the International Emergency Preference Scheme (IEPS) [14] and International Emergency Multimedia Service (IEMS) [15]. Substantial work has also been conducted by the International Emergency Preparedness Working Group of the Internet Engineering Task Force [16] [17].

When resources are scarce, those users and user applications that are of higher value or importance should be given greater access to resources (i.e., non-congested network access). During congestion that occurs in normal conditions, those of greater importance to network operators would be those which have greater revenue-generating capability. During exceptional conditions like emergencies or disasters, however, users and user applications of greatest importance would likely be those which address danger to life and property. A mechanism for prioritizing user traffic would therefore be valuable in both normal and exceptional operating contexts. This work proposes a mechanism for doing just that. Traffic management mechanisms to preferentially handle priority traffic are being developed elsewhere [13] [14] [15] [16]; this work addresses how those priority flows are first identified and authorized.

1.1 REQUIREMENTS

Exceptional conditions can occur suddenly and the context in which the network would be operating would be highly dynamic. Effective administration of priorities in such circumstances should meet the following requirements.

1. Dynamic context awareness - Users and user applications should be evaluated in light of the current dynamic context. This would include not only consideration of the presence of a crisis, but also phases of relief efforts (i.e., search and rescue, medical relief, economic relief, etc.). During normal operations, context awareness would entail knowledge of network loading and traffic engineering constraints.
2. Ability for any user to potentially be considered high priority - A simple approach to administering priorities would be to assign static priorities based on user identity. This would not be sufficient during a crisis, however. Network resource allocation should more closely match the E-911 paradigm where any user at any time could be experiencing or observing an event which would warrant a high priority request for network resources. Stories from recent disasters have talked about members of the general public who were very useful at coordinating and observing disaster response.
3. Transference of priorities across multiple network domains - Priority assignments should be consistent across the multiple domains that a traffic flow traverses. Users should not be required to obtain priorities separately for each domain, nor should the differences in priority assignments be significant between domains. While most of such capabilities would likely be developed through contractual and trust arrangements between governments and service providers, a generally accepted approach needs to be adopted to communicate priorities across domains.
4. Feedback to users and the ability to negotiate the priorities they are being given - In a crisis

situation users are already distressed by the emergency itself. They need not be given additional undue frustration from an inability to obtain communications resources. User satisfaction is, therefore, an important design consideration. Users should be provided information from the network about why their flows cannot be supported (i.e., they should be given more than a simple busy signal). They should also be able to provide information to the network about who they are, their view of the current context, and why they believe their flows should be important in that context.

5. Ability to apply priorities to connection-oriented, connectionless, and aggregated flows - The administration of priorities should not require use of connection-oriented mechanisms. The scalability limitations of those mechanisms have been widely documented (e.g., [18] [19]). Priorities would most likely be implemented in conjunction with traffic admittance and policing functions.

Of the above five requirements, the one which places the highest demand on an architecture to distribute priority assignments is the requirement that any user can potentially be considered high priority. If such a requirement did not exist, then a set of people could simply be given passwords or access codes that they could use when needed. No ticket servers would be necessary; this is how current PSTN systems work today, as with GETS [4]. However, strong objections have been raised about implementing resource prioritization for wireless networks, precisely because only those with passwords could be considered priority [20]. In addition, a report by the National Research Council said,

"Priority policy may be a function of the situation, the role of each participant, their locations, the content being transmitted, and many other factors. The dynamic nature of some crises may be reflected in the need for dynamic reassignment of such priorities ([12], Chapter 2)."

This paper discusses an architecture and system prototype that automatically and dynamically determines which flows or packets should be treated with higher importance than others, so services can be provided accordingly. It consists of geographically distributed ticket servers that issue tickets for resource allocation. Users directly contact ticket servers to negotiate for tickets. This architecture serves as a supplement to work being conducted in the Internet Engineering Task Force (IETF) on the development of policy frameworks [21] [22] [23] [24] by providing a dynamically adaptive mechanism for prioritizing user traffic.

Section 2 describes the overall architecture, its use of importance ticket servers, and its contribution to work being conducted on policy frameworks. Section 3 then discusses the difficulties to overcome, with Section 4 describing an intelligent agent communication approach that addresses these difficulties. The conclusion of the paper qualitatively examines the benefits of the architecture, presents results from a performance analysis of the architecture using a system prototype, and discusses how the benefits of the architecture outweigh the complexity of implementation.

2. IMPORTANCE TICKET SERVER ARCHITECTURE

The proposed architecture is illustrated in Figure 1. The key components of the architecture are the importance ticket servers. The concept being used here is similar to authentication servers such as RADIUS [25] servers and Kerberos [26], but is extended beyond simple user authentication to an assessment of the user's importance based on context and intended activity. Ticket servers can be controlled either by network providers or emergency management agencies. To be treated with high importance, the user obtains an importance ticket from a regional importance ticket server. The ticket is then given to network manager agents that allocate resources to establish flows with one or more other end users. If the ticket server cannot verify identity or some other claim from the user, the user may contact authentication resources. Flows could be connection-oriented flows (e.g., ATM or IETF Intserv

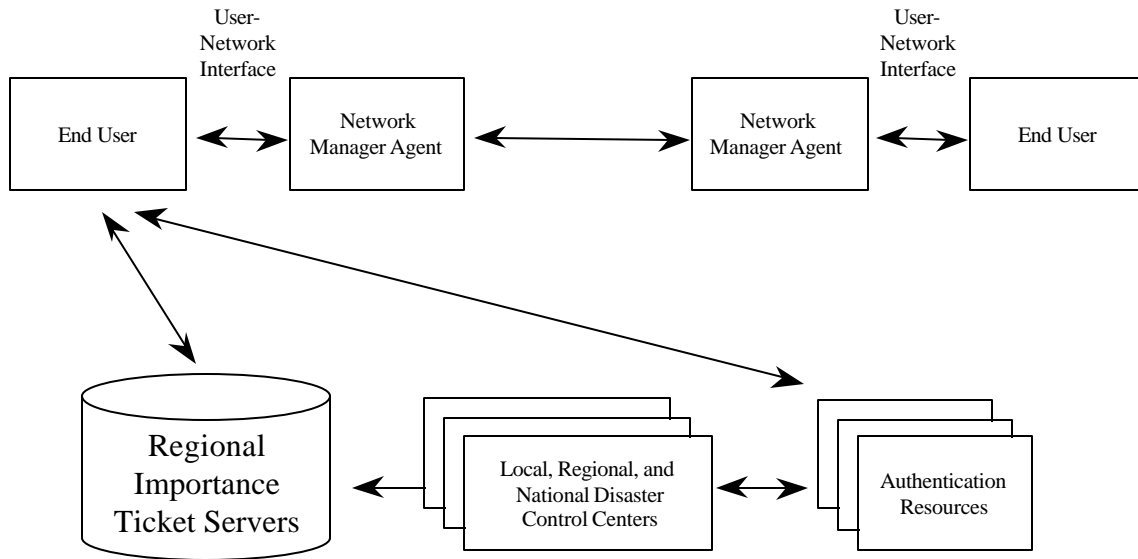


FIGURE 1 - FLOW IMPORTANCE ADMINISTRATION ARCHITECTURE

[27]), in which case the network manager agents would perform connection admission control (CAC) and establish flow state information in nodes along the flow's path. These flows could also be connectionless in environments like best effort IP domains or those which employ IETF Diffserv [28], where tickets would be used to determine packet marking, traffic conditioning, and access to resources at ingress routers.

2.1 IMPORTANCE TICKET SERVERS

Ticket servers provide a mechanism that is dynamic, interacts directly with end users, and provides flexibility in the definition of which flows are important in particular contexts. Tickets are issued based on a close monitoring of the dynamics of the network and the context in which it is operating. This is accomplished by maintaining a context model from the observations of network users, from emergency personnel, and from interaction with local, regional, and national disaster control centers as illustrated in Figure 1. These inputs can then be used to manually (by human operators) or automatically change the servers' models of the current network operating environment.

Users have the ability to directly interact with the ticket servers. Users contact ticket servers using a high priority signaling connection and provide the following information to the ticket servers for a ticket request.

1. User identity (sender and possibly receiver).
2. User organizational affiliation
3. User's view of the current context - This could be normal context, local emergency (i.e., E911 situations), general emergencies (e.g. tornadoes, hurricanes, bombings) and phases of disaster response.
4. User work function - Normal function or some type of special activity.
5. Authentication information or third party authentication resources.

By providing some or all of this information, the importance of a flow can be determined more accurately than if importance were simply determined based on user identity. This functionality would allow the general public to present E-911 requests and provide observations about the current context. It would also allow local emergency agencies, rescue workers, doctors, and hospitals to communicate. If necessary, servers can use authentication sources to verify information that is presented. It is also assumed, however, that claims by anyone that exceed a certain level of urgency will be believed and checked for their veracity later (as is normally the case with most E-911 systems).

By interacting directly with ticket servers, users can request tickets, ask about why a ticket decision was made as it was, provide further information, provide additional authentication information, or ask that the ticket server reconsider its view of the current context based on the input provided by the user. This ensures that servers have accurate information and provides users with opportunities to understand and influence the server's decision processes. Users and servers have the ability to negotiate decisions on ticket requests by further communication. User satisfaction is high because users can learn the server's

view of the current context and adjust their expectations accordingly. For example with the Oklahoma City Bombing, the general public was not aware for some time that the disaster had occurred [9]. A ticket server architecture could have provided such information to users as they were denied access to network resources.

Ticket servers are trusted, automated representatives of network operations organizations. They correlate information provided by users with their knowledge of the current network context (network overloading and failures, natural disasters, societal conditions, etc.). In terms of the five basic requirements listed in Section I, the ticket servers themselves meet four of those requirements.

1. Dynamic context awareness is provided through server dynamic context models and model updates through input from operators, disaster control centers, and users.
2. The ability for any user to potentially be considered priority is provided through the ability for any user to attempt to obtain a ticket.
4. Feedback to users and the ability to negotiate the priorities they are being given is provided through direct interaction between users and ticket servers.
5. The ability to apply priorities to connection-oriented, connectionless, and aggregated flows is provided through network manager agents taking tickets and applying them to their own environment.

Additional benefits of this architecture are the following.

- Detailed tracking of network usage - Importance tickets allow networks to track network usage based on the type and number of tickets that are issued. This provides a detailed, instantaneous view of the numbers of users and the proportion of users with different emergency needs.
- Rapid assessment of the availability of communication resources – Decision processes for issuing tickets are performed at servers, not at switches or routers.

- Low overhead – No overhead is added to the standard access process if users choose not to contact the servers.
- Regional deployment – Geographically distributed deployment of ticket servers allows users to contact, with high probability, at least one server to request tickets.

2.2 *IMPORTANCE TICKETS*

The remaining basic requirement not addressed in the previous section, the ability to transfer priorities across multiple network domains, is provided through the concept of the tickets themselves. Here tickets issued by ticket servers are named *importance tickets*, rather than priority tickets to reduce confusion related to multiple uses of the term "priority" in broadband networks. Priority has been used to apply to packet level mechanisms such as priority queueing or priority packet discarding, as well as at flow, session, and application levels. Importance here is meant to apply specifically to users and their communication sessions. It is important to note that a flow of high importance need not require high levels of quality of service (QoS). An importance ticket is simply needed to provide better access to the resources necessary for whatever level of QoS is being required. Important flows would probably require better QoS than best effort service (so as to obtain a predictable service quality), but not all important applications would require low delay or high data rates.

Importance tickets provide the ability to transfer priorities across multiple network domains because they are similar to electronic cash in that they are a generally accepted mechanism, have a generally accepted value, and are securely stored and transferred. Importance tickets are recognized across multiple domains as having been issued by a trusted source. They are not similar, however, to electronic cash in that they expire and are not transferable. Since the value of a ticket depends on the current context, two requests by the same user for the same amount of network resources may receive very different ticket values at different times.

The use of importance tickets only provides a mechanism by which priorities can be assigned and transferred, but the recognition of such tickets would have to be developed through contractual and trust arrangements between governments and service providers. It is not likely that service providers would be mandated to accept such tickets. Business models of network providers will likely also require that they be reimbursed for importance tickets they accept, but the specifics of such models could vary widely.

The purpose for using importance tickets is threefold.

1. Importance tickets supersede payment mechanisms. Even if a user can spend a large amount of money for a flow, it is always possible to issue a ticket that is more valuable based on context and urgency.
2. Importance tickets are not necessary when a user does not expect special treatment from the network, causing no unnecessary network overhead.
3. Importance tickets provide an alternative to payment mechanisms for users who cannot pay for an important or emergency connection.

An importance ticket has three base attributes: duration, a flow identifier, and value. The duration parameter specifies a time period (possibly short) until a ticket expires, and the flow identifier specifies one flow for which it can be used. Ticket values and pricing mechanisms are related, but, beyond a certain point, adding value to an importance ticket can only come from further justification of the flow's importance in the current context. In addition, optional attributes could also be added to the ticket, such as maximum bandwidth allowed, allowable destinations, etc.

No attempt has been made here to determine the number of priority levels that should be used, whether priorities would be static or dynamic throughout the duration of a flow, or whether priority values should have local or global significance. Resolution of those issues would likely depend how national and regional regulatory policies are applied to contracts with network service providers. All that is proposed here is use

of a ticket that would be widely understood and recognized.

2.3 SOLUTION CONTEXT

Now that an overview of the proposed architecture has been presented, it is important to review the historical context of NS/EP communications and the way this technology would integrate with other technologies currently being developed.

Connection importance has been used in today's United States PSTN with the Telecommunications Service Priority [29] which uses static priorities within a fixed context based solely on user identity. To implement such a function, the United States National Communications System has established the Government Emergency Telecommunications System (GETS) [4] to meet requirements for a survivable, interoperable, nationwide public switched telephone service for authorized government users engaged in emergency activities. GETS satisfies these requirements by providing emergency access and specialized call processing in local and long distance telephone networks. This provides a high rate of successful call completion during network congestion or outages arising from natural or manmade disasters. Users access GETS services by using a special telephone number and personal identification number. In the wireless domain, the Federal Communications Commission has developed rules for the Priority Access Service (PAS) similar to GETS that can be made available to public safety personnel to meet national security and emergency preparedness needs [30]. In military networks, users employ Multi-Level Precedence and Preemption (MLPP) [31] [32] to specify their own priority (flash override, flash, immediate, priority, or routine) within authorized limits.

In contrast to these systems, the architecture presented here dynamically determines connection importance based on the current context (not just user identification) and does not allow users to decide their own importance but rather works with users to negotiate for tickets. In cases where using only user identity would be sufficient, the ticket server architecture still would be useful to implement, because the

tickets could serve as a secure and generally accepted priority administrator.

This architecture supplements technologies in the area of policy frameworks that were devised to provide a generalized, policy-based mechanism where devices, users, and traffic can effectively be managed in diverse network environments [22] [23] [24]. Decisions about network operations are performed based on general policies that are stored and used uniformly across an administrative domain. Policy enforcement points (PEP's) perform the function of network manager agent that was introduced above, as well as other functions. Policy decisions are made based on policies that are stored in policy repositories and administered and interpreted by policy decision points (PDP's).

In general, policies are meant to be relatively static and reflect how decisions should be made for different traffic demands in particular contexts. A ticket server does not implement policy, but rather provides a ticket that PEP's can use to determine which rule to apply to a user's request. Ticket servers adapt to dynamic network conditions and user requirements, and policy frameworks respond to those conditions and requirements by applying predetermined policies. Some types of policies could be developed that would only execute in the presence of an importance ticket.

3. DIFFICULTIES IN DETERMINING IMPORTANCE

Even when an architecture is defined that uses importance ticket servers to assist the network in assigning resources, a significant problem still must be addressed. How do these servers decide which flows are more important than others? Answering this question is especially difficult when considering network failures, users who are new to a geographic area, and dynamic context changes.

The architecture must address the following implementation issues to ensure that the decisions of servers are generally acceptable and reliable.

- Ensure that information users provide is verified.
- Provide high user satisfaction.

- Resolve differences in viewpoints between servers and users.
- Clarify misunderstood information, e.g. the use of terminology not recognized by the ticket server.
- Make correct decisions within the dynamic context and the phases of an emergency.
- Manage server performance by controlling the volume of messages from each negotiation session.

4. DESIGN

To address the issues above, the design of the ticket server architecture is based on two basic technologies -- intelligent agents and agent communication. Agent communication provides a rich negotiation protocol, and intelligent agent technology exploits capabilities for autonomy and reasoning. An intelligent agent working on behalf of the user provides the server with information, provides verification, requests tickets, asks for explanations, and presents new information. The intelligent agent in the ticket server works as an automated representative of a network provider or emergency management agency to decide on the value of a ticket. It determines if it agrees with the user's facts and view of the current context, then determines the value of the ticket for the current context. The ticket server intelligent agent also controls the flow of a negotiation for a ticket and ends a negotiation when necessary. Table 1 lists implementation issues for the architecture and shows whether agent communication (AC) or an intelligent agent (IA) technology provides benefits to address that issue.

TABLE 1 - CORRELATION OF AGENT COMMUNICATION AND INTELLIGENT AGENT BENEFITS TO
DESIGN REQUIREMENTS

Issue	AC	IA	Discussion
Verification of Information	✓		Servers verify information by using agent communication to have users provide references to verification sources or provide verification directly.
User Satisfaction	✓		User satisfaction will primarily involve how well the server communicates to the user why it made its decisions and how it gives the user a chance to affect those decisions.
Differences in Viewpoint	✓		When the server disagrees with the user's view of the context, the user can inquire about the server's perspective and provide new information.
Misunderstood Information	✓		Agent communication allows a user to find out what information might have been missing or misunderstood and update it.
Dynamic Context		✓	Server intelligence correlates users and groups with the current context and phases of an emergency.
Manage Server Performance		✓	Server intelligence controls the flow of negotiation sessions and ends those sessions which are unnecessary to continue further.

4.1 INTELLIGENT AGENTS

The ticket server architecture uses user and server processes that exhibit intelligent behavior, namely autonomy, communication ability, negotiation, and reasoning. These are some of the behaviors characteristic of intelligent agents [33] [34]. User agents act in a semi-autonomous fashion to negotiate for tickets for the user. Server agents act semi-autonomously to decide how user claims correlate with the server's view of context and how ticket values should be assigned. Disaster response organizations and network providers provide the guidelines on how decisions should be made and ticket server intelligent agents carry out those decisions. Servers also perform context model updates and decide how to service user requests for explanations. Servers must be able to control the flow of a negotiation with a user and, for performance purposes, end a session with a user when it is not likely that continuation of the negotiation would result in any more benefit to the user.

Many options exist for implementing the agents' intelligence. The purpose here is simply to argue that

intelligent agents would provide the capabilities and flexibility to apply the knowledge and instructions of human experts. It is important that different organizations be able to implement decision processes to best fit their organizations' goals - a government organization would likely make decisions differently for tickets than a general network provider. For the purposes of a system prototype, a rule-based system was used for the intelligence in the server to decide on ticket values [35] [36].

4.2 *AGENT COMMUNICATION*

In the agent communication approach, communication between the user and the ticket server is not performed directly by the people involved, but rather by automated processes acting on their behalf. These automated processes have the ability to share information, make some decisions on their own (e.g., ask for a reason when too low a ticket value seems to have been issued), and carry out the communication process.

Since intelligent agents can act on behalf of humans and in some ways act similar to humans, attempts have been made to develop protocols for agents to also communicate similar to humans. This began as an attempt to help agents share their knowledge with each other in the same way humans share their beliefs and mental states. This was seen as having great potential at enhancing agents' ability to work together to solve common problems. Here these capabilities are extended to also make ticket requests and carry out negotiations.

Agent communication languages provide mechanisms based on speech acts [37] [38] [39] and enhance the ability for agents to share knowledge and cooperate. Speech acts (also called performatives) are human communication mechanisms which explicitly assert, direct, commit, permit, or prohibit actions in others [40]. In the study of human natural language utterances, Austin [41] noticed that sentences uttered by humans do not always simply assert facts. Sometimes they actually perform actions by having certain forces associated with them. A significant action can be performed simply by performing an utterance.

Examples of speech acts are making verbal commitments or issuing requests, both of which communicate more than just the speakers' view of the world, but also how they intend for others to respond.

Use of an agent communication language provides both a standard mechanism and flexibility in communication. An agent communication language limits the set of speech acts that can be performed. Here the allowable performatives are “tell”, “ask-about”, and “request” [35]. The user will “tell” particular information about a requested network flow, “ask-about” the status of the negotiation and information that was not verified, and “request” tickets and that the server update its view of the context based on the information the user has given. The server will also use these same performatives in the negotiation session. By using these performatives, the negotiation can proceed in a standardized manner, but with flexibility and in a manner consistent with how a human interaction might take place.

This project used the Knowledge Query and Manipulation Language (KQML) [38] as a tool to implement the agent communication.

4.3 *COMMUNICATION MODEL*

Figure 2 provides the communication model for the interaction of users and servers for the acquisition of importance tickets. Boxes indicate intelligent agent decision processes, while lines indicate communication interactions. The interactions could possibly be accomplished with a protocol that does not use explicit speech acts, but the wide variety of interactions that occur here make explicit speech act representations most effective. This allows for a lack of ambiguity in the communication process and a clear representation of what is occurring.

The following observations can be made about the model.

- Users first choose how to try to exert authority. A user can either choose to pursue negotiations as a general user or a user with high authority.

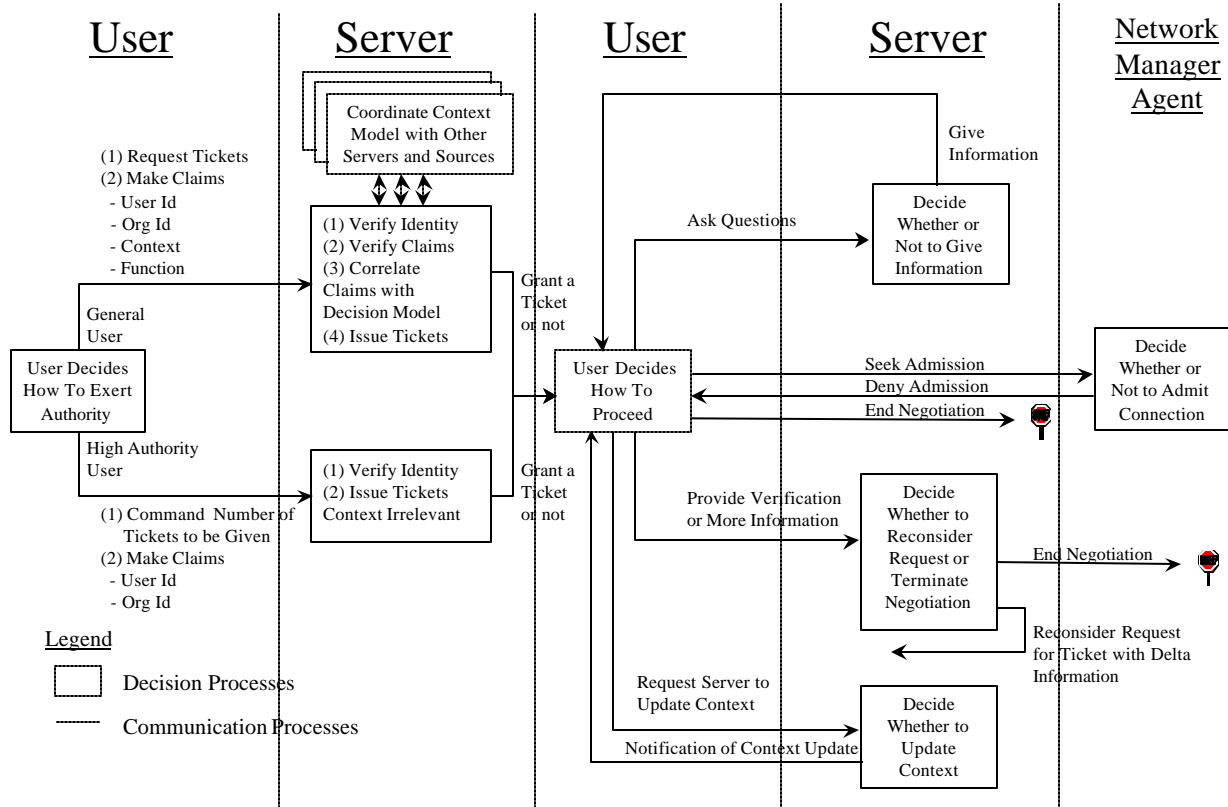


FIGURE 2 - IMPORTANCE TICKET SERVER AGENT COMMUNICATION MODEL

- Users can ask for explanations, provide more information or verification, or request that the server update its model of the current context.
- Users can ask about their verification status and provide further verification information. This verification can be provided directly through a mechanism like a digital signature or indirectly by giving a reference to a trusted third party.
- User satisfaction is high by allowing users to ask if servers verified their information, if servers agreed with their claims about the context, and how their request was classified. Ticket servers could even decide to always provide simple answers to some of these questions automatically with any denied request, so user agents would not have to request that information.

- Differences in viewpoint are addressed by allowing users to ask about the server's agreement with the user's claims.
- Users can request that the server update its context model with other servers, or they can ask that their own information influence a change in the context model.
- Intelligent agent reasoning processes decide on the value of tickets that are issued and decide how to proceed with negotiations.
- A series of security issues must be addressed that are not investigated here. It is assumed those issues can be addressed through the variety of key management, authentication, encryption, and secure transmission mechanisms that are being developed and deployed [25] [26] [42] [43] [44]. Ticket servers would also need to implement common mechanisms that are being used to prevent denial of service attacks.

It is not intended that users know that they are satisfied simply from the value of a ticket. Users will only know if their ticket values are sufficient when they attempt to acquire communication resources or if they are preempted by other users.

5. QUALITATIVE DISCUSSION OF BENEFITS

As a comparison to the ticket server communication model provided above, Figure 3 provides an example of how a standard signaling protocol [45] might be extended to support traffic of different importance levels. An extension would be required to allow users to include information in a resource request about their user identity, organization affiliation, work function, view of the current context, and authentication. The basic flow of operations is that a network manager agent contacts the ticket server on its own to get a ticket for the user's flow. Users do not interact with the ticket servers at all and are only given an acceptance or denial notification from a network manager agent.

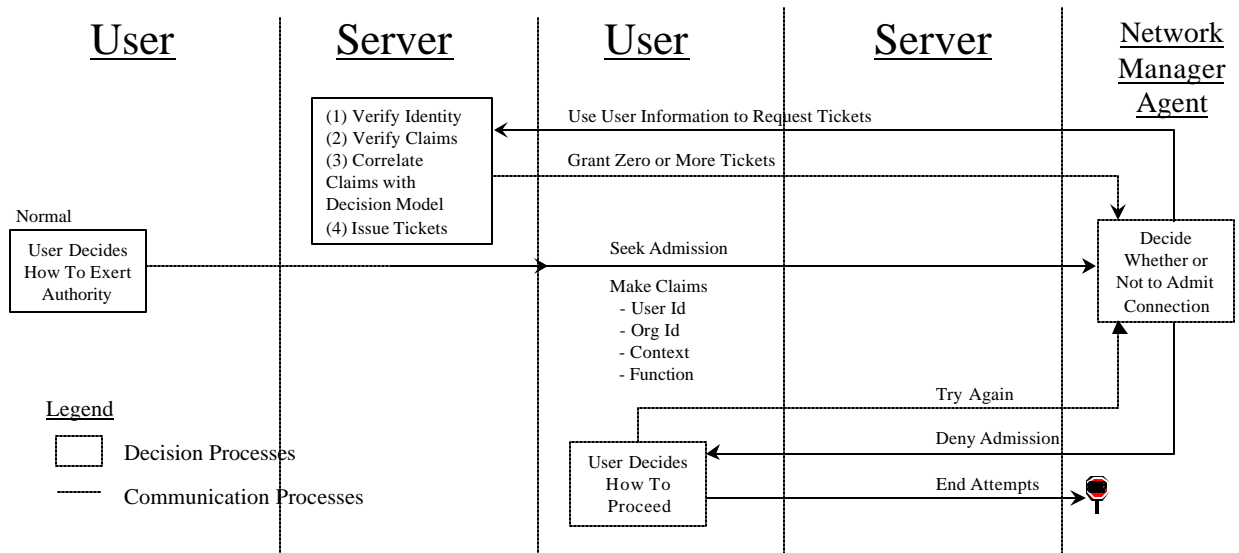


FIGURE 3 - EXTENDED SIGNALING COMMUNICATION MODEL

Many typical scenarios were considered, and the importance ticket server architecture was found to provide significant benefits [35], especially in the area of user satisfaction, over the extended signaling approach. Users have a greater chance for flows being admitted, greater awareness of how their flows were treated, and greater flexibility for influencing outcomes. As disaster response users find greater satisfaction and efficiency, they are more capable of performing their intended tasks well.

6. ARCHITECTURE PERFORMANCE ANALYSIS

There are a series of issues in the implementation of this architecture of ticket servers using intelligent agents and agent communication processes. These include the following.

- Processing of agent communication.
- Implementation of intelligent agents with autonomy, reasoning, and communication ability.
- Secure exchange of tickets and verification information.
- User and server decision processes about how to proceed with negotiations for tickets.
- Frivolous communication by users who misuse the ability to ask questions or seek to hurt the performance of the system.

- Updating of context models.

In [35] [36], the implementation feasibility and performance of this architecture was studied through prototyping and simulation. A system prototype was developed to simulate user and ticket server processes and their negotiation processes using Java Agent Template, Lite (JATLite) [46] that provided extensions to the Java language to support agent communication using KQML [38]. To provide intelligent agent capabilities, the Java Expert System Shell (JESS) [47] was used. This provided a rule-based expert system shell for user agents to generate requests for tickets and respond to those requests. JESS was also used for server processes to respond to requests for tickets based on the current dynamic context and respond to user requests to renegotiate tickets.

This prototype was tested through a variety of possible negotiation situations. Then processing demand on the ticket servers was assessed using a combination of negotiation results from the prototype and a model of ticket server processing using an M/M/1 network of queues. Each ticket server process was modeled as a server with a queue and messages were routed between those queues depending on the type of message. In addition to assessing server performance requirements, the model was also used to assess extra delays users might experience because of interaction with ticket servers and extra network overhead traffic that would be generated from ticket server negotiations.

Two scenarios were studied that simulated a hurricane event and an office building bombing event (like the Oklahoma City Bombing in 1995 [9]). Reasonable assumptions were made for input traffic load patterns, balance of emergency and non-emergency users, disaster dynamics, and network capacity [35] [36]. Figure 4 shows results from the hurricane scenario. Setup delays are shown for the traffic classes that placed highest demands on the server, with the worst setup delays peaking in the range from 5 to 10 seconds. Figure 5 shows the extra overhead traffic that was generated in the hurricane scenario for three network links. Peak overhead traffic was on the order of 1.5% of the capacity. The general conclusion

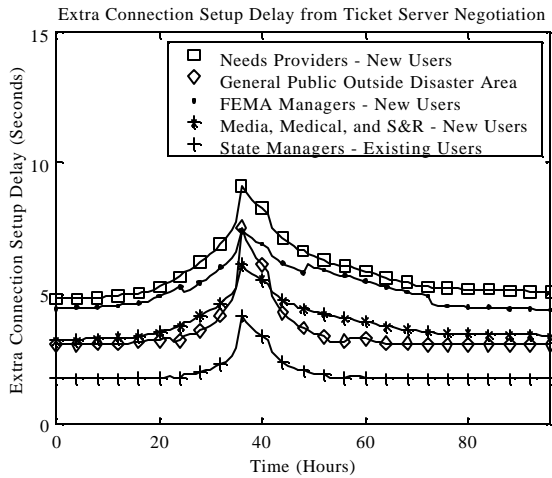


FIGURE 4 - USER SUBCLASSES WITH LONGEST DELAYS FOR HURRICANE SCENARIO

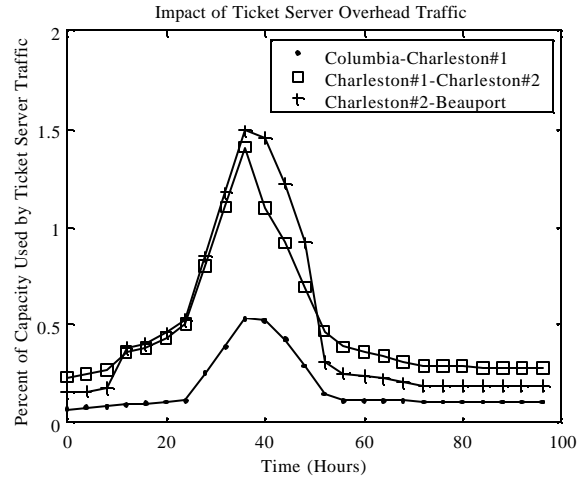


FIGURE 5 - NETWORK OVERHEAD FOR TICKET SERVER TRAFFIC IN HURRICANE SCENARIO

from studying both the hurricane and office building bombing scenarios was that ticket server performance requirements, network overhead, and connection setup delays were not found to be prohibitive to the implementation of the ticket server architecture.

7. CONCLUSION

This work proceeded from the assumption that modern, integrated networks must treat some flows with greater importance than others, especially in a crisis. An importance ticket server architecture was then introduced that gave ticket servers a responsibility separate from network manager agents for determining connection importance. Users provide servers with five basic types of information and then servers issue tickets for gaining admission to the network.

It is proposed that users directly interact with the servers using explicit intelligent agent communication methods. This gives users flexibility to deal with common scenarios. It is of great benefit to users to be able to provide verification of their claims, to find out why they received tickets as they did, and to try again for higher valued tickets. This work is significant because it defines the importance of flows without relying upon access code mechanisms and applies agent communication to network resource

management. The architecture is reasonable to implement, a viable way of determining the importance of flows, and a significant step forward in making integrated networks fully useful to the society in which they operate. By implementing this architecture in conjunction with the prioritized resource allocation mechanisms presented in [13] [14] [15] [16], the Internet can become a more useful resource to save lives and property in response to a disaster.

ACKNOWLEDGEMENTS

This work was partially supported by the Madison and Lila Self Graduate Fellowship at the University of Kansas. An earlier version of this work was presented at the IEEE International Conference on Communications, Vancouver, Canada, 1999.

REFERENCES

- [1] B. Brewin, Nation's Networks See Sharp Volume Spikes After Attacks, *Computerworld*, September 17, 2001.
- [2] E. Noam, Testing the Communications Network, *The New York Times*, September 24, 2001.
- [3] The President's National Security Telecommunications Advisory Committee, Network Group, *Internet Report: An Examination of the NS/EP Implications of Internet Technologies*, June 1999.
- [4] National Communications System, *Fiscal Year 2000 Annual Report: Exploring Solutions for Communications Reliability*, 2000.
- [5] Federal Emergency Management Agency, Technology Applications by the Federal Emergency Management Agency in Response, Recovery, and Mitigation Operations, *Paper Presented to the 27th Joint Meeting of the U.S./Japanese Panel on Wind and Seismic Effects*, Tokyo/Osaka, Japan, May 16-27, 1995.
- [6] C. Wilson and A. Lindstrom, Survival of the network: The single greatest disaster in U.S. history couldn't take down telephone service, *Telephony*, October 23, 1989.
- [7] J. S. Griswold, T. L. Lightle, and J. G. Lovelady, Hurricane Hugo: Effect on State Government Communications, *IEEE Communications Magazine*, June, 1990, pp. 12-17.

- [8] A. Taylor, Andrew is Brutal Blow for Agency, *Congressional Quarterly Weekly Report*, Sept. 12, 1992, p. 2703.
- [9] The City of Oklahoma City, *Alfred P. Murrah Federal Building Bombing: Final Report*, Fire Protection Publications: Stillwater, Oklahoma, 1996.
- [10] S. Adamson and S. Gordon, Analysis of Two Trunk Congestion Relief Schemes, *Proceedings of IEEE MILCOM '93*, pp. 902-906, 1993.
- [11] G. Philip and R. Hodge, Disaster Area Architecture, *Proceedings of IEEE MILCOM*, 1995, pp. 833-837.
- [12] Computer Science and Telecommunications Board, National Research Council, *Computing and Communications in the Extreme: Research for Crisis Management and Other Applications*, 1996.
- [13] C. Beard and V. Frost, Prioritized Resource Allocation for Stressed Networks, *IEEE/ACM Transactions on Networking*, Vol. 6, no. 5, October 2001, pp. 618-633.
- [14] Description of an International Emergency Preference Scheme (IEPS), International Telecommunication Union (ITU), Geneva, Switzerland, *ITU-T Recommendation E.106*, March 2000.
- [15] Draft ITU-T Recommendations F.706 - International Emergency Multimedia Service (IEMS).
- [16] International Emergency Preparedness Working Group, Internet Engineering Task Force, <http://www.ietf.org/html.charters/ieprep-charter.html>
- [17] H. Folts and C. Beard, "Requirements for Emergency Telecommunication Capabilities in the Internet," Work-in-Progress, Internet Engineering Task Force Internet Draft, draft-ietf-ieprep-requirements-00.txt, June 19, 2002.
- [18] X. Xiao and L. M. Ni, Internet QoS: A Big Picture, *IEEE Network*, March 1999.
- [19] Y. Bernet, The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network, *IEEE Communications Magazine*, February 2000.
- [20] S. Romero, Hurdles to Wireless Priority Access, *The New York Times*, October 22, 2001.
- [21] R. Rajan, *et. al.*, A Policy Framework for Integrated and Differentiated Services in the Internet, *IEEE Network*, September 1999.
- [22] R. Yavatkar D. Pendarakis, and R. Guerin, A Framework for Policy-based Admission Control, RFC 2753, Internet

- Engineering Task Force, January 2000.
- [23] R. Boutaba and A. Polyraakis, Extending COPS-PR With Meta-Policies for Scalable Management of IP Networks, *Journal of Network and Systems Management*, Vol. 10, No. 1, March 2002.
- [24] T. M. T. Nguyen, N. Boukhatem, Y. G. Doudane, and G. Pujolle, COPS-SLS: A Service Level Negotiation Protocol for the Internet, *IEEE Communications Magazine*, May 2002, pp. 158-165.
- [25] C. Rigney, S. Willens, A. Rubens, and W. Simpson, Remote Authentication Dial In User Service (RADIUS), RFC 2865, Internet Engineering Task Force, June 2000.
- [26] J. Kohl, C. Neuman, The Kerberos Network Authentication Service (V5), RFC 1510, Internet Engineering Task Force, September 1993.
- [27] J. Wroclawski, The Use of RSVP with IETF Integrated Services, RFC 2210, Internet Engineering Task Force, September 1997.
- [28] S. Blake, *et. al.*, An Architecture for Differentiated Services, RFC 2475, Internet Engineering Task Force, December 1998.
- [29] Telecommunications Service Priority, *Code of Federal Regulations*, Title 47, Chapter 1, Part 64, Appendix A.
- [30] Federal Communications Commission, In the Matter of The Development of Operational, Technical and Spectrum Requirements For Meeting Federal, State and Local Public Safety Agency Communication Requirements Through the Year 2010: Establishment of Rules and Requirements For Priority Access Service, *FCC Report and Order Number 00-242*, July 13, 2000.
- [31] D. Choi, Multi-Level Precedence for B-ISDN, *Proceedings of IEEE MILCOM '90*, pp. 334-338, 1990.
- [32] Integrated Services Digital Network (ISDN) General Structure and Service Capabilities: Multi-Level Precedence and Preemption Service, *ITU-T Recommendation I.255.3*, 1990.
- [33] M. Woolridge and R. Jennings, Intelligent Agents: Theory and Practice, *Knowledge Engineering Review*, Vol. 10, No 2, June 1995.
- [34] M. Genesereth and S. Ketchpel, Software Agents, *Communications of the ACM*, July 1994.

- [35] C. Beard, *Dynamic Agent Based Prioritized Resource Allocation for Stressed Networks*, Doctoral Dissertation, University of Kansas, 1999. Available at <http://unofficial.umkc.edu/beardc>.
- [36] C. Beard and V. Frost, Ticket Server Performance Evaluation Using a Hybrid Simulation Approach, *Proceedings of the SCS 2001 Applied Telecommunication Symposium*, Seattle, Washington, April 2001.
- [37] FIPA - Foundation for Intelligent Physical Agents, FIPA ACL Message Structure Specification, Document XC00061, August 15, 2001.
- [38] The DARPA Knowledge Sharing Initiative External Interfaces Working Group, DRAFT Specification of the KQML Agent-Communication Language, plus example agent policies and architectures, June, 1993.
- [39] M. K. Chang and C. C. Woo, A Speech-Act-Based Negotiation Protocol: Design, Implementation, Test, and Use, *ACM Transactions on Information Systems*, vol. 12, no. 4, October 1994, pp. 360-382.
- [40] M. P. Singh, *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*, Berlin; New York: Springer-Verlag, 1994.
- [41] Austin, J.L., *How to Do Things with Words*, Oxford University Press, 1962.
- [42] S. Kent, and R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401, Internet Engineering Task Force, November 1998.
- [43] D. Harkins, and D. Carrel, The Internet Key Exchange (IKE), RFC 2409, Internet Engineering Task Force, November 1998.
- [44] IP Security Protocol Working Group, Internet Engineering Task Force, <http://www.ietf.org/html.charters/ipsec-charter.html>
- [45] ATM Forum, ATM User-Network Interface (UNI) Signalling Specification, Version 4.0, *Doc. af-sig-0061.000*, July, 1996.
- [46] <http://java.stanford.edu>.
- [47] <http://herzberg.ca.sandia.gov/jess>.

Cory C. Beard is currently an Assistant Professor of Computer Science and Electrical Engineering at the University of Missouri, Kansas City. His current research funding includes grants from the Sprint Corporation and also a grant from the National Science Foundation Faculty Early Career Development (CAREER) Program for his project “Priority Users and Applications on the Internet”.

Victor S. Frost is currently the Dan F. Servey Distinguished Professor of Electrical Engineering and Computer Science and Director of the Information and Telecommunications Technology Center at the University of Kansas. Both industry and government have sponsored his research. His research interests are in the areas of integrated broadband communication networks. He is a Fellow of the IEEE and received the Presidential Young Investigator Award from the National Science Foundation in 1984.