Real-time Collaborative Environments and the Grid

Colin Perkins
University of Glasgow
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ, UK
Email: csp@csperkins.org

Ladan Gharai
University of Southern California
Information Sciences Institute
3811 N. Fairfax Drive
Arlington, VA 22203, USA
Email: ladan@isi.edu

***Abstract*— We describe some of the issues inherent in running real-time collaborative environments in heterogeneous and fragmented networks, showing how these issues are only partially addressed by both IETF and Grid standards. We propose a new architecture, based on a peer-to-peer overlay IP network to abstract the complexity and heterogeneity of the underlying network. Our architecture is realised as a middleware layer, simplifying deployment of new collaborative environments.**

I. Introduction

The vision of Grid computing is to provide ubiquitous and secure access to high-end computing and database resources for scientists, researchers, and industry [3]. This enables new classes of application to be developed in communities as diverse as high energy physics, meteorology and oceanography. These applications typically require significant compute resources and access to large data sets, often integrated from disparate locations. Advances in middleware technology and the growing prevalence of high-speed networks are bringing us closer to this vision, while placing greater demands on the real-time collaboration tools, security infrastructure, middleware and transport protocols needed to support these developing virtual communities.

Much has been written about middleware technologies and standards for secure data transfer, access and integration using the framework of web services, and how these can be used to build virtual communities around particular application domains. Support for such communities is enhanced through the use of real-time collaboration tools, such as those provided by the AccessGrid framework [8]. This framework brings together the web services and security infrastructure of the Grid community, with multicast multimedia conferencing tools [5, 9] and application sharing services. Unfortunately, while these real-time multimedia services have enjoyed a successful lifetime, their limitations are becoming increasingly apparent and many in the community have recognised the need to develop collaborative work tools better suited for use in the emerging Grid-based collaborative environments [7, 11, 12]. This recognition of the need for better real-time and collaborative tools is indicative of a general need for research and development in real-time middleware infrastructure.

In this paper, we discuss the challenges and technical obstacles to providing future real-time Grid collaborative environments, and present a road-map to attaining this vision. Our paper is structured as follows: in section II we discuss the difficulties in the provision of real-time collaboration environments and outline the limitations of existing systems. After this, in section III, we propose a new architecture for the development of these systems, and discuss how this could be implemented in sections IV and V. We discuss our solution and compare it to related work in section VI, and conclude in section VII.

II. Challenges in Provision of Real-Time Collaborative Environments

Issues relating to the provision of real-time collaborative environments fall into two categories: those relating to the transport of multimedia data, and those relating to the secure discovery of session partners and initiation of the collaborative work session. Emerging virtual collaborative environments tend to use the transport protocols originally devised by the multicast conferencing ("Mbone") community, but replace the session initiation and control protocols with others built using Grid technologies. In the following, we discuss the limitations of these protocols, and highlight the challenges inherent in developing systems based on these choices.

A. Real-Time Transport Protocols

A key component of a collaborative environment is the transport protocol. Collaborative environments require real-time delivery of audio/visual data and shared application state, along with metadata about the participants in the venue and about the venue itself. This has to be achieved with varying and appropriate degrees of robustness, reliability and security, for both point-to-point and many-to-many interactions.

Existing systems primarily use RTP [17] on multicast UDP/IP for transfer of audio/visual data and for presence information, with various ad-hoc protocols used to support application sharing. Unfortunately, inter-domain IP multicast has proved difficult to deploy, unstable, unscalable, insecure and poorly supported. Reasons for this vary from implementation quality to poor protocol design (e.g. the use of MSDP for inter-domain source discovery). The difficulty in deploying IP multicast has limited the growth of virtual communities, and the strong push from router vendors to deprecate the "traditional" IP multicast service in favour of source-specific multicast only raises further issues.

Also, despite the existence of multicast support in both RSVP and in the Differentiated Services framework, there is no deployed quality of service (QoS) infrastructure for multicast conferencing systems. Due to over-provisioning of the network core this is currently less of an issue that might be expected, but has the potential to surface as more and higher-quality collaborative environments are deployed, and as they move from experimental to production status. Given the limited deployment of QoS mechanisms, it becomes necessary for applications to be designed such that they can adapt to available network capacity. This congestion control issue is difficult, since there are unsolved research issues in making adaptive collaborative applications [4], and it is unclear whether the human factors requirements of such applications allow them to be used in today's heterogeneous network environment.

Finally, there are problems inherent in using ad-hoc and non-standard protocols for application sharing which limit interoperability, and hinder the development of middleware to encapsulate best practices.

B. Session Initiation and Control

Two frameworks exist for initiation and control of collaborative work sessions. The IETF has developed an architecture based around the use of SIP [15] for presence and negotiation, with authentication, authorisation and accounting using, for example, Diameter [1]. This contrasts with the Grid community, which uses the AccessGrid [8] with its concept of venues as a meeting place, leveraging the Grid security infrastructure.

The IETF architecture provides for flexible peer-to-peer collaboration, whereas the AccessGrid venues provides an infrastructure that encourages communities to form, since the identity of venues and their participants is more secure and tightly managed, and there is a clear rendezvous point.

A disadvantage of the AccessGrid control infrastructure is that there is currently no systematic approach to NAT and firewall traversal; issues that have received wide attention in the IETF community. The SIP framework has an extensive toolkit for detecting and traversing middleboxes [13, 16] which is lacking in the AccessGrid world. The increasing fragmentation of the Internet into realms with limited connectivity and addressability is a significant problem, and has not been addressed by the AccessGrid community.

III. Architectural Directions

To address the challenges discussed in Section II, we believe it necessary to evolve the design of real-time collaborative environments to reduce their dependence on specific lower-layer technologies, allowing them to adapt to the growing heterogeneity of networked systems and protocols. One way to approach this is to move away from the requirement that the network natively support group communication, and instead build a peer-to-peer group communication infrastructure upon which applications can be hosted. That is, we propose to use the Grid infrastructure and trust model, combined with the session initiation framework of the IETF, to leverage the deployment of a peer-to-peer overlay network on which existing group communication applications can run. The overlay will provide a multicast IP service to client applications, using private address space to distinguish it from other networks. Our approach to overlay building has some similarities to the Xbone [18] and to the Windows peer-to-peer networking framework [6, 10], but leverages the security infrastructure and virtual organisation management tools from the Grid community to build a secure overlay, encompassing a community with known membership.

There are several advantages to such an approach: 1) it is easy to support multicast since the overlay network is relatively small and forms a single administrative domain; 2) applications running on the overlay can avoid problems due to NAT traversal, since the overlay presents a single address space; and 3) the overlay is secure and covers a known set of participants, reducing the need for each application to contain complex security and membership management infrastructure. These issues are pushed down into the middleware that builds the overlay, and hence solved for all applications.

A peer-to-peer overlay network can also be used to enable support for enhanced QoS. This requires support from the underlying network, but if present, the overlay can arbitrate between application QoS requirements and the services of the network. Moreover, applications see a consistent service, irrespective of the underlying network, which may be (enhanced) IP, MPLS, a lambda switched path, etc.

This architectural change has profound impact, beyond its immediate goals: it greatly simplifies application development, facilitating deployment and fostering collaboration through interactive, trusted, virtual environments.

IV. Basic Implementation Outline

In the following we outline an implementation strategy for our architecture. We demonstrate how the Grid security and virtual organisation infrastructure can be extended using the ICE methodology and peer-to-peer protocols as a way to provide NAT transversal, and to establish overlays for real time collaborative environments.

A. The Venue Server Infrastructure

The first step in the initiation of a collaborative environment is to locate and authenticate participants. We assume the existence of well known servers that provide both a virtual environment where participants can rendezvous, along with authentication services for both participants and the meeting environment. These servers – called venue servers in the terminology of the AccessGrid – are assumed to be reachable by all participants.

The venue servers maintain a list of venues: virtual meeting points which have specific identity and may hold state relating to that identity, and which can be authenticated as being the legitimate holders of that state. For example, these might be conference rooms associated with a particular project or research group, or institutional venues associated with a site.

Venues are authenticated, ensuring the entity running the venue can be determined securely via a certificate authority.

Participants are also identified by user-certificates, with a well-known certificate authority running the public key infrastructure necessary to enforce this. We recognise the well-known scaling limitations of a global certificate authority and do not require such: rather we expect user communities will run appropriate infrastructure for their members.

The security services we envisage are consistent with the standard Grid Security Infrastructure components, and other existing services such as the AccessGrid.

B. Session Initiation

Once the desired participants have been authenticated, it is necessary to determine their mutual connectivity. This is a complex operation since it is likely that some participants are located behind firewalls, and others still may be located in different address realms, for example behind IPv4 NAT devices, or using IPv6 when the majority of the session remains an IPv4-only affair.

In addition to the Grid services infrastructure used to authenticate users and query stored state, the venue servers run several protocol components used for session initiation. These include servers for the TURN media relay protocol [14] and signalling proxies suitable for use with the ICE method [13] of connectivity establishment.

To establish connectivity the participants determine the set of possible network addresses on which they can be reached. This set will include all local addresses on the participant's host, along with an address derived from the TURN server running on the venue client. The TURN ("traversal using relay NAT") server provides a last-ditch relay which will provide minimal connectivity should all the participants be unable to directly communicate; it is not expected to be commonly used.

Participants then submit a preference ordered list of possible network addresses to the venue server, which acts as a signalling relay to invite participants to conduct pair-wise ICE exchanges to establish their mutual connectivity. This process can occur incrementally, with a newly joining participant running an ICE exchange with an existing session member, using the venue server as a signalling relay. This gets connectivity between the new participant and some member of the session, if it is possible to do so (in the worst case, with the venue server acting as a media relay, but any direct paths will be found if they exist). The ICE exchange is repeated as needed, to locate enough peers to form an overlay network.

The ICE methodology uses repeated STUN requests [16] to the range of possible addresses for the peer, until one succeeds. The STUN exchanges are keyed using a shared secret derived during the initial authenticated signalling exchange, to ensure that malicious hosts cannot intercept the session initiation.

C. Building the Overlay

A key point of the ICE exchange is that it determines connectivity for traffic between a pair of hosts on a single UDP port (since firewall and NAT devices will behave differently depending on the destination address and port used). Future communication between the participants *must* use ports determined, since it is likely that communication to different ports will not reach the same host (since that host may be a NAT device fronting connections to several hosts in another addressing realm). This requires another ICE exchange for each new application started, unless we construct an overlay using the single available port, and tunnel application traffic on that overlay.

Once connectivity has been established between sufficient participants, it is possible to build a peer-to-peer network overlay connecting them. Such an overlay will run over UDP/IP, since that is the form of connectivity an ICE exchange establishes, and since it allows for real-time communication. Our proposal is not strongly tied to a particular overlay type, although it is beneficial if the overlay provides low-latency routing and a multicast delivery service.

Once the overlay topology has been established an IP addressing realm is created, and each participant's host creates a virtual network interface with an address in that realm. The IPv4 link local address range (169.254.0.0/16) may be used, with hosts choosing their address according to [2], or an IPv6 network may be allocated for this purpose. Packets sent to this virtual interface are then routed across the overlay network to the host corresponding to their destination address. We therefore run IP, using private address space, tunnelled on the UDP/IP based peer to peer overlay, to simulate a single local network segment.

If supported by the overlay routing protocol, hosts can run native multicast on the overlay IP network. If the overlay routing protocol does not support multicast, hosts can run a simple IP multicast routing protocol, for example DVMRP to construct a multicast routing tree on the overlay, at the cost of some additional complexity. Complexity is, however, reduced compared to Internet-scale multicast, since the group is restricted to a single domain with known participant hosts and topology.

D. Running Collaborative Work Tools

Once the overlay is built, collaborative work tools can be run in the usual manner. These applications will see a new IP network interface on the host, with it's own address space, and can use it to communicate securely with other participants in the collaborative work session (the communication is secure in that the participants have been authenticated, confidentiality can be provided either by running IPsec on the overlay IP network, or at the application level). The overlay IP network will be transparent to the applications, irrespective of underlying network address space translation and firewalls, will appear as a single administrative entity, and can also support multicast if desired.

By running an IP overlay on the peer-to-peer network, we provide a simple and well understood service to applications. The burden of maintaining a complex peer-to-peer overlay network, traversing NAT and firewall devices, and multicast

routing is pushed down into an infrastructure component, and can be implemented as a middleware library. This contrasts strongly with a traditional peer-to-peer system, where the application participates in the construction and maintenance of the overlay.

V. Other Implementation Considerations

A concern with our proposal may be the complexity of the signalling, with it's reliance on multiple STUN exchanges between participants, using the ICE methodology in a pair-wise manner. This requires more involvement from the venue server than does the existing AccessGrid, since the venue server must act as a signalling proxy during these exchanges, and may be required to relay media streams if there is no other connectivity between participants. It also places more burden on participant hosts, which can no longer use light-weight client software, and must use a complex signalling protocol to create a peer-to-peer overlay multicast network, rather than relying on network supported multicast.

We accept the increase in complexity because of the benefits it provides. The Internet is no longer the simply connected network it was when the existing AccessGrid framework and media tools were developed: there is much higher use of network address translation, firewalls are more common, and the transition to IPv6 is beginning to take place. Each contributes to the "fog on the Internet" making connectivity more difficult, and forcing us to introduce more complex signalling: without this signalling we cannot automatically establish connectivity between participants residing in different addressing realms, or separated by firewalls.

Additional signalling is also needed if we are to leverage the emerging all optical network infrastructure to provide enhanced quality of service. Once we have signalling in place it is possible to use other types of link in the overlay instead of the underlying IP connectivity, by trying the other links as alternatives during the ICE exchange. This requires some changes to the way the ICE methodology is used, to allow hosts to include non-IP addresses and an address type field in ICE messages, and to perform network-type specific connectivity checks in place of the STUN exchange used to test for IP connectivity, but none of these extensions change the fundamental nature of the exchange.

By using a systematic methodology to enumerate host addresses in order of preference we allow the system to choose the best connection type between pairs of hosts that form the overlay. This may be a best effort IP network, an IP network with negotiated quality of service, or a non-IP network, depending on the available options. For example, a host may advertise to the venue server that it has two network interfaces: an interface that uses standard best effort IP, and an interface that supports QoS negotiation via RSVP. If the latter interface is listed as higher priority, peer hosts that follow the ICE methodology will first attempt to connect using negotiated QoS (using the QoS negotiation protocol as connectivity check), and if that fails will fall back to best effort IP connectivity, using STUN to test for IP connectivity.

Our approach allows the overlay to run on network paths with enhanced QoS, but since the overlay provides a best effort IP service, we provide no explicit QoS support to applications: the quality of service provision applies equally to all applications running on the overlay; there is no notion of priority between applications.

VI. Discussion and Related Work

A. Relation to IETF and Grid Standards

Our proposal reuses many signalling protocols developed by the IETF for session initiation, NAT detection and traversal [13, 14, 16]. Our usage of these protocols is somewhat non-traditional, since we propose to integrate them as a middleware component that can be used to build an IP overlay network, rather than directly integrating them into applications. This usage has practical benefits: it pushes complexity out of the applications into a middleware layer, and brings NAT detection and traversal functions to unchanged existing applications. Our reuse of the Grid services framework and security infrastructure for interactions with venue servers matches existing usage, although our venue servers also provide additional functions to allow the ICE message exchange to take place.

We require some modifications to ICE to support multiple address types. We expect these modifications to be generally useful: they allow connectivity to be determined across a range of network types, and can be used for both collaborative work and other applications (e.g. bulk data streaming).

This combination of standards and frameworks is powerful, and builds on the strengths of the two communities. In particular, we retain the strong authentication and community building properties of the AccessGrid model while overcoming it's limitations with respect to NAT and firewall traversal. The service we provide to applications is simple, and allows us to run existing collaborative work applications unchanged once the overlay has been started, easing deployment.

B. Naming, Addressing and Tunnelling

Many peer-to-peer overlay building protocols have been proposed. These are often used to build application-level overlays, running above the existing network connections of participating hosts. If some of the hosts are using private IP addresses and NAT, the result can be several peers with the same IP address. This forces applications to run a naming service on the overlay to identify and route traffic. This works well for file-sharing applications that wish to route requests based on content hashes or similar application level names, but is less than optimal for collaborative work applications that just need a simple host identifier.

Microsoft's peer-to-peer networking SDK [10] approaches the addressing issue by assigning each peer host a globally unique IPv6 address (e.g. using Teredo [6] tunnels) which is used to address hosts in the overlay. This allows direct communication between peers without an additional address translation table (the SDK provides a naming service that can be layered above the IPv6-based addresses, for applications that need it).

In contrast, our approach accepts that hosts may have non-globally unique addresses, but builds from the observation that a host can be reached via a unique address as seen from any other host. This is because a NAT device hides an addressing realm (the internal realm) behind an address which is unique in the external realm. For example, hosts A and B may have the same IP address, due to being in different addressing realms, but communication from A to B can be sent via a unique address/port – that of the NAT device behind which B is located – in A's realm. Our overlay building approach uses this property, first performing an exchange to determine the external addresses of NATs, then addressing hidden hosts by their externally visible address/port combination. Once we have built the overlay, we layer a simple host identifier scheme above, to hide diverse addressing views from applications.

Both approaches derive unique addresses in place of the fragmented global address space present in the Internet by tunnelling. A related tunnelling system is the Xbone [18], also used for deploying overlay networks. The systems differ in that the Xbone uses IP multicast messages for resource discovery whereas we propose to use a centralised venue server, and because the Xbone does not include a NAT traversal component. Similarities would include tunnel and route configuration to set-up the overlay once participant nodes have been located.

VII. Conclusions

The current heterogeneity of network infrastructure is due to a variety of different forces and historical reasons. Some aspects of this heterogeneity hinder technical advancement and the ubiquity of services by making the network brittle and applications complex, other aspects contribute to the richness and availability of the network, and the applications it supports. To ensure the future development of real-time collaborative environments though, we believe it essential that this heterogeneity be tamed before it overwhelms application developers. To do this, we have proposed the development of middleware to deploy a peer-to-peer overlay substrate, above which applications can flourish independent of the increasing heterogeneity and fragmentation of the network.

The contribution of our work is to present the notion of a peer to peer IP overlay network (distinct from an application-level overlay); we enumerate the steps needed to build such an IP-based overlay across a network fragmented into multiple address realms, and show it can be used as the basis for deployment of advanced collaborative environments. The fragmentation of the network is making it increasingly difficult to deploy such networked collaborative environments without NAT/firewall traversal, yet the protocols and methods needed to effect such traversal are complex and difficult to perfect. To overcome these, we believe middleware such as we propose, abstracting the network traversal functions out to build an overlay networking on which applications can run unchanged, is a desirable goal.

VIII. Acknowledgements

This work is supported by the NSF under grants 0230738 and 0334182, and by the UK National e-Science Centre.

References

[1] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter base protocol. Internet Engineering Task Force, September 2003. RFC 3588.

[2] S. Cheshire, B. Aboba, and E. Guttman. Dynamic configuration of IPv4 link-local addresses. Internet Engineering Task Force, July 2004. Approved for RFC publication.

[3] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3), 2001.

[4] L. Gharai and C. S. Perkins. Implementing congestion control in the real world. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 2002.

[5] O. Hodson and C. S. Perkins. Robust-audio tool, version 4. http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/.

[6] C. Huitema. Teredo: Tunneling IPv6 over UDP through NATs. Internet Engineering Task Force, March 2004. Work in progress.

[7] ANU Internet Futures Laboratory. Video presence v0.7.1. Software available online, June 2004. http://if.anu.edu.au/SW/VP.html.

[8] Futures Laboratory. Accessgrid v2.2. Software available online, June 2004. http://www.accessgrid.org/.

[9] S. McCanne and V. Jacobson. vic: A flexible framework for packet video. In *Proc. ACM Multimedia'95*, San Franscisco, November 1995.

[10] Microsoft. Windows Peer-to-Peer Networking SDK v1.0 and Advanced Networking Pack for Windows XP. Software available online, July 2003. http://www.microsoft.com/windowsxp/p2p.

[11] C. S. Perkins and L. Gharai. UltraGrid: A High Definition Collaboratory v0.3.0. Software available online, August 2004. http://www.east.isi.edu/projects/UltraGrid/.

[12] C. S. Perkins, L. Gharai, T. Lehman, and A. Mankin. Experiments with delivery of HDTV over IP networks. In *Proceedings of the 12th International Packet Video Workshop*, Pittsburgh, April 2002.

[13] J. Rosenberg. Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Multimedia Session Establishment Protocols. Internet Engineering Task Force, February 2004. Work in progress.

[14] J. Rosenberg, R. Mahy, and C. Huitema. Traversal Using Relay NAT (TURN). Internet Engineering Task Force, February 2004. Work in progress.

[15] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. Internet Engineering Task Force, June 2002. RFC 3261.

[16] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN - Simple Traversal of UDP Through NAT. Internet Engineering Task Force, March 2003. RFC 2489.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF, July 2003.

[18] J. Touch. The x-bone. Software available online, May 2004. http://www.isi.edu/xbone/.