A DISTRIBUTED AGENT MEDIA ACCESS FRAMEWORK

Craig M. Gelowitz[1], Luigi Benedicenti[2] and Raman Paranjape[3]

[1]Software Systems Engineering, University of Regina,
Regina, Saskatchewan, Canada
craig.gelowitz@uregina.ca
[2]Software Systems Engineering, University of Regina, Regina,
Saskatchewan, Canada
luigi.benedicenti@uregina.ca
[3]Electronic Systems Engineering, University of Regina, Regina,
Saskatchewan, Canada
raman.paranjape@uregina.ca

ABSTRACT

This paper presents a distributed software mobile agent framework for media access. The access and delivery of media is enhanced by an agent architecture framework that utilizesmobile agent characteristics to provide access and delivery of media. Migrating agents to the media source and destination devices enables the framework to discover the properties of devices and media within networks dynamically. The mobile agents in this framework make decisions and work together to enable access and delivery of media through the transcoding of media based on the properties of the data path and device constraints.

KEYWORDS

Distributed Media Access, Mobile Software Agents, Real-time Transcoding & Media Framework

1. INTRODUCTION

Mobile agent software technology provides the methodology used to create the framework. The following subsections introduce mobile agents and the distributed framework. In section two, the framework approach is presented and identifies some common issues. In section three, the design of the framework is presented. In section four, the issues identified in section two are discussed with respect to the framework. In section five, the video transcoding methodology results are discussed and section six concludes the paper with potential future work.

1.1. Mobile Agents

The mobile agent paradigm is a branch of traditional computing areas such as distributed computing and artificial intelligence. The differentiators mobile agents provide over the

David C. Wyld et al. (Eds) : CCSIT, SIPP, AISC, PDCTA, NLP - 2014
pp. 27–36, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.4203

traditional computing areas are the properties of mobility, reactivity, persistence, social ability and autonomy. Mobile agents are able to react to their environment, communicate, negotiate and migrate to network resources [1, 2].

Mobile agents are created to be goal oriented. When a mobile agent requires a resource to complete a task, it will move to that resource and execute until the resource is no longer required. Mobile agents will migrate to resources when required to reach an execution goal [3, 4]. As a result, the agent model is a useful paradigm to create distributed applications that have the ability to migrate and provide a media access service.

An agent's communicative ability increases the potential advantages of the mobile agent paradigm. This ability enables the development of multi-agent systems. Multi-agent systems differ somewhat from single agent systems. A multi-agent system must include provisions to handle interaction and the current context of other agents in the system. In a multi-agent system, agents or resources can be added or removed from a system as services are initiated or removed. Mobile agents are able to adjust to changing conditions such as network nodes, devices, media services or other agents which may be available or unavailable dynamically. Multi-agent systems can be created that use agent communication to collaborate and work together to provide services, reach a desired outcome or achieve a common execution goal [5, 6].

For these reasons mobile agents are useful for deploying a media access service. Mobile agents exhibit characteristics such as flexibility, reliability and scalability. They are also able to provide different types of distributed applications and services to users with desired preferences and outcomes [7, 8].

1.2. The Distributed Agent Framework

Mobile agents exhibit characteristics such as autonomy, mobility, flexibility and the ability to collaborate and communicate [9]. The distributed agent framework inherits these characteristics from the mobile agent paradigm. These characteristics provide useful mechanisms to provide media access.

The framework provides users with access to media services through the use of agent mobility and communication. Multiple agents work together to move through local and global networks and create message paths for the transfer and consumption of media through networks. Users are given the ability to access media across networks and between devices. Agents within this framework also provide other functions for media access such as the determination of media availability, transcoding of content, media serving and media caching.

A user is able to transfer and consume media at selected devices within the framework. Agents within the framework collaborate to determine the current properties of media resources and the current network conditions in order to execute the appropriate steps to provide content to the user.

2. Approach

The approach taken by the developed prototype for media access utilizes the framework to create a communication path for the real-time transfer of media data. The following section provides a short overview of the framework.

2.1. Distributed Agent Framework Approach

The framework provides a mobile agent enabled mechanism to allow users to share content between devices. The following diagram illustrates an overview of the agent system and its migration and data path which provides content between example private networks.

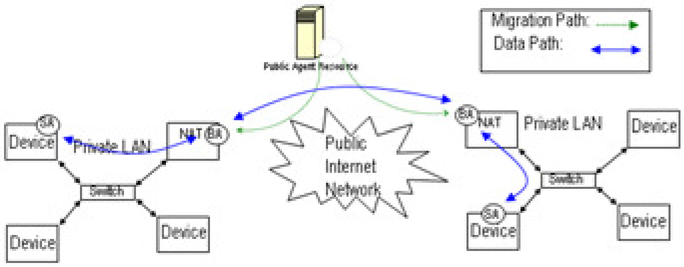

Figure 1. Agent Migration Overview

The interface to the system is provided by a web-server through a web browser interface. The public agent resource is where mobile agents are launched when a media service is initiated by a user. The web-server initiates mobile agent actions for migration and dynamic set-up of the data path between devices.

The framework enables access to content by transferring media data through agent message relaying. The interface enables users to select a source gateway and a destination gateway followed by a source device and a destination device. After the selections are made, mobile agents move out from the public resource to the selected source and destination. The media currently available on the source device is presented to the user through the interface. After a media selection is made, the media is transferred across the networks by relaying messages through the bridge agents from the source device to the destination device. Depending on the type of media available, the bandwidth available between networks and the nature of any time-sensitive selected media such as video, the agents will perform actions to compensate for these conditions to provide real-time consumption at the destination.

2.2. Media Framework Issues

The framework provides a mechanism for content access between devices. The framework faces some challenges which are presented here:

2.2.1. Network Connection

In order to provide user access to media across networks, agents must traverse the network and create a data path for media information. In the case of local private networks, the internal devices are behind a NAT (network address translation) router. This makes it difficult for data paths to be established when they are initiated from the public network. Some peer-to-peer applications use NAT traversal protocols such as TURN [10] and STUN [11]. Agents can provide an alternate mechanism to create a data path. Multiple agents can use message forwarding to relay messages and create a data path for media transfer or application communication [12, 13].

2.2.2. Bandwidth Constraints

The available bandwidth between networks can be highly variable. Video and other time sensitive data such as audio require a certain level of bandwidth to be available for real-time consumption of content. A system that provides real-time media services to a community of users needs to account for available bandwidth between devices.

2.2.3. Media Availability

Mobile agents can aide in file and media access [14, 15]. An agent's ability to migrate enables them to move directly to a resource to inspect content. Upon arrival at the media resource an agent can determine the availability and context of the media. Agents can use this information to make decisions with respect to delivering the content to a destination.

2.2.4. Scalability and Context

Mobile agents require a platform to migrate and execute. This implies that available devices must have an agent execution platform. The availability of these platforms can be dynamic in nature. Devices and network nodes may be available or unavailable at any given time. In the case of a large community of devices and networks, a framework must be able to handle the dynamic nature of available devices in a scalable manner.

3. Design And Implementation

The following sections provide further explanation of the framework with respect to media access and its delivery. The current prototype implements the architecture illustrated below:

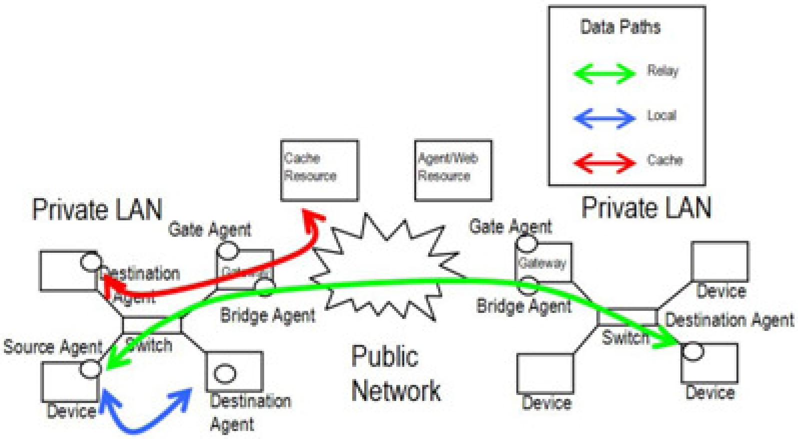

Figure 2. Agent Communication Paths

The figure shows two private networks where agents have moved out to their respective locations to provide a media service. The numbers of private networks that are part of the system are not restricted as a community of users might span multiple networks. The public resource provides the user interface to the system and the launching point for agents to migrate out to devices.

3.1. Device Selection

After the respective source and destination networks have been chosen, a list of devices currently available within the selected networks are made available to the user.

A message is sent to each selected gateway requesting the information about the devices connected to their respective internal networks. The gate agents respond to the public resource with their respective list of devices connected to their internal private networks. The device list for each gateway is created dynamically using a broadcast mechanism which allows the gate agent to keep an updated list of respective devices available on the internal network. Any device in the local network which is executing the mobile agent environment will make it itself known to the gate agent and dynamically be added to the list of devices. In this way, the user is presented with a current list of source and destination devices.

After the source and destination devices are selected, bridge agents and source/destination agents are created at the public repository and move to their respective locations as seen in figure 2.

3.2. Media Selection

When the source agent arrives at the source device, the agent inspects the device for available media. The list of available media is sent back to the user interface for selection. Once a selection is made, the framework determines how to transfer the media.

Within the developed prototype, there are currently four fundamental alternatives for media transfer/consumption. The prototype determines which alternative to use based on the state of the system. The four alternatives are as follows:

3.2.1. Agent Message Passing

Message passing is the default mechanism for media transfer. Media data is passed using agent messaging by relaying messages through the bridge agents and to the final destination as seen in figure 2. The source agent uses the bandwidth determined between the devices through-out the data path to adjust media for real-time transfer. As an example and in the specific case of video data, the source agent will begin transcoding video to a resolution and bit rate achievable by the available bandwidth to transfer.

3.2.2. Local Data Stream

This approach includes an alternate mechanism for media transfer which is performed only in the local area network case. When the identical source and destination network are chosen, the agents will move out and setup in the same fashion as they do when they set up across separate networks. The difference in this case is that the source and destination agents are able to communicate directly without relaying messages using the bridge agents. As such, it becomes no longer necessary to use message passing to transfer media data. The source agent (where the requested file originates) transforms itself into a streaming media service. It opens an IP port for streaming available content and proceeds to wait for a connection from the destination agent. The destination agent connects to the port directly with the appropriate streaming directives. The media is now streamed directly between the agents on the local area network and displayed in its original form. Once the process begins there is very little overhead with respect to bandwidth or computational resources other than what is normally required to consume the content. The gateway and its resources are not utilized because no gateway resource is required after the initial setup has taken place.

4. DISCUSSION

The following section discusses the challenges identified in section two with respect to the framework.

4.1. Network Connection

The agents within the framework traverse the public to private network using a series of consecutive hops. Source and destination agents move first to the respective network gateways by travelling to their public IP addresses first from the public resource. The internal IP addresses of the devices are now available for migration once they have made the first hop to the gateway. The internal IP address of a source or destination device is known through information gathered from the gate agents.

Agent to agent data paths are handled by a relay mechanism using the bridge agents on the gateways. The source and destination agents communicate by relaying messages through the bridge agents. Other data paths such as access to the cache or the public resource is handled in the normal fashion whereby the initiation of the path can originate from within the private network because an agent is executing within the network. The cache and public resource exist at publicly addressable IPs and can be accessed from any device that has connection to the public network.

4.2. Bandwidth Constraints

The framework handles bandwidth constraints by making decisions based on the content required and the conditions of the system. It will attempt to provide content in the most appropriate way to allow users within the community the highest quality experience it can deliver. The methods presented included various ways that media can be accessed and transferred within the framework to adjust for bandwidth constraints.

4.3. Media Availability

The framework determines media availability by sending an agent directly to the source of the media. In this way, inspection of the file system locally provides a list of media currently available at the device. This also allows the system to determine the context and properties of the media at the source. As such, an agent can make decisions locally about how to handle a media sharing request using the information gathered.

4.4. Scalability and Context

The prototype provides an automatic discovery mechanism. By default, the agent platform accepts broadcast query packets and responds to those queries with its IP address, device name and the names of agents currently executing on its platform. This allows the gate agent to query its local subnet for currently available devices. The lists of devices within the larger system of networks are distributed and not centrally located. A query from the system to a gate agent will dynamically identify the current available platforms/devices for the selected network. This distribution of the current system context also provides the prototype the ability to easily determine the current state of the system on a request by request basis. The framework does not centrally maintain the state of the system and as such the community of networks and users can potentially grow in a scalable manner. The mechanism mimics the well-known DNS mechanism and provides the framework a dynamic way of determining currently available devices and content in a distributed and on-demand manner.

5. Results

The results discussed in this section refer to video media because video media requires the framework to adjust for bandwidth and device constraints. To determine a transcoding methodology for the distributed agent framework the following tests were performed.

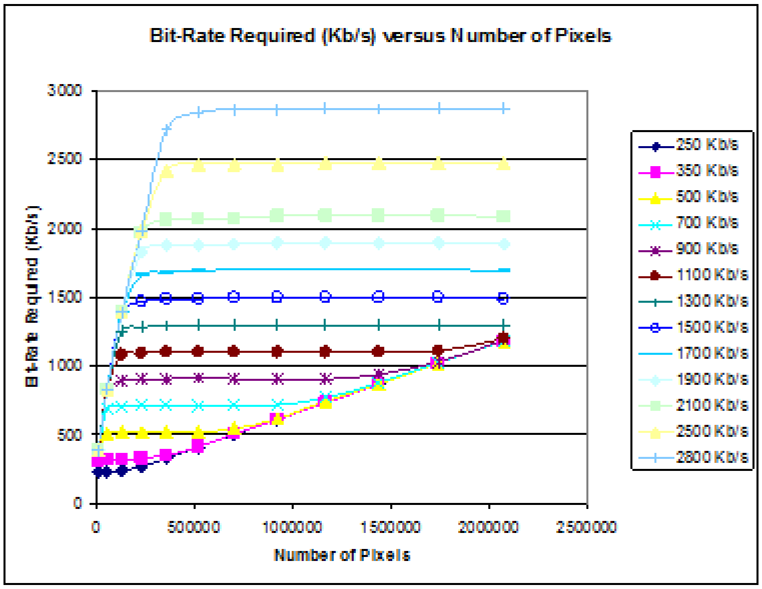

Figure 3. Multiple Transcodes of One Video Media File

Figure 3 graphs the resulting bit-rate of numerous encoding trials for the same video media. Each line represents a specific bit-rate parameter where the resolution parameter was varied and corresponds to the number of pixels identified in the graph. The Y-axis is the resulting average bit-rate of the media and the X-axis is the number of pixels of the resolution selected for the transcode.

The region to the left of the horizontal region is a saturation region because the number of bits required to represent the video information reaches a saturation point. The saturation area indicates that these smaller resolutions do not require the addition of more bits to encode the information and is reflected by a resulting lower average bit-rate than the parameter selected for encoding. The horizontal region shows the convergence of the transcode toward the bit-rate parameter selected for the encoding process. The diagonal region to the right of the horizontal region indicates a shortage of bits in the encoding parameter to provide higher resolutions at the selected bit-rate parameter.

With respect to the framework, it is the source agent’s responsibility to select the transcoding parameters and account for bandwidth constraints. The resolution parameter selected for a given encode can greatly affect the quality of the resulting viewing experience at the destination.

It is desirable for the agents to set the encoding parameters to the maximum resolution of a given destination device, but as shown in Figure 3, if the resolution of the destination device is to the right of the horizontal region for a given bit-rate parameter, the resulting bit-rate will be higher than the parameter selected. This translates into a failed real-time viewing experience because the available bandwidth is lower than the resulting transcoded bit-rate. The media transfer will not be able to keep up with the encoded bit-rate. The bit-rate parameter selection is further complicated by the fact that different video media do not exhibit consistent encoding characteristics.

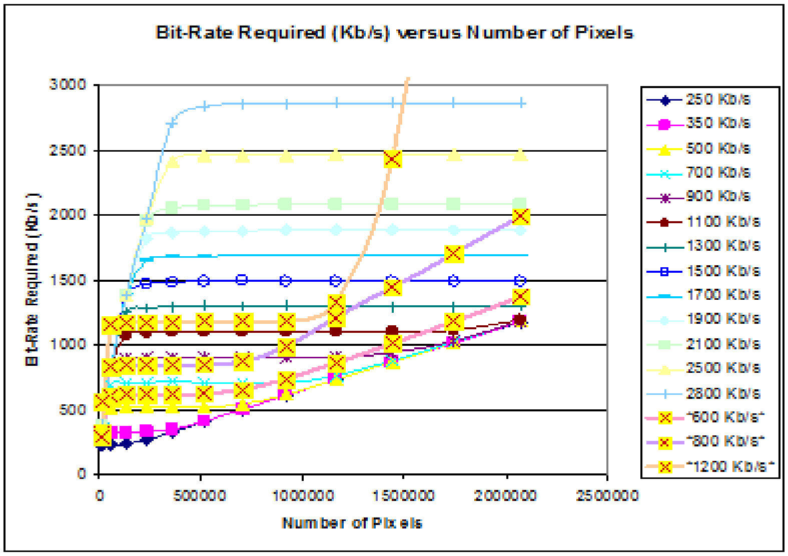

Figure 4. Multiple Transcodes of Various Video Media Files

Figure 4 shows the same graph as Figure 3 with overlays of different video transcoding lines as indicated by the lines and boxes with the X's. Different video media data exhibit similar characteristics for media encoding with respect to the three regions, but the size of the horizontal region varies. This depends primarily on the amount of changes and detail contained in the original video images per unit time.

The agents in the system are not able to predict the point at which the graph of a given media selection will go from the approximate zero slope of the horizontal region to a positive slope of the corresponding diagonal region. Despite the variable characteristics for encoding different media, the decision regarding the encoding of a video file is much less complex because of the known properties of the original media file and the known constraints of the situational context that is provided through agent execution at local devices.

The source agent is aware of the properties of the original media file and the resolution of the destination device. The two most important factors with respect to the parameters selected for real-time transfer are known imposed constraints. These factors are the bandwidth availability constraints and the resulting resolution of the transcode. The following is a simple decision table based on these two important considerations:

Table 1. Agent Decision Table

BW available >= BW Orig.	Res. Dest. >= Res. Orig.	Action
Yes	Yes	Adaptation: Not Required
Yes	No	Adaption: Resolution decrease
No	Yes	Adaptation: Bit-rate decrease
No	No	Adaptation: Bit-rate decrease and Resolution decrease

For the first two cases, the bandwidth available is greater than the bandwidth required for transferring the media data. As such, the source agent does not have to make any bit-rate changes to the media. In the first case, increasing the resolution of the original file does not provide any new information. If the destination device has a higher resolution than the original media it makes little sense to increase the resolution from the original resolution. The media should simply be displayed at full screen on the destination device. In the second case, decreasing of resolution will likely result in a decreased bit-rate for the resulting encode and so the bandwidth constraint can be ignored and an equal bit-rate can be used.

It is in the last two cases where the real-time adaptation decisions need to be made with respect to the parameters selected for a resulting transcode. It is the parameters adjusted for bandwidth constraints that pose the greatest difficulty in determining the optimal parameters for transcoding. However, since the source agent inspects the properties of the original media before transcoding begins, it can determine the average number of bits-per-pixel used by the original media file given its current resolution and codec.

It is these properties that provide bounding constraints for agent decision making which significantly reduces the complexity of the decision. The upper bound of the bit-rate parameter is determined through bandwidth estimation and therefore the only decision left to be made is the resulting resolution. The evaluation of the bits-per-pixel of the original file also provides a bounding constraint for the source agent. A decrease in the resulting average bits-per-pixel may force the transcode to the right in the standard encoding graph and into the diagonal region.

This would increase the resulting bit-rate and not meet the real-time constraints of the bandwidth available as discussed previously. The resulting resolution is calculated by determining the maximum resolution that can be achieved which results in the number of bits-per-pixel being greater than or equal to the observed number of bits-per-pixel found in the original file. It is generally understood that lower bits-per-pixel values are a result of similar quality higher resolution encodes. The lower bound constraint of the bits-per-pixel of the original media ensures the resolution cannot be towards the right-side of the graph at a lower resolution.

6. CONCLUSION

This paper presents a distributed framework for media access. Mobile agents use their unique properties to provide delivery methods and access to desired content on distributed devices.

The framework provides access to distributed content through a web-browser interface that is used to send media from one device to another. These devices can be local devices or devices across private networks.

The prototype functions as discussed but there still exists several opportunities for further research and development. The framework could be more standards compliant by utilizing well established protocols like UPnP (universal plug-n-play) to provide integration into other IP service systems. The framework provides a discovery mechanism for media available but does not include the ability for an automated search of media. The user is expected to know what device the desired media is on. Other opportunities include further optimization of the current system and further authentication through user location technologies.

REFERENCES

[1] Bradshaw, J (editor). “Software Agents,” AAAI Press /MIT Press, Cambridge, Massachusetts, 1997.

[2] Wooldridge, M. and Jennings, N. R., “Intelligent agents: Theory and practice,” Knowl. Eng., Rev. 10, 2, 115–152, 1995.

[3] Harrison, C. G., Chess, D. M. and Kershenbaum, A., “Mobile Agents: Are They a Good Idea?” IBM Research Report, 1995.

[4] Gray, R. S., Cybenko, G., Kotz, D. and Rus, D., “Mobile Agents: Motivations and State of the Art,” Handbook of Agent Technology, J. Bradshaw, ed., 2001.

[5] Tozicka, J., Jakob, M.and Pechoucek, M., “Market-Inspired Approach to Collaborative Learning,” Cooperative Information Agent X, 2006.

[6] Dutta, P. S., Jennings, N. R. and Moreau, L., “Cooperative information sharing to improve distributed learning in multi-agent systems,” Journal of AI Research 24 407-463, 2005.

[7] Günter, M., Braun, T., “Internet Service Delivery Control with Mobile Agents,” Extended Abstract for the 2nd International Symposium on Agent Systems and Applications / Mobile Agents, September 2000, Zürich, Switzerland.

[8] Kotz, D. and Gray, R. S., “Mobile agents and the future of the Internet,” SIGOPS Oper. Syst. Rev. 33(3):7–13, 1999.

[9] Franklin S., Graesser A., “Is it an agent, or just a program? A taxonomy for autonomous agents”, Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996

[10] Rosenberg, J., Huitema, C. and Mahy R., “Traversal Using Relay NAT (TURN)”, Internet Draft, draft-rosenberg-midcom-turn-08, Sept. 2005.

[11] Rosenberg, J., Weinberger, J., Huitema, C. and Mahy R., "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.

[12] Satoh, I., “Building reusable mobile agents for network management,” Systems, Man and Cybernetics, Part C, IEEE Transactions on Volume 33, Issue 3, Page(s):350 – 357, 2003.

[13] Chen, R.Y. and Yeager, B., “Java mobile agents on project JXTA peer-to-peer platform ,” Proceedings of the 36th Annual Hawaii International Conference 2003 Page(s):10 pp., System Sciences, 2003.

[14] Gendelman, E., Bic, L., Dillencourt, M.B., “Fast File Access for Fast Agents,” Int'l Conf. on Mobile Agents, Atlanta, GA, Dec. 2001.

[15] Villate, Y., Illarramendi, A. and Pitoura, E., “Data Lockers: Mobile-Agent Based Middleware for the Security and Availability of Roaming Users,” Data In CoopIS, 2000.