2009

Scalable Architecture for Distributed Video

Stefan Karapetkov
Polycom, Inc., stefan.karapetkov@polycom.com

Scalable Architecture for Distributed Video

Stefan Karapetkov
Polycom, Inc.
Stefan.Karapetkov@polycom.com

ABSTRACT

As video applications become more important in organization's communication, they require a new kind of architecture that meets the scalability requirements. Video applications are distributed in nature, and run almost exclusively over IP networks today. This paper investigates the architectural approaches for creating a scalable video network, and discusses the key potential bottlenecks in performance that the architecture has to address. Due to the limited size, the paper may not be able to cover scalable recording, streaming, firewall traversal, and integrations with scheduling and management applications. Since this content exists, the outstanding issues will be addresses during the presentation and in the Q&A session.

Keywords

scalability; video application; telepresence; desktop video; video network; video gateway; directories; video applications

INTRODUCTION

Video is leaving the video conference room and becoming a standard communication tool and part of the individual's daily workflow. In turn, this trend is having a profound impact on the scalability requirements of the visual communication system which now must support tens of thousands of users where once it only had to support several dozen or at most a few hundred video rooms.

Distributed video is a solution and an architecture that combines the quality of room-based telepresence and the usability (ease of use) of simplified desktop video. This paper discusses approaches to creating a scalable, distributed video architecture to support the requirements of video-enabled organizations. The scalability mechanisms we discuss can also be deployed by service providers who offer video services.

Critical components of the visual communications system are the communication server, the conference server (MCU), and the gateway to other networks. A complete solution requires directories and the ability to integrate with third-party applications, while presence has become an important functionality defining Unified Communications, and at the same time is considered a core requirement for communication systems. Therefore, this paper will address not only the scalability of communication servers, conference servers, and gateways, but also the scalability aspects of directories and presence servers and the means for integration with third-party applications.

SCALABLE COMMUNICATION SERVER ARCHITECTURE

The heart of any communication system is the communication server. There are many names for it—call manager, communication manager, Gatekeeper, SIP[1] server, IP-PBX and so on—but they all perform essentially the same core tasks. Traditional communication servers keep track of all communication endpoints (terminals) in the network and provide services according to a profile set up for a particular endpoint, for example, endpoints installed in corporate lobbies may be restricted from making external calls. Modern communication servers recognize users (through a logon/authentication procedure) and can apply policies to the user instead of the endpoint. For example, a support technician in customer service might be authorized to place high definition video calls while an accountant in the finance group might only be able to place standard definition video calls.

Communication servers process calls among endpoints, keep track of the call state, and interact with the endpoints in the network to provide logical prompts and options to users. Communication servers keep records for each call (call detail

records) which are often used for interdepartmental accounting and for billing (in case the communication system is shared among several companies or a service provider is deploying it to offer services).

What makes a communication server scalable? Scalability is basically the ability to serve more users; that is, if one server can support a maximum of 1,000 users and another server can support 10,000, the second server is 10 times more scalable than the first. However, maintaining 1,000 or 10,000 users in the user database of the server is usually not an issue. What really limits scalability is the amount of calls per second that the server can process. Statistically, when more users are registered with the server, more calls are placed. If the number of calls per second exceeds the maximum supported by the server's architecture, the server slows down and starts rejecting or dropping calls. The number of calls per second that a server can process depends mainly on the complexity of the networking protocols.

Endpoint Intelligence

Protocols between an endpoint and server can be stimulus or functional. For example, proprietary signaling protocols used in legacy PBXs are stimulus protocols and some standard protocols such as MGCP[2] are stimulus, too. Stimulus protocols are used to keep endpoints simple and inexpensive. Information such as the endpoint's profile (number of keys, size of display, for example) and the call state information (whether the endpoint is on-hook or off-hook or whether there is an active call, for example) are kept in the server. If the user lifts the handset or presses a key, the endpoint sends a code to the server. The server then interprets the user's action and sends instructions to the endpoint on how to respond, that is, what string of symbols to show on the display. Therefore, the system fully controls the endpoint and can place calls, answer calls, and perform all call features on the endpoint's behalf. Since all information about the device is centrally stored in the communication server, scalability of such systems is limited.

Standard protocols such as H.323[3] and SIP on the other hand are functional, that is, the endpoint itself is intelligent and can, for example, place calls and initiate transfers based on user input. The server only receives signaling messages, executes them, or passes them to other network elements that can execute them. Putting intelligence in the endpoints allows communication servers to be simplified and made more scalable.

Lightweight Protocols

Lightweight protocols, such as SIP, require fewer messages to setup and tear down a call. The server also has to store less call state information. This automatically increases the scalability of the communication server.

The Lightweight Directory Access Protocol (LDAP)[4] is another example for a such protocol and is used by endpoints and communication servers to retrieve information from directories. The directory is the list of all users with their contact information, access rights, etc. If the directory is embedded in the communication server, the complexity of the directory access protocol directly affects server performance, thus using LDAP is a requirement for a scalable communication system. To increase scalability and decrease the load on the LDAP server, some of the information is cached and updated periodically.

Separating Signaling and Media

Media includes the audio and video streams among endpoints. Audio is typically compressed using one of the standard G.7xx codecs, while video is usually compressed by one of the standard H.26x codecs. Processing media—and especially video media—is very resource-intensive; therefore, the best way to keep the communication server scalable is to process the media separately. This is possible with most modern protocols and both SIP and H.323 clearly separate signaling from the media. If the communication server processes signaling messages, but no media, its scalability is higher by several magnitudes. Figure 1 depicts an example of a point-to-point call between two endpoints.

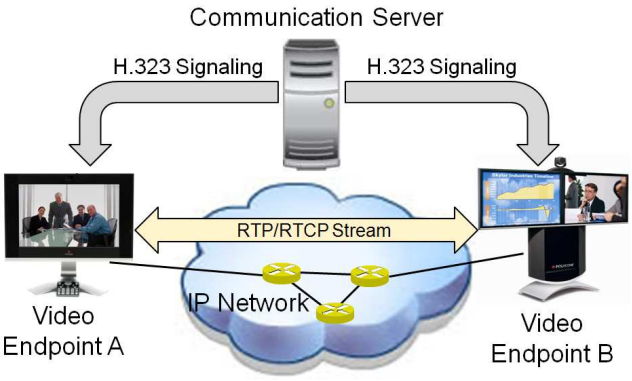

Figure 1: Scalability through signaling and media separation

The actual audio and video packets are transported over the Real Time Transport Protocol (RTP)[5] directly between the two network elements (endpoints, conference servers, or gateways) and assures the highest possible quality.

Single-protocol vs. Multiprotocol Servers

Single-protocol servers can inherently scale better than multiprotocol servers. This is because multiprotocol servers must translate every call from one protocol to another, that is, they have to understand both message formats and keep call state information for both call legs. Figure 2 depicts a multiprotocol server that supports H.323 and SIP.

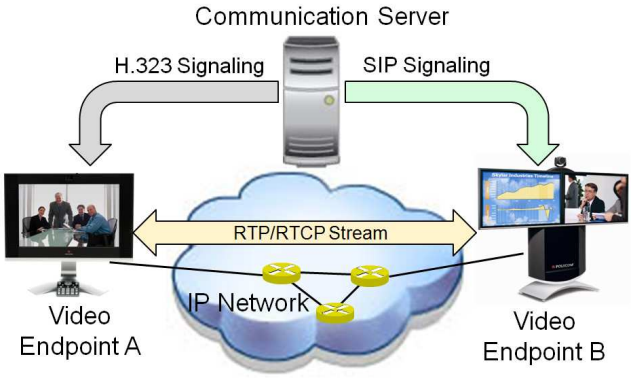

Figure 2: Multiprotocol server architecture

Why is a multiprotocol server more scalable than a single-protocol one? An analogy would be a group of people speaking the same language and another group of people speaking different languages using a translator. The second group will be slower in their discussion.

Server Networking

Following the guidelines discussed above allows creating a communication server that scales from 5,000 to 40,000 users. What if we need a solution for a company that has 100,000 employees worldwide? Additional scalability can be achieved through networking communication servers. Networking is connecting two or more communication systems through a protocol, that is, a common language understood by all systems in the network. Protocols for networking systems are a little different from the protocols between an endpoint and a server. The difference is mainly that in a system-to-system protocol each side is a server that is responsible for many endpoints. It is inefficient to exchange information about each endpoint associated with a server separately. The information is aggregated and communicated through call routing rules.

In H.323 networks, this is done by prefixes, that is, server A is configured to know that if a user dials '4' plus 5 digits, the destination is a user on server B. If the user dials only 5 digits, server A knows the call is local and will try to route it to the appropriate user on server A. SIP uses the domain name concept, similar to the e-mail system. Server A has its own domain, that is, serverA.organizationX.com, and server B has its own, for example, serverB.organizationX.com. User S (for 'source') on server A is identified by the address userS@serverA.organizationX.com while user D (for 'destination') on server B is identified by the address userD@serverB.organizationX.com. If user S dials the address of user D, server A will recognize the domain of server B and send the call to server B. Addresses in the above format are called Universal Resource Identifiers (URIs).

The question is how users will remember these numbers and prefixes (in H.323) and URIs (in SIP). Fortunately, they do not need to do so because this information is stored in the directory and can be accessed, searched, and selected (clicked on) to place a call. Figure 3 depicts server networking and the use of directories.

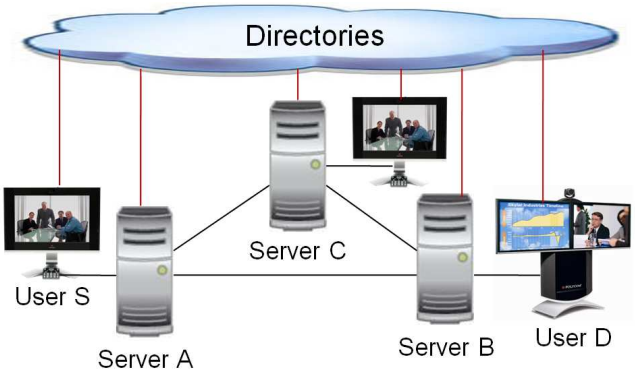

Figure 3: Scalability through server networking

Each of the networked servers as well as the individual endpoints can access directories independently.

Server Pool

In the discussion above, each of the communication servers in the network has different users. Another way of deploying multiple servers is as a redundant pool of resources that serve the same but larger group of users. This configuration has excellent redundancy with two servers and can have even higher redundancy with three or more servers. If one server fails or is taken down for maintenance (for example, for upgrading its software), incoming calls are simply routed to another operational server. Figure 4 shows the configuration with two servers.

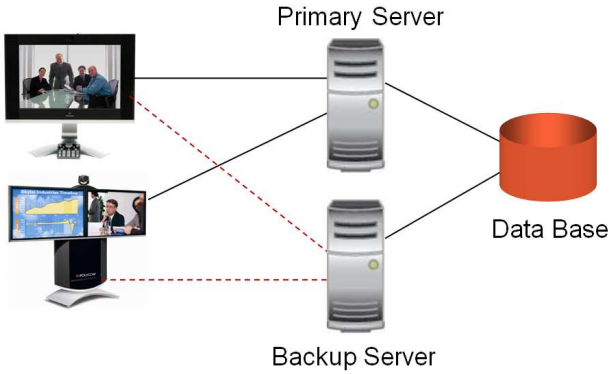

Figure 4: Server redundancy and load balancing

The two servers which are deployed in a cluster configuration; additional servers could be added if needed. The database engine and storage are deployed on both servers and are kept in sync in real time using a data replication mechanism. This approach provides a simple deployment scenario and removes the single point of failure.

SCALABLE CONFERENCE SERVER

The conference server, also called Multipoint Conferencing Unit or MCU in the H.323 architecture, is the main component for multipoint calls. It receives audio and video streams from each endpoint participating in the conference, combines multiple images into one (this technology is known as Continuous Presence) and sends the combined image to the participating endpoints. The conference server can translate the audio and video from one format to another, for example, it can receive video in H.264 and send video in H.263 format, receive audio in G.722.1 and send audio in G.711 format.

This function is known as transcoding and requires significant computing resources (typically through Digital Signaling Processors (DSPs). This is especially true for video because it involves decoding the digital video stream from one format into uncompressed video and then encoding it in another format. Scalability can be increased by using conference servers in video-switched mode which circumvents transcoding (and therefore the server needs far less computing resources) but also limits the flexibility because all parties have to use the best common codec, resolution, and bit rate.

The external interfaces of the conference server require very high input and output speeds for the multiple audio-video streams. For example, if a server supports 80 participants @ 4 megabits per second each (normal speed for High Definition video with High Definition content sharing), the conference server must support 80*4 or 320 megabits per second input (from endpoints to server) and another 320 megabits per second output (from server to endpoints). Internally, the server works with uncompressed video which takes many gigabits per second on the internal interfaces, and requires fast internal communication links.

Scalable Hardware Architecture

One way to achieve scalability in the conference server is through deploying scalable hardware architecture. For example, Advanced Telecommunications Computing Architecture (ATCA or AdvancedTCA) is designed to meet the requirements for the next generation of "carrier grade" communications equipment, and is specified by the PCI Industrial Computer Manufacturers Group (PICMG). ATCA is standard blade architecture, that is, the standard ATCA blades are plugged into a standard ATCA chassis. This architecture delivers the high speed external interfaces, the even higher speed internal interfaces, and large blades with ample space, electrical power, and cooling capacity to accommodate an array of DSPs necessary to process video. Figure 5 depicts different ATCA form factors for conference servers, using from 2 to 14 blades.

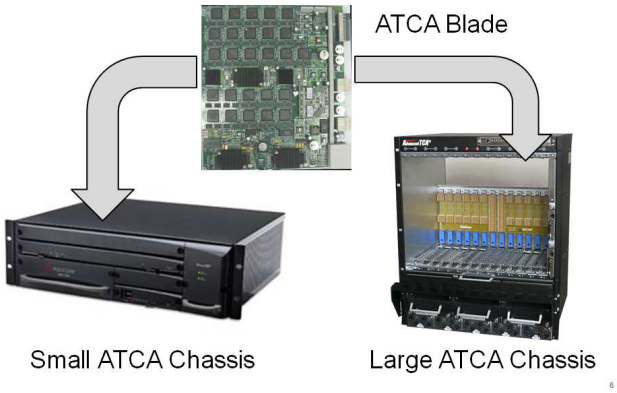

Figure 5: Conference server HW scalability

ATCA allows for building larger and even more powerful servers. The existing ATCA blades can be used in larger chassis hosting up to 14 blades. Note that an alternative path to scalability is through stacking of small conference servers. However, this approach introduces additional delay since media travels from one server to another through external interfaces. The stacking approach is also inefficient from power consumption and cooling perspective because each server has separate power supply unit and separate fans for cooling. The “green” aspects of Polycom’s ATCA technology are discussed in detail

in the article, "AdvancedTCA—Green Technology for Data Centers' in *CompactPCI and AdvancedTCA Systems Magazine*."[6]

Cascading Conference Servers

When the scalability of a single conference server is exhausted, multiple servers can be connected through so called 'cascading' to handle larger number of conferences and participants.

Cascading is a mechanism by which one conference server creates a link to another conference server. This is necessary, for example, when more participants want to join a conference than resources are available on any of the single servers. The conference server, or an application monitoring the server, recognizes that participants on two or more conference servers have joined conference with the same conference identification and password. It then creates a link between the servers, thus connecting all participants in a single conference. The speed of creating the cascading link is critical (ideally, this process should be hidden from the user) and the capability to mask the additional delay from the additional (cascaded) call leg. Another technical issue is the picture in picture in picture effect from multiple Continuous Presence instances.

Centralized Conference Resource Management

To make a pool of conference servers behave as one huge conference server, we need a resource management application that tracks the incoming calls, routes them to the appropriate resource (for instance, this can be done based on available server resources but also based on available bandwidth to the location of this server) and that automatically creates cascading links if a conference overflows to another server. If the conference is prescheduled, the application server can select a conference server that has sufficient resources to handle the number of participants at the required video quality (bandwidth). Overflow situations are probable with ad-hoc conferences where participants spontaneously join without any upfront reservation of resources. Figure 6 shows an application that manages distributed conference resources: three conference servers.

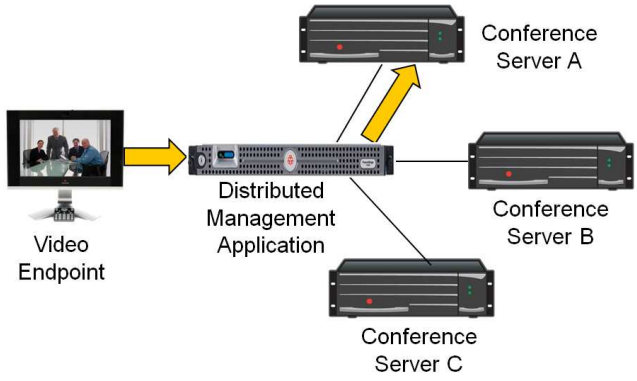

Figure 6: Managing distributed conference resources

The management application is designed to provide uninterrupted service by routing calls around failed or busy media servers. It also allows to "busy out" media severs for maintenance activities. From the user point of view, the service is always available. The system can gradually grow from small deployments of 1-2 media servers to large deployments with many geographically dispersed media servers. System administrators can monitor daily usage and plan the expansion as necessary. This approach also provides a centralized mechanism to deploy a front-end application to control and monitor conferencing activities across all media servers.

The management application can also act as a load balancer in this scenario, that is, it can distribute the load over a group of conference servers. The larger the resource pool, the more efficient the load balancing function is, a feature that is very important to large global organizations with offices and conference servers spread around the world. The same technology can be used by service providers who can offer conference services globally by using the Distributed Management Application and deploying conference servers in central points of the network.

The scenario works well in architectures such as SIP, where the Registrar function is separate from the Proxy function, that is, where the endpoint is registered with a SIP Registrar in the network but sends its calls to a pool of SIP Proxies.

SCALABLE GATEWAYS

Gateways are the gates to other networks. If we assume H.323 deployment, gateways will be necessary to connect to the SIP or ISDN systems. For connectivity to mobile video deployments, a gateway to H.324M may be required. Gateways are especially important when a new technology is rolled out, e.g., when new SIP systems are installed, because most of the users you want to talk to will likely still be using legacy systems. Most of the calls in these early stages will therefore be gateway calls. Gateways are not important in green-field installations (those with no legacy equipment or when connection to outside legacy systems is not desired) or when the new network has reached critical mass and most of the calls stay within the same domain/protocol.

Signaling Gateway

If a gateway is required, the scalability of the gateway is critical in the early day of deployment of new technology. As with communication servers, the best way to achieve scalability here is through separating signaling from media and limiting the gateway function to signaling only. In this case the gateway is no different from a multiprotocol communication server. It receives messages in one format (SIP, for example) and translates them into another (for instance, H.323) and vice versa. This architecture does not allow the gateway to scale to the levels of a single-protocol communication server but it can handle much higher load than if media is involved.

Just as an example for the performance impact, if the single-protocol communication server scales to 30,000 users, then adding the support of a second protocol (in effect, creating a signaling gateway) may reduce the scalability to 3,000 simultaneous calls. If media processing is added to signaling processing, scalability may drop to 300 simultaneous audio-only calls or to 30 simultaneous audio-video calls.

A signaling gateway is only a feasible solution if the media (audio and video) is in the same format. For example, both H.323 and SIP use the Real Time Protocol (RTP)[5] for media and therefore are candidates for signaling-only gateway interoperation. Figure 7 depicts the configuration.

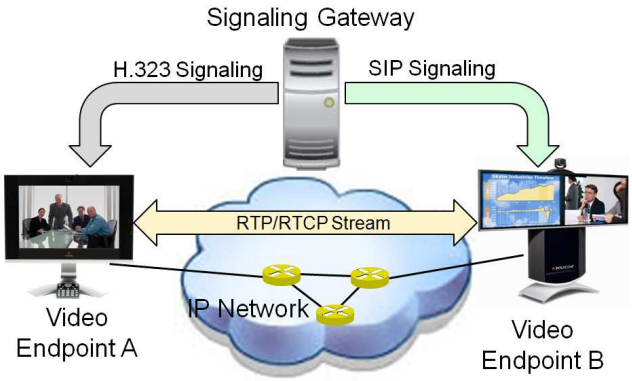

Figure 7: Signaling gateway between SIP and H.323

Media Gateway

Using a media gateway helps overcome the security problem and gives network administrators more flexibility during the transition from one protocol to another. It does limit the scalability since the media gateway often needs to transcode video, that is, it requires DSPs and fast external and internal interfaces.

Similar to conference servers, media gateways can scale by avoiding transcoding. The media gateway controls the communication with the endpoints, and transcoding is only necessary if the endpoints negotiate different audio/video algorithms, resolutions, and bit rates. If the gateway enforces the same audio/video algorithm, resolution, and bit rate between the endpoints, then no transcoding is necessary.

Therefore, the media gateway is very similar to a conference server, and if a call goes through a conference server and through a gateway (see Figure 8) it may be transcoded twice which typically results in decreased picture quality. Why is this? Let's remember the analogy with language translation. If you translate from English to German, you lose some information

but the quality is still acceptable. If you then give the German version to someone else to translate it into Russian, the final version will be farther from the original, and probably not acceptable.

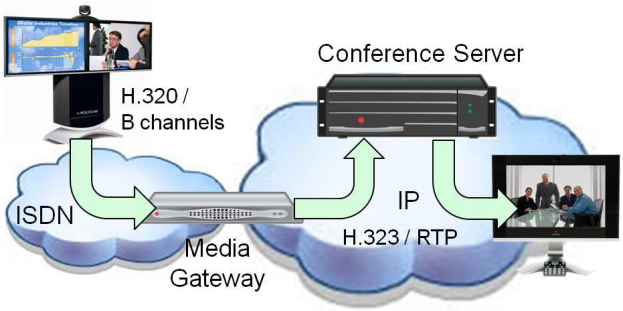

Figure 8: Media gateway configuration

Therefore, the logical question is, why not use the conference server as a gateway? It already must be in the network, and it does have the required functionality to support multiple protocols. Modern conference servers support H.323, SIP, ISDN, and PSTN. Note that a media gateway (with transcoding) is a must when the connected networks have completely different physical layers, H.323 to ISDN or SIP to H.324M, for instance. The only disadvantage of using the conference server as a gateway is the relatively high price per port/resource.

PRESENCE SCALABILITY

Over the last few years, presence became entrenched in business communications, mainly through the use of Microsoft Office Communications Server and IBM SameTime. In industry discussions, presence is always cited as a key component of unified communications.

Presence is delivered through client-server architecture; XMPP[7] and SIMPLE[8] are the two prevailing protocols for implementing it (Figure 9). The user interacts with the presence client which communicates with the presence server. The server keeps track of the presence status for all users on the system and can obtain presence information for users on other presence systems through the so-called federation (a form of networking). As with other servers, there are two ways to scale: create a scalable presence server that can handle tens of thousands of users or interconnect multiple presence servers in a network.

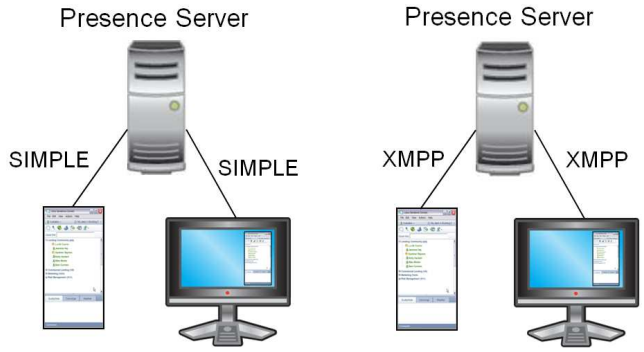

Figure 9: Scalable presence server architectures

With the ever-growing number of tools for automatic changes of the presence status and the growing number of contacts that users add to their buddy lists, presence servers are required to handle many frequent updates and communicate them to a large group of users. The server must then keep larger tables and communicate the change of presence status to larger group

of clients. When the scalability of a single server is exhausted, networking techniques such as federation are used to support larger deployments.

Note that while Instant Messaging is usually mixed with presence, it is a completely separate functionality that does not necessarily belong to a presence server.

Federation

Federation is a trust relationship between presence servers that allows them to exchange information about the presence status of their users. Figure 10 shows federation relationship between two presence servers.

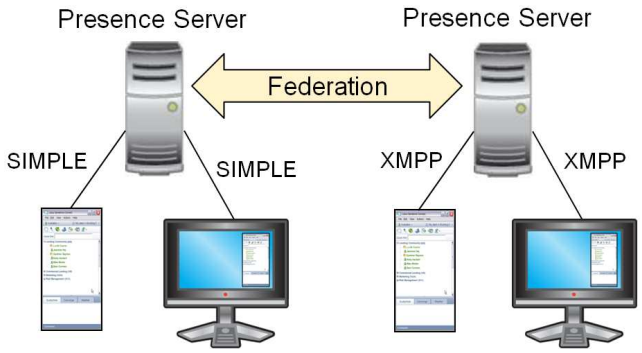

Figure 10: Scalability through federation

The term 'federation' is sometimes used for more than just exchange of presence information, e.g. exchange of directory information, gatekeeper neighboring information and licensing information may be also called 'federation'.

From the two standard protocols for presence—XMPP and SIMPLE—XMPP has found wider adoption in the Internet and this indicates that higher scalability is expected from this protocol. XMPP server federation follows the proven and scalable model of Internet e-mail which meets the needs of the individual domain for flexibility and control. Each XMPP domain can define the level of security, quality of service, and manageability that make sense for the organization. Exchange of presence information within one XMPP domain is accomplished through the XMPP server in this domain. Through federation, the server exchanges presence information with peer XMPP servers in other organizations.

SCALABLE DIRECTORIES

As discussed in the section on the communication server, directories solve the problem with different dialing formats. Many corporate IT organizations have been converging dozens of directories into one directory structure that allows changes (adds, moves, and deletes) to be automatically propagated to applications across the organization. The goal is to be able to add or remove an employee in one master database and have all tools that employee uses (e-mail, phone, presence/IM, and Web) automatically learn about the change. The trend towards closer integration of visual communication in the corporate IT environment leads to the need for integration with the organization's directory.

LDAP[4] emerged as the standard for accessing directories and almost all directories today have an LDAP interface. The ITU-T H.350[9] standard describes a LDAP schema for visual communication users, that is, H.350 describes how to store video-specific parameters into an LDAP database. Figure 11 describes the configuration.

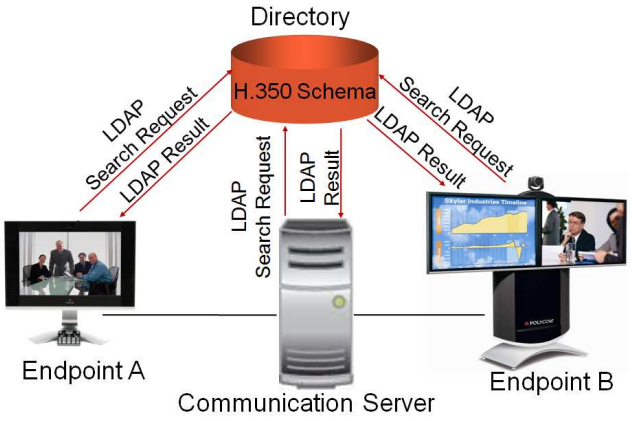

Figure 11: Directory access mechanism

LDAP and H.350 are supported in many popular directories, such as Microsoft Active Directory and Sun OpenLDAP.

Dedicated Video Directory

Video directories will eventually fully integrate with IT directories that keep user information for network access, Virtual Private Network (VPN) access, e-mail, Web access, and so on. This integration, however, will be phased in through multiple stages. In the first stage, a dedicated video directory will communicate with the IT directory using the standard LDAP protocol.

There are two reasons for using a dedicated video directory. First, endpoints today use pre-LDAP protocols. Keeping the video directory separate allows support of these protocols and, therefore, of legacy endpoints. Second, including a directory with the communication server makes a lot of sense because organizations may want to pilot new technology and will want to get the system up and running quickly, without immediate integration with the corporate IT directory. Third, organization IT directories may not be yet configured for the H.350 schema (although they all have the capability to support H.350 schema). Figure 12 describes the network configuration with dedicated video directory.

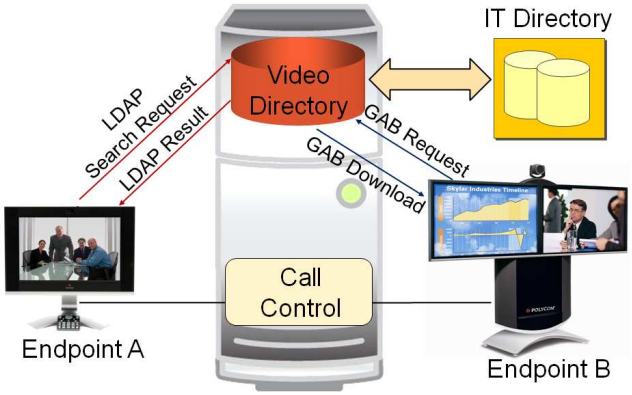

Figure 12: Using a dedicated internal directory connected to IT directory

Once the system is approved and a decision is made to integrate it with the IT directory, the dedicated video directory can be connected to the organization's IT directory. If there is no need to support legacy endpoints, the dedicated directory can be turned off and only the main IT directory can be used (see Figure 13).

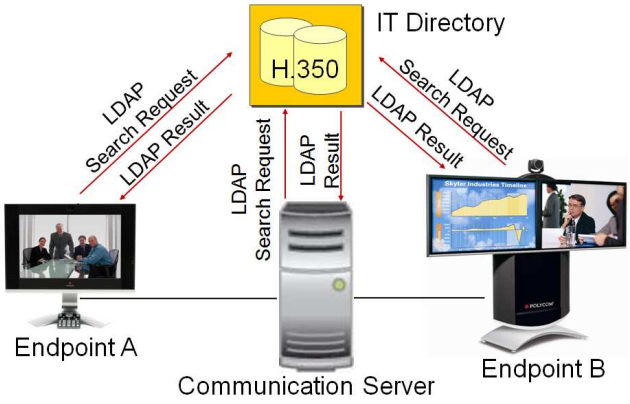

Figure 13: Direct integration with corporate IT directory

The IT directory is scalable as it includes all employees with their profiles. It is important to understand the impact of the H.350 extensions to the scalability of the directory.

Depending on the implementation, the directory may be queried more or less frequently. LDAP is really not designed for fast lookups and the directory behind the LDAP server component maybe an old directory running on a mainframe. It is therefore a best practice to cache the information in the communications server or in the endpoints for a predefined period of time, so that frequent multiple queries for the same information are avoided.

CONCLUSION

Our vision for the future stresses the importance of scalability for transforming today's video conferencing into tomorrow's visual communication which will be scalable, reliable and deeply integrated with the core IT infrastructure.

Scalability is a very complex topic and touches on all system components. Any of the components has the potential to create a bottleneck and limit the system performance and the scale of the system. The technologies discussed in this paper overcome these bottlenecks and are the solid foundation of a scalable visual communication system.

ACKNOWLEDGEMENTS

I would like to thank my colleagues Rick Flott, Piotr Drozdzewicz, and John Antanaitis for their contribution to this work.

REFERENCES

1. Rosenberg, J. et al (2002) SIP: Session Initiation Protocol. Request For Comments (RFC) 3261, Internet Engineering Task Force (IETF) Document, June 2002. http://www.ietf.org/rfc/rfc3261.txt

2. Arango, M. et al (1999) Media Gateway Control Protocol (MGCP) Version 1.0. RFC 2705, IETF Document, October 1999. http://www.rfc-editor.org/rfc/rfc2705.txt

3. ITU-T Recommendation (2006) H.323 v6 - Packet-Based Multimedia Communications Systems. November 2000. http://www.itu.int/itudoc/itu-t/aap/sg16aap/recaap/h323/h323v6.html

4. Zeilenga, K. (2006) Lightweight Directory Access Protocol: Technical Specification Road Map. RFC 4510, IETF Document, June 2006. http://www.ietf.org/rfc/rfc4510.txt?number=4510

5. Schulzrinne, H. et al (2003) RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF Document, July 2003. http://www.ietf.org/rfc/rfc3550.txt?number=3550

6. Karapetkov, S. (2008) AdvancedTCA - Green Conferencing for Data Centers. Article in CompactPCI and AdvancedTCA Systems. http://www.compactpci-systems.com/articles/id/?3104

7. Saint-Andre, P. (2004) Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921, IETF Document, October 2004. http://www.ietf.org/rfc/rfc3921.txt?number=3921

8. Campbell, B. (2002) Session Initiation Protocol (SIP) Extension for Instant Messaging. RFC 3428, IETF Document, December 2002. http://www.ietf.org/rfc/rfc3428.txt?number=3428

9. ITU-T Recommendation (2003) H.350 : Directory services architecture for multimedia conferencing. August 2003. http://www.itu.int/rec/T-REC-H.350