A Secure and Scalable Video Conference System Based on Peer-assisted Content Delivery Networks

Hao Yin[1], ChangLai Du[1], Chao Ren[2], Zhijia Chen[1], Geyong Min[3], Chuang Lin[1]
[1]Computer Science and Technology Department, Tsinghua University, China
[2]School of Computer Science, Northwestern Polytechnical University, China
[3]Department of Computing, University of Bradford, U.K.
[1]{ hyin, dcl, zjchen, clin }@csnet1.cs.tsinghua.edu.cn, [2]renchao@cdqdc.edu.cn, [3]g.min@brad.ac.uk

***Abstract*—** With the ever-increasing demand for multimedia conferencing over the Internet, video conferencing systems that enable efficient and secure data delivery have been a hot yet challenging research topic in both academy and industry. To deploy large-scale commercial video conferencing services, we need to address critical issues, such as Quality-of-Service (QoS), security and scalability. In this paper, we propose a secure and scalable video conferencing system over the Internet, namely Video Conference Network Platform (VCNP). VCNP organizes users into a Chord-based Peer-to-Peer (P2P) overlay, and utilizes Peer-to-Peer Session Initiation Protocol (P2PSIP) for the management of conference members, which includes the setup of media sessions and the storage of user contact information and conferencing group information over distributed peers. A VPN session over P2PSIP is implemented to act as a secure infrastructure for VCNP. For video data delivery, we design a framework of Peer-assisted CDN (Content Delivery Network), which uses both edge servers and fiber network in CDN, and the principle of structured P2P management to enhance the QoS, scalability and flexibility of the system. Based on the proposed architecture and protocols, VCNP has already been implemented and deployed over the Internet. Experimental results demonstrate the effectiveness of VCNP in delivering the unprecedented security, scalability, and certain QoS simultaneously in a unified architecture.

Index Terms—Video Conferencing, P2PSIP, VPN, CDN

1. INTRODUCTION

With the increasing demand for video conferences over the Internet, the researches on multimedia conferencing have attracted great attention from both academy and industry. There have been tremendous efforts towards the design and implementation of video conferencing systems in recent years; yet efficient and secure deployment in large scales remains an elusive goal [1]. Specifically, to deploy large-scale commercial video conferencing services, we face the following challenges:

First, Quality-of-Service (QoS) of the system must be guaranteed. Video conferencing has strict bandwidth, delay, and packet loss requirements. However, there is no QoS guarantee for interactive multimedia data (video, audio, text, *etc*) transmission over the current best-effort Internet [16-17]. To ensure that real-time audio and video to be delivered reliably, QoS mechanisms must be implemented at every part of the system and deployed all over the network.

Second, video conferencing is sensitive to security problems. Different from file sharing[32] or live media streaming[33] services, conferencing contents are mostly private or highly related to commercial secrets, so security mechanisms must be taken into account, both for the session setup procedure and for media contents delivery. However, this is quite challenging because of the inherently non-secure attribute of the Internet.

Third, the system must be scalable. With the increase of conferencing participants, a video conference system has to support a large number of users online simultaneously who are divided into size-limited conference groups. Its performance should not be degraded too much as the number of user increases.

The success of a large-scale commercial Internet video conferencing system is critically dependent on how well the above-listed issues are addresses, i.e., security, QoS, and scalability. This paper proposes a novel system named Video Conference Network Platform (VCNP), which logically organizes users into a structural P2P [2] overlay network and physically employs a peer-assisted CDN [19-20] for video delivery. Meanwhile, security mechanism is provided within VPN (Virtual Private Network) sessions using extended Session Initiation Protocol (SIP) [3]. To enhance the quality as well as scalability and flexibility of the system, the proposed peer-assisted CDN architecture consists of three layers: the *content transmission layer*, the *content distribution layer* and the *network access layer*. The content transmission layer is mainly composed of fiber networks with configured routers, the content distribution

layer contains hundreds or even thousands of Multipoint Control Unit(MCU) servers, and the network access layer is compose of the end users, possibly in millions.

The main contributions of this paper include: 1) to propose a novel video conferencing architecture, which utilizes the best feature of CDN, structured P2P and VPN technology, thus possessing the unprecedented capability of addressing the critical issues in video conferencing simultaneously, i.e., security, QoS, and scalability. 2) to develop a set of security management mechanisms which are implemented in our VPN session over P2PSIP; 3) to implement a prototype of the VCNP for successfully servicing an Internet video conferencing system over China. The real industry success demonstrates the effectiveness of VCNP and meanwhile gains valuable insights for large-scale commercial applications.

The rest of this paper is organized as follows. Section 2 reviews the background and related literature. Section 3 presents the control plane architecture of VCNP and Section 4 proposes the data plane architecture of VCNP. Section 5 specifies the implementation details. Section 6 presents the performance evaluation of the system. Finally, Section 7 concludes this paper.

2. RELATED WORK

Although there has been no shortage of efforts in academy and industry in recent years, the design of video conferencing systems that enable efficient and secure delivery of multimedia data is especially challenging due to the following unique characteristics of the applications [1]:

• *Performance requirements:* Conferencing applications require low latencies and need to sustain high bandwidth between the source and receivers. In contrast, other popular media applications like live media streaming is more tolerant to latency. In addition, conferencing applications deal with media streams that can tolerate loss through degradation in application quality. Table 1 lists the typical parameter values for a satisfactory video conferencing.

Table 1 Video conferencing network parameters

Parameter	Value
Packet Loss	<0.1%
Packet Latency	<=150ms
Packet Jitter	<40ms

• *Session lengths*: Conferences are generally long lived, lasting tens of minutes. In contrast, live streaming or Video-on-Demand (VOD)[31] users generally view the program in much shorter time duration.

• *Group characteristics*: Conferences usually involve small groups, consisting of tens to hundreds of participants. Membership can be dynamic. This is different with applications like live streaming, which may have millions of receivers at the same time.

• *Source transmission patterns*: Typically, conferencing applications have a source that transmits data at a fixed rate. While any member can be the source, there is usually a single source at any point in time. In contrast, large scale broadcasting applications may have a single static source throughout a session.

To address these unique features of video conferencing, there comes along with many technologies and solutions. In this section, we introduce the background technologies in designing video conferencing systems, and review the related literature on video conferencing systems with the aim of highlighting the difference between our proposal and existing solutions.

2.1. Content Delivery Network

A Content Delivery Network (CDN), which is formed by dedicated edge servers for content distribution, offers fast and reliable applications and services by distributing content to cache or edge servers located close to users[19][20]. There are two major advantages of using CDN: 1) scalability: user requests should be handled by all servers with a combined throughput and I/O greater than the single-server architecture; Meanwhile, it can help to avoid the congestion occurred in backbone network by pushing the content to the edge servers. 2) QoS, edge severs nearby users can provide shortened packet delivery paths thus providing better QoS (smaller network latency and lower packet loss rate).

In building CDNs, there are generally two approaches, e.g., overlay and network approach [21]. In the overlay approach, application-specific servers and caches at selected places in the network handle the distribution of specific content types (e.g. web content, streaming media, and real time video). In comparison, the core network components such as routers and switches, only need to provide the basic network connectivity and guaranteed QoS for specific request/traffic, relieving from the pressure of mass content delivery. Most of the commercial CDN providers such as Akamai [22], AppStream [23], and Limelight Networks [24] follow the overlay approach. These CDN providers replicate content to thousands of cache server worldwide. When content requests are received from end-users, they are redirected to the nearest CDN server, thus improving web site response time. As the CDN providers do not need to control the underlying network infrastructure elements, the management is simplified in an overlay approach and it opens opportunities for new services.

In the network approach, the network components including routers and switches are equipped with code for identifying

specific application types and for forwarding the requests based on predefined policies. Examples of this approach include devices that redirect content requests to local caches or switch traffic coming to data centers to specific servers (which is optimized to serve specific content types). Some CDN (e.g. Akamai, Mirror Image) use both the network and overlay approach for CDN organization. In such a case, a network element (e.g. switch) can act at the front end of a server farm and redirects the content request to a nearby application-specific surrogate server.

On the other hand, P2P technology has been widely used by some new entrants to the content delivery market. P2P technology allows for the exchange of content directly among network end users without the need for central servers, thus posing novel opportunities to scale the Internet for serving more users. In a large scale distributed network, multimedia object placement mechanisms must be in place and the authors in [30] and [34] addressed the problem for transparent data replication.

Our network architecture combines the technology features of Network-centric CDN, Overlay-based CDN model, and structured P2P model to achieve security, scalability, and QoS simultaneously. Configured edge servers are specially deployed at strategic locations around China and private optical fiber is served as a dedicated backbone. Meanwhile, we incorporate P2P technology to the architecture. On one hand, P2P enhance the scalability of the VCNP. On the other hand, the idea of structural P2P is involved to the group management of the VCNP. In VCNP, edge servers equipped with data transfer components act as Multipoint Control Units (MCUs). Those MCUs are aware of each other and can look up into the P2PSIP DHT to locate every active peer. Data delivering between MCUs can be in a peer-to-peer manner or through the fiber networks, the details of which will be discussed in Section 4 and 5.

Meanwhile, VCNP leverages P2P technology with modification to overcome certain limitations of P2P technology. Internet access providers in China often assign P2P traffic a low priority because P2P users consume large volumes of chargeable bandwidth but Internet access providers reap no direct monetary benefits from such traffic. Our VCNP adapts P2P technology with local proximity, allowing users only within the same geographical area to connect to one another. Since the Internet access providers generally do not experience congestion within their operated networks, our VCNP helps relieve the network traffic congestion typically caused by P2P networking.

2.2. Peer-to-Peer Session Initiation Protocol

Session Initiation Protocol (SIP) [3] is a widely-used application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences). SIP is a text-based protocol derived from HTTP; therefore SIP message types are similar to web messages. SIP messages make up the signaling portion of a multimedia session. Actual media stream travels directly between the endpoints, usually in forms of RTP packets. SIP is a component that can be used with other IETF protocols to build a complete multimedia architecture. Besides, SIP is open and has been widely implemented. There are open-source implementations from simple protocol stacks to fully fledged multimedia communication systems.

As an emerging open research, P2PSIP (Peer-to-Peer Session Initiation Protocol)[2] employs a set of protocol standards and mechanisms for using SIP in settings where the service of establishing and managing sessions is principally handled by a collection of intelligent endpoints, rather than centralized servers in current deployed SIP. As the benefits, P2PSIP overcomes the infrastructure independence on central servers, and enables simple discovery and setup while promoting scalability.

Specifically, P2PSIP peers manifest a distributed namespace in which overlay users are identified and provided mechanisms for locating users or resources within the P2PSIP overlay [2]. P2PSIP overlay is used not only to transfer session initiation messages but also to store user contact information. The overlay acts as a distributed database using hash algorithms to reflect a node's address or a user name into a hash namespace.

To enable a scalable and secure video conferencing system, in our proposed VCNP, P2PSIP is employed to organize and manage users and groups. P2PSIP is also responsible for session creation, maintenance and ending.

2.3. QoS Mechanisms for Video Conferencing

Quality of Service (QoS) mechanisms for video conferencing fall into two categories: network-oriented QoS and application-oriented QoS. Approaches may include priority queuing, application specific routing, bandwidth management, traffic shaping, etc. As the dominating approach, network-oriented QoS is implemented both at layer 2 and layer 3 in the Open System Interconnection (OSI) reference model. The representative ones are listed below.

Over Provisioning: The most direct way to provide QoS is to increase bandwidth and other resources so that queues on routers keep empty most of time and incoming packets will be forward without delay. Based on such philosophy, over provisioning is actually not a true QoS mechanism. With the increase of new bandwidth consuming applications and the burst nature of data applications, congestion will occur and it will cause jitter and packet loss.

IntServ [16]: The essence of IntServ is to reserve resources

for each individual flow so that the service quality can be guaranteed. Before starting the session, an application must specify its requirements and an admission control routine decides whether the request for resources can be granted [27]. This approach is similar to the traditional telephone switching infrastructure where resources are reserved for each call. One problem is the increased burden on the routers, which need to know and store the state of many flows at the same time. Moreover, all routers along the connection between the two end-points must be enabled to support protocols and procedures like RSVP, admission control and packet scheduling, which may result great difficulty in real deployment.

DiffServ [17]: DiffServ divides traffic into different classes and gives them differentiated treatment. To distinguish the classes, the Type of Services field is used in the IPv4 header. Using different classification, policing, shaping and scheduling rules, several Classes of Services can be provided. Since there are only a limited number of service classes, the routers only have to store data proportional to the number of classes instead of depending of the flow. DiffServ is therefore more scalable than IntServ and easier to implement. However, there is still an unsolved problem when high priority traffic concentrates on one router: the bandwidth can be so saturated that it adversely affects performance.

In our proposed system, proper QoS mechanisms are employed to suit the needs in large-scale conference system. On one hand, routers in fiber network are configured with DiffServ and MCUs are responsible for data classification and labeling. On the other, MCUs and peers also make decisions at application layer to choose a QoS satisfactory and the most cost-saving path. Details will be covered in section 4 and section 5.

2.4. Security Schemes for Video Conferencing

Security issues in video conferencing systems mainly include user authentication and content confidentiality. Content confidentiality can be ensured by encryption and the key point is session key distribution mechanism.

VPN (Virtual Private Network) can be used to create a secure, policy-based overlay network within the Internet. VPN uses the Internet as a transport while creating a secured tunnel within it. Setting up a VPN tunnel includes at least two steps: authentication and session key negotiation. PKI (Public Key Infrastructure) is the mostly used technique for authentication. Each user has a public/private key pair generated by Certificate Authority (CA). This key pair can then be used for encryption and signature.

L2TP [27] is a protocol that tunnels traffic over variety of networks. L2TP includes support for tunnel authentication, which can be used to mutually authenticate the tunnel endpoints. However, it does not define tunnel protection mechanisms. IPSec is a protocol suite which is used to secure communication at the network layer between two peers. L2TP/IPSec [28] VPN is the most popular remote access solution with high security performance. SSL VPN is much easier to manage and suitable for movable users to access private resources. Among the SSL VPNs, OpenVPN [7] is the first and most popular SSL VPN products. For a practical implementation, Polycom[8] proposed a solutions on how to deploy VPN within a corporate enterprise for video conferencing. However, the scheme proposed in that guide focuses on enterprise applications, and only supports site-to-site VPN.

In the proposed VCNP, MCUs connect directly to transmission layer routers and set up L2TP VPN tunnel to these routers. Thus all MCUs are in a private network, and the paths between every two MCUs are definite. As to the links between peers and MCUs, VPN sessions over P2PSIP are introduced to act as a security infrastructure for the system. A username/password scheme is employed for user login, a public/private key pair for authentication and to negotiate session key. A symmetric encryption algorithm is adopted to encrypt media data.

3. P2PSIP-BASED CONTROL PLANE

Logically, our system is composed of two layers: the control plane and the data plane. The control plane is to start up VPN and media sessions and to maintain these sessions. The data plane is responsible for efficient data delivery through network. In this section, we will first introduce the overview for the system architecture and then specify the design issues for control plane in VCNP.

3.1 System Architecture Overview

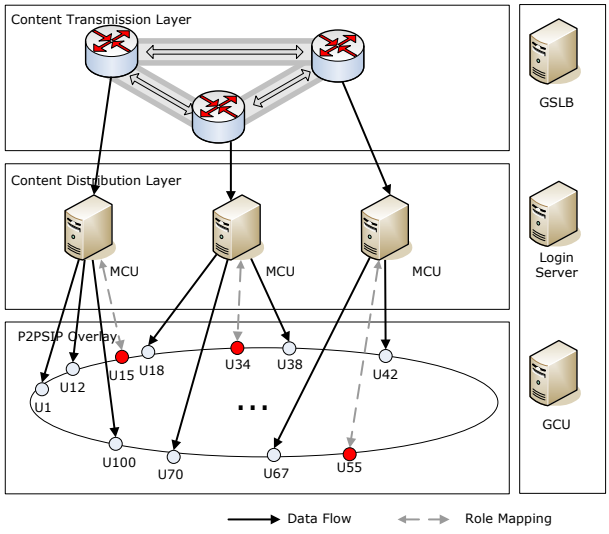

Figure 1 VCNP System Architecture

As shown in Figure 1, physically, our system consists of three layers: the *content transmission layer*, the *content distribution layer* and the *network access layer*.

The main functions of the *content transmission layer* are to transmit content and to optimize content routing. The content transmission layer is comprised of fiber networks with configured routers across the main ISP. Those routers are configured to support the differentiated service model defined by IETF. The content transmission layer can also solve the inter-connection problems among ISPs and facilitate prompt network resource sharing and balancing.

The second layer of our network architecture is the *content distribution layer*, composed of edge servers equipped with our data transfer components to act as Multipoint Control Units (MCUs) in video conference. With the guidance of Global Service Load Balancing (GSLB) policy in the central system, the content distribution layer allows end-users to connect to edge servers that are closer to them, in network terms, than our customers' own servers.

The MCU node can be mapped to a chord ring as a super peer in group. Its main responsibility include: 1) store the conference group information; 2) serve as the conference chair proxy for broadcasting information (user list, current session key, etc); 3) handle the data transfer for video data. The data transit through MCU helps to reduce the cross-ISP traffic and provide services with QoS guarantee. To achieve such a goal, two functions are enabled: application-layer routing and MCU multicast. For peers connected to the same MCU, if their link can meet the QoS requirement, data is directly delivered among them, instead of taking the resources of MCU or servers in content transmission layer. Otherwise, MCU takes the responsibility for data transfer among peers and meanwhile avoid traffic in the backbone network. In the addition, in case for data delivery for all participates, MCU merges the video of each user and broadcasts to everyone in the group.

The last layer of our network architecture is the *network access layer*, which consists of multiple public access networks within every geographic region, each belonging to different regional telecommunication operators. Our VCNP technology allows the exchange of content directly between network users without the need for central servers, as well as data transfer by the content distribution layer.

In deployment, to address the inappropriate interconnectivity and peering bottlenecks issue in the public Internet of China, we have developed a sophisticated infrastructure and an innovative technology platform for data delivery. Through our highly automated intelligent network with a dedicated fiber network, distributed edge servers, and advanced network operating and maintenance systems, we increase the level of interconnectivity and ensure the quality and reliability of our services. Details will be discussed in section 4.

3.2 Control Plane Architecture

Based on Chord [9], we design our underlying P2P overlay to store user contact and video group information, and to deliver sip messages. Chord [9] is a ring-based distributed hash table (DHT) for structured P2P systems. Chord is selected because of its simplicity, convergence properties, and general familiarity within the P2P community.

Figure 2 illustrates a basic control plane network structure for our VCNP system. In our system, end users/peers and MCUs in the same group together form a chord overlay, with peer storing the user information and MCU storing the group information. In the overlay, peers and resources are hashed into peer-ID and resource-ID respectively in the same namespace. Resource with resource-ID *k* will be stored by the first peer with peer-ID equal to or greater (mod the size of the namespace) than *k*, ensuring that every resource-ID is associated with some peer. As shown in Figure 2, peer with Peer-ID 47 is responsible for storing contact information of user 20. With the flow of arrows, this figure also shows a simple setting-up procedure for a Video Conferencing Group (VCG), which will be further specified in Section 3.3.

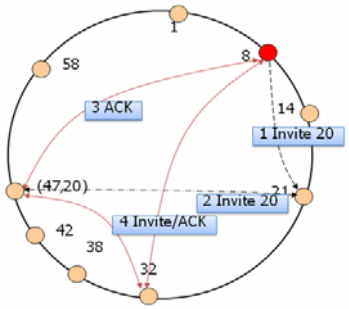

Figure 2 VCNP Control Plane Architecture

Generally, we define three types of IDs in our system:

• PID is the unique ID for a peer in the overlay, which is calculated using a Hash algorithm:

$$PID = Hash(peer\text{-}IP{:}port) \quad (1)$$

A peer with a PID is responsible for holding resource data assigned to it with the algorithm in Chord. Peers report their *PIDs* to *father MCUs.* A father MCU is the "nearest" MCU from a specific peer. The algorithm for finding father MCUs will be discussed in Section 4.

• UID is the unique user ID calculated using the same Hash algorithm:

$$UID = Hash(username) \quad (2)$$

Username is a SIP URI, and its definition can be found in later in this section. A <*username, IP:port*> is called the

user contact information, which is stored in responsible peers according to Chord algorithm.

• GID is the unique video conference group ID calculated using the same Hash algorithm:

$$GID = Hash(groupname) \quad (3)$$

A < *groupname, IP:port*> is stored in responsible MCUs according to Chord algorithm.

Definition 3.2.1: Username is a SIP URI. Here we assume Alice and Bob are users in the video conference. The email address of a user is chosen to become globally unique, e.g. *sip:alice@thu.edu.cn*. Using a valid email address as the username has other advantages. For example, it allows the system to generate a random password and email it to the user for authentication.

Definition 3.2.2: Groupname is also a SIP URI. But we include a 'isfocus' feature parameter in the groupname contact header field to express that the SIP dialog belongs to a conference, e.g., <*sip:alice@thu.edu.cn*>*;isfocus*.

Definition 3.2.3: User contact information is a key/value pair stored in the overlays distributed database. User contact in VCNP is defined in XML format, as is shown below

```
<key>sip:alice@thu.edu.cn</key>
<value >
<contacts>
    <contact displayName="Alice">
    sip:alice@166.111.1.2:5060
    </contact>
</contacts>
</value>
```

A user can have more than one contact. There is an added contact property "displayName" implying the user's human friendly name.

Definition 3.2.4: Group contact information is a list of users who are now in the group. Group contact in VCNP is defined in XML format, as shown below

```
<key>sip:alice@thu.edu.cn;isfocus </key>
<value >
<contacts>
    <contact displayName="Alice">
    sip:alice@166.111.1.2:5060
    </contact>
    <contact displayName="Bob">
    Sip:bob@59.66.134.121:5060
    </contact>
</contacts>
</value>
```

A user can have more than one contact. The group contact information will be updated as users join or leave the group.

3.3 VCG Set-up Process

We now specify the Video Conferencing Group (VCG) session set-up process in detail.

User registration: A user must first join the VCNP overlay and register its location. Actually, before the user can join the overlay, it must first login to a global *VCNP Login Server*. If the user passes the login check, it then gets *GSLB Server* addresses which will help it to find the nearest MCU and to join into the overlay following the rule in Chord. Once the user joins the overlay successfully, it gets a PID using formula (1) for the node and a UID using formula (2) for its name as discussed in section 3.2. Now the peer has a finger table including some <*PID, IP:Port*> pairs called finger table entry. The user then sends a sip REGISTER message to another peer whose PID is just larger than the user's UID. The REGISTER message contains the user's contact information, mainly a <*username, IP:port*> pair.

User location: We assume there are two users, Alice and Bob, who have joined the VCNP overlay, and now Alice wants to invite Bob to start a private video session. The first task Alice has to do is to find Bob and sends him a SIP INVITE message. Assume that VCNP peers act as stateless sip proxies. As shown in Figure 2, for example, Alice's PID is 8; Bob's PID is 47 and his UID is 20. Alice knows Bob's username, and gets his UID using formula 1). She then sends an INVITE message to peer 21 whom she considers to be responsible for storing Bob's contact. Peer 21 has Bob's contact, and transfers the INVITE message to Bob on peer 47. The INVETE message contains Alice's contact and Bob's answers to Alice with an ACK message. After this is completed, a session can be started.

User Authentication: VCNP has a global *Login Server*, which generates its own public/private key pair, (S_S, V_S) using RSA algorithm. V_S is distributed to all VCNP clients at start-up time so that every VCNP client can verify it is really talking to the Login Server. Each client generates its own key pair, e.g., Alice with (S_A, V_A) and Bob with (S_B, V_B). When Alice and Bob login to Login Server for the first time, Login Server generates an IC_A by signing Alice's username and V_A using its own signing key S_A. Bob also gets his IC_B. Alice and Bob can now trust each other using a challenge response mechanism.

Register the Group to VCNP: All members in a video conference session are called a Video Conference Group (VCG), and the member who starts the session is called the VCG owner. VCG owner is responsible for registering the VCG to VCNP overlay. Each VCG has a unique *groupname*. Group registration is more or less the same as user registration, as long as VCNP makes sure that *username*s and *groupname*s are unique. The only difference is that group registration information is stored in MCUs only, and the information will be updated as long as there are users join or leave the group.

User Invitation: Given that Alice registers and sets up a VCG, it then sends an INVITE message to Bob. This

message takes a SDP [10] description that tells Bob the session type is a VPN session. Alice selects a 256-bit AES key, G_K, as the session key. Alice signs to this key with S_A, and sends it to Bob together with IC_A after encrypted by Bob's V_B, that is $E_{VB}(E_{SA}(G_K), IC_A)$. Bob receives the message and decrypts it to get G_K. Bob gets the other VPN parameters including its virtual IP address and connects to Alice to setup a VPN session.

User Leaving the Session: When a user, say, Bob, leaves a VCG, it must send a BYE message to the VCG owner, *i.e.*, Alice. Alice then updates the group contact stored in the responsible MCU and broadcast the new group contact to VCG members.

4. PEER-ASSISTED CDN DATA PLANE

4.1 Data Plane Architecture and System Components

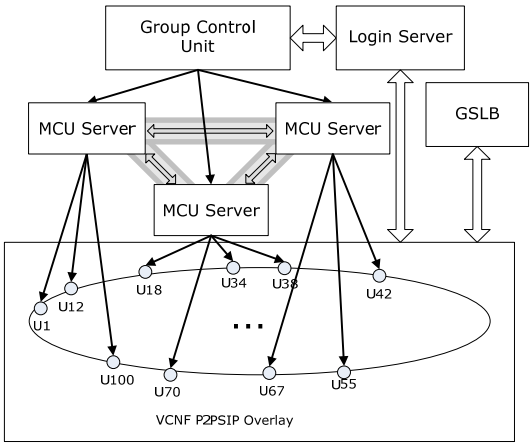

Figure 3 Data Plane Architecture

As shown in Figure 3, the system architecture to be designed is based on peer-assisted CDN for data plane. Our network-centric CDN part of the system is composed of fiber-linked router nodes and edge servers. It is a "highway" for data delivery, responsible for efficient data transmission through ISPs and states/provinces in the country. These nodes are routers with special configuration that can identify our video conferencing data and forward data accordingly. These fiber-linked router nodes in content transmission layer (as shown in Figure 1) usually locate in different ISP domains, and are interconnected by fiber lines, providing high quality physical paths for data delivery. Channels between every two router nodes are QoS-guaranteed. Packet delay can be limited to no more than 40ms between any pair of fiber-linked router nodes; jitter is limited to less than 1ms; packet loss is almost zero. We adopt a dynamic bandwidth allocation algorithm on router nodes to handle the possible congestion scenario during burst.

The DiffServ QoS is mainly achieved by the routers in the fiber network. Since the routers transfer both the video conference data and other FTP or HTTP data, the video conference data has to compete for the bandwidth of fiber networks in the backbone network. To ensure the QoS, the routers have to provide differentiated service for our video conference data by setting the head TOS label in IPv4. Theoretically, either peer or MCU can label the data. But to ensure the security, our MCU will take the role of labeling the data.

A Multipoint Control Units (MCUs) here is a server equipped with a Linux operating system, and runs some of our proxy servers programs. MCUs are responsible for data transfer. MCUs determine how to route the media data and when to provide mechanisms for multicast to save backbone network bandwidth. A MCU connects to a fiber-linked backbone node directly or within limited hops. It connects to backbone node using some tunnel protocol to make sure that the link between MCU and the fiber-linked backbone node is QoS-guaranteed.

The primary tasks of MCUs in traditional multipoint videoconferencing systems are to mix the incoming audio and video signals, so that single streams of audio and video are transmitted to all participants. However, for the purpose of scalability and MCU resources saving, these works are done by end users instead in our system. The client software is responsible for media coding, mixing as well as synchronizing. Since MCU is also one peer in the conference group and meanwhile it can be accessed easily, it serves as an ideal in-between node for penetrating peers behind NAT (Network Address Translation)[29].

Another element is the Group Control Unit (GCU). As the system aims to serve ad hoc environments, where even millions of users may be on line at the same time, an efficient user and group management scheme must be in place. GCU is responsible for group management, including group creation and delete, session key distribution and updating.

The Global Service Load Balance (GSLB) subsystem in Figure 3 is used to locate users and help users to find the "nearest" MCU, the *father MCU*. The GSLB solution here is to improve the availability of the system and to help optimizing the structure of the whole system. GSLB solutions are often base on DNS resolution. As GSLB is not the key feature of this paper, details about GSLB is not presented here.

In addition, end users also monitor the connections conditions and make decisions about application level routing policy choosing. When an end user connects to a nearby MCU and congestion occurs on the link, user must degrade its data transmitting rate and make decisions about what data types should be kept and what can be dropped.

4.2 Multipoint video conference procedures

The main procedure of a multipoint video conference is described in Table 2:

Table 2 Video Conference Procedures

1) *Peer-assisted CDN network is setup.* Fiber-linked backbone routers together forms content transmission layer of the network, and MCUs compose the content distribution layer. MCUs then measure the network parameters. Delay and jitter values of every two MCUs are measured and stored in a database on GCU.
2) Users login and register to VCNP.
3) Presence function of signaling protocols (here P2PSIP) informs users the status of their friends.
4) VCG owner creates a VCG.
5) VCG owner invites appropriate users to join the VCG and authenticates the invited users.
6) Network conditions are probed and the source decides whether the internet paths to the destinations meet QoS requirements. If not, it will need MCUs' assistance. The nearby MCU should make a similar decision.
7) P2PSIP helps negotiating the session parameters like used codec, frame rate and the data receiving protocol and port(s).
8) The media session is setup. Network conditions are monitored during the session and data paths may be changed accordingly.

5. IMPLEMENTATION

5.1. System implementation framework

Our system implementation framework is shown in Figure 4. The system includes one global login server for user administration; certain bootstrap server(s) to help peers join the VCNP overlay; hundreds of MCU servers to help video groups delivering video content; and each peer contains a P2PSIP module which all together constructs a distributed database to maintain and manage conference users/ groups.

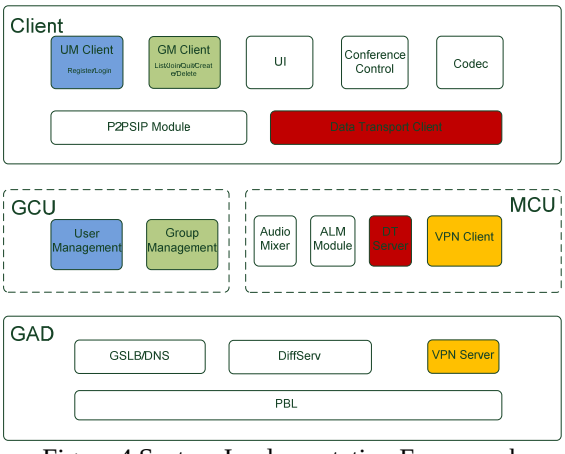

Figure 4 System Implementation Framework

Conference group can apply to a MCU server to help deliver media content. An MCU server receives video streams from an active user and sends them to all other members in the same group. Suppose that the video stream bitrates is 128Kbps, then a MCU with a gigabyte network connection can support about 600 users simultaneously.

For the client software architecture, as shown by Figure 5, it contains components including P2P protocol implementation, a SIP protocol stack, VPN server and client, a video codec and a user interface.

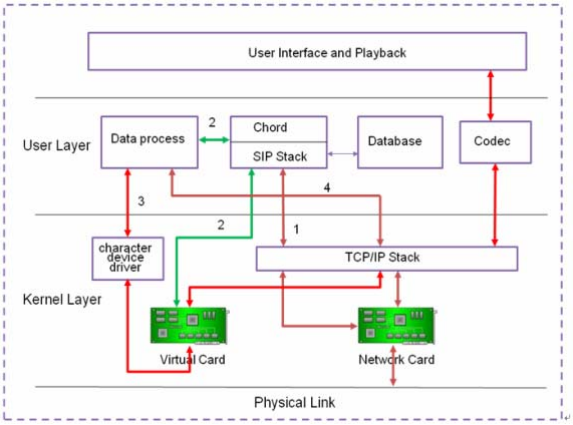

Figure 5 VCNP Client Framework

We use the openssl [11] library to setup VPN tunnels between users and MCUs. The client work flow is as follows:

i) First, after the *user* logins to the *login server*, Chord starts a process of joining the overlay (step 1 in Figure 5).
ii) Then the user selects one of its friends to start a video session, SIP stack gets the friends name and starts a User Location process. The user invites its friends to start a VPN session first, and they together negotiate for a session key and their virtual IP addresses.
iii) Data process component gets the session key and the virtual IP, as the green line shows in Figure 5 step 2.
iv) The user then starts a video camera to capture videos, and starts to receive media data to playback. The media data are first sent to the virtual card. VPN data process component gets the data from character device driver, encrypts the data with the session key and sends the data to the real network card. A converse process takes place when data are received.

5.2. Source coding

We use xvid [12] codec library to encode and decode captured video frames. Xvid is a MPEG-4 video codec library with high compression ratio and fast compression speed. VCNP can be configured to support different video definitions and frame rates. An active user has a video window with a definition of 320*240, and a 15 fps frame

rate. The inactive users' video window is 176*144 with a 5 fps frame rate. As our experimental result shows, the active user video stream bit rate is about 70Kbps after xvid decoding. And an inactive video stream is about 30Kbps. The bit rates are mean values and differ a lot with static or dynamic background. In addition, our system VCNP adopts G.729 as its audio encoding method, with a constant bit rate, 8Kbps.

Figure 6 shows a user interface demo of our VCNP system with an active user and up to 16 inactive users in a conference group.

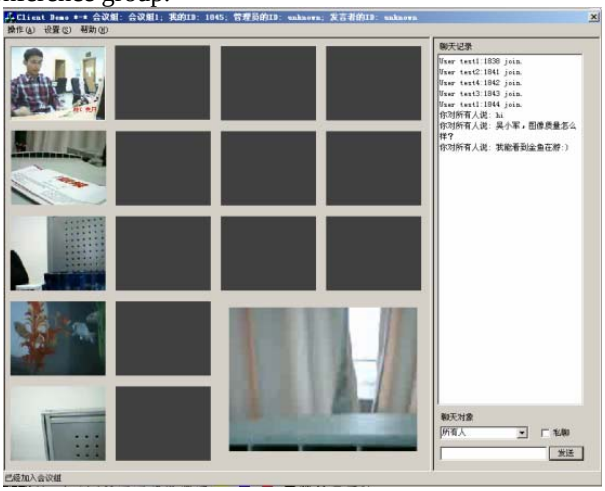

Figure 6 User interface demo of VCNP

5.3 Authentication and encryption

To ensure the expected security in video conferencing environment, we adopt a set security mechanisms for our system. We use RSA [13] algorithm to generate user public/private key pairs. The asymmetrical encryption scheme is used for user authentication and session key negotiation. For session cryptography, use AES-256, which provides sufficient security performance and a fairly good encryption speed. Note that when a user joins or leaves a session, MCU is responsible for updating the session key and broadcasting it to all valid members.

5.4 Data Delivery Schemes

In our system, MCU maintains a Global Permission Table (GPT) and each member maintains a local permission table (LPT). Members can control whose streams it wants to receive. If a member denies receiving another one's video or audio streams, the request will sent to MCU and transferred to the target member. The target member's LPT is updated and it will never send the forbidden stream data to the request user. MCU can control every link between any members, acting as a chairman of the conference. LPT is useful when a user's network can't afford the received bit rate. For example, it can easily disable all the inactive video windows and only receive audio and active video window data.

A video conference session may have several data streams. In VCNP, there are four types: active user video stream, inactive user video streams, audio streams and text streams. VCNP delivers these four types by different ways: active video streams with high definition and high frame rates as well as audio streams are delivered first to MCU, and MCU transfer these streams to all group members; inactive video streams and text data are sent to all group members directly in a mesh way.

The reasons for our system to adopt this scheme are that: 1) active user video streams consume too much network bandwidth if delivered to many other members; 2) although audio streams are not bandwidth consuming, audios from different members are required to be mixed. 3) inactive video streams with low definition and low frame rate are selectable. In some scenarios, they are not necessary. 4) text messages need little bandwidth, and in addition, sometimes members have the requirement to talk to each other privately. This is why text messages are sent in a mesh manner to all group members.

6. PERFORMANCE EVALUATION

In video or voice communications, traditional performance metrics include session setup time and media delay time. Besides the evaluation on the time metric, we also make security performance as an additional metric in this part.

6.1 Session setup time

As presented in sections above, we use Chord for the user lookup algorithm. In Chord overlay, for any user contact information, the node whose range contains the user is reachable from any node in no more than log_2N overlay hops, where N is the size of Chord namespace. Assuming that mean network delay time between overlay hops is *TTL*, then it takes log_2N*TTL for the first INVITE message to reach the target peer. Session key negotiation involves a 3-way handshake process. Thus the total time taken to setup a session is about:

$$T = log_2N*TTL + 3*TTL \qquad (4)$$

For a numerical example, suppose N = 1,000,000 and TTL < 100ms, then T < 2.3s. As is shown in fig.7, our experiment result verified the conclusion, e.g. the session setup time seldom extends 3s.

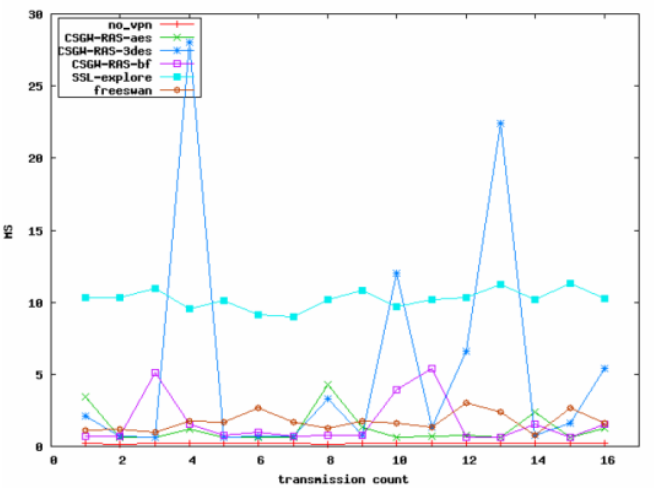

Figure 7 Session setup time

6.2 Video/audio Stream Delay

In our system, active video stream and audio stream are delivered via MCU, and inactive video data are delivered directly between group members. Audio data also need to be mixed in MCU and thus audio streams have the largest delay. Assume that data encoding or decoding time is Co_T, encryption or decryption time is Cr_T, mean network delay is *TTL*, then the total audio stream delay can be expressed as:

$$D = 2* Co_T + 2* Cr_T + 2* TTL = 2*(Co_T + Cr_T + TTL) \quad (5)$$

Experiment result shows that $Co_T <= 40ms$, $Cr_T < 10ms$, and $TTL < 100ms$, so D < 300ms. According to VoIP standard, it is an acceptable delay time.

6.3 Security and its overheads

As shown in our previous work [14], SSL VPN has almost equal safety with IPSec whose high security performance is well-known all over the network research area.

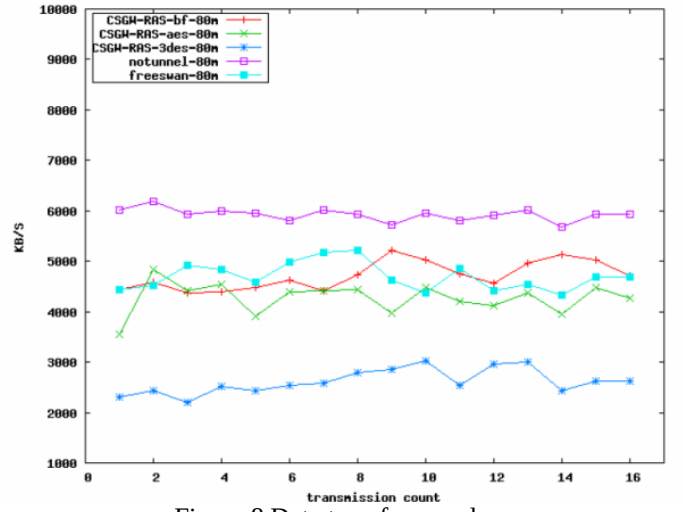

Figure 8 Data transfer speed

In addition, we examined on the possible negative impact of security management over the QoS of the service. Figures 7-8 demonstrate that using VPN to implement security schemes in a video conferencing system takes a negligible session start time and about 20% data transfer delay.

6.4 Video Quality

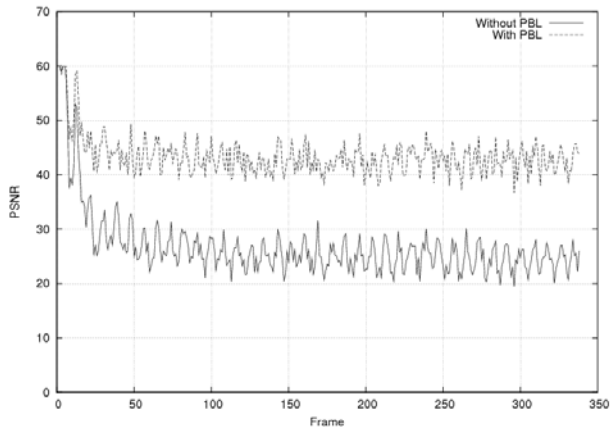

Figure 9 PSNR value on Different Links

Figure 9 illustrates the PSNR values on different links, from which we can see that, comparing with direct and uncontrolled Internet connections between peers, peer-assisted CDN remarkably improves the PSNR values: from about 30 to above 40. According to the PSNR to MOS conversion formula [26], VCNP improves the video quality from fair or good to excellent.

7. CONCLUSIONS

There is an ever-increasing demand for multimedia conferencing over the Internet, but efficient and secure deployment of video conferencing systems in large scales remains an elusive goal. Challenges mainly lie in three aspects: 1) security, 2) scalability, and 3) QoS.

This paper has presented a scalable and secure video conferencing system called VCNP which is a comprehensive solution including not only data delivery mechanism but also user/group management functions. For the control plane, we have organized users into a chord-based P2P overlay network and introduced a VPN session over P2PSIP to act as a security infrastructure for our conferencing system. As for the data plane, we have designed a system architecture called Peer-assisted CDN, which combines the network-centric and overlay CDN model, and structured P2P model to enhance the QoS as well as scalability and flexibility of the system. Both our analysis and experiment results have reveals that VCNP is an efficient and secure Internet video conferencing system.

8. ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (No. 60673184 and No. 60873254), the National High Technology Research and Development Program of China (No. 2007AA01Z419), the National Basic Research Program of China (No. 2008CB317101), and Tsinghua – ChinaCache CDN Program.

9. REFERENCES

[1] CHU Y H, RAO S G, SESHAN S, ZHANG H. Enabling conferencing applications on the Internet using an overlay multicast architecture[J]. ACM SIGCOMM Computer Communication Review, 2001, 31(4):55-67.
[2] P2PSIP status pages. http://tools.ietf.org/wg/p2psip
[3] J. Rosenberg and H. Schulzrinne, G. Camarillo, A.R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.
[4] S. Deering. Multicast Routing in Internetworks and Extended Lans. In Proceedings of ACM SIGCOMM, August 1988.
[5] CHU Y H, RAO S G, SESHAN S, ZHANG H. A case for end system multicast[J]. ACM SIGMETRICS Performance Evaluation Review, 2000, 28(1):1-12.
[6] PENDAKARIS D, SHI S. ALMI: an application level multicast infrastructure[A]. Anderson T. The 3rd USENIX Symposium on Internet Technologies and Systems[C]. San Francisco, CA, USA: USENIX Association, 2001. 49-60.
[7] OpenVPN homepage. http://openvpn.net/
[8] Deploying Secure Enterprise Wide IP Videoconferencing Across Virtual Private Networks. http://www.h323forum.org/papers/polycom/DeployingSecureIPVideoNetworks.pdf
[9] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrish-nan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, San Diego, CA, USA, Aug 2001.
[10] M. Handley and V. Jacobson. SDP: Session Description protocol. RFC 2327, Internet Engineering Task Force, April 1998.
[11] http://www.openssl.org/
[12] xvid homepage. http://www.xvid.org/
[13] B. Li, and H. Yin, "The Peer-to-Peer Live Video Streaming in the Internet: Issues, Existing Approaches and Challenges," IEEE Communications Magazine, Vol. 45(6), 94-99, June. 2007.
[14] Yada Hu, Hao Yin, Chuang Lin, Xin jiang, Ying Ouyang, Chao Li. CSGW-RAS：A Novel Secure Solution for Remote Access Bases on SSL. ISPACS 2007.
[15] Polycom homepage. http://www.polycom.com
[16] Intserv Charter. http://www.ietf.org/html.charters/OLD/intserv-charter.html
[17] DiffServ Charter. http://www.ietf.org/html.charters/OLD/diffserv-charter.html
[18] WebEx homepage. http://www.webex.com
[19] Al-Mukaddim Khan Pathan and Rajkumar Buyya. A Taxonomy and Survey of Content Delivery Networks, Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, Feb. 12, 2007.
[20] F. Douglis, and M. F. Kaashoek. Scalable Internet Services. IEEE Internet Computing, Vol. 5, No. 4, 2001, pp. 36-37.
[21] I. Lazar, and W. Terrill. Exploring Content Delivery Networking. IT Professional, Vol. 3, No. 4, pp. 47-49, 2001.
[22] Akamai homepage. http://www.akamai.com
[23] AppStream homepage. http:// www.appstream.com
[24] Limelight Networks. http://www.limelightnetworks.com
[25] ChinaCache homepage. http://www.chinacache.com/
[26] Jirka Klaue, Berthold Rathke, and Adam Wolisz. Evalvid- A Framework for Video Transmission and Quality Evaluation. Modelling Techniques and Tools for Computer Performance Evaluation, 2003.
[27] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol L2TP", RFC 2661, August 1999.
[28] B. Patel, B. Aboba, W. Dixon, G. Zorn, and S. Booth, " Securing L2TP using IPsec", RFC 3193, November 2001.
[29] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.
[30] Keqiu Li, Hong Shen, Francis Y. L. Chin, and Weishi Zhang. Multimedia Object Placement for Transparent Data Replication. IEEE Trans. on Parallel and Distributed Systems, Vol. 18, No. 2, pp. 212-224, February 2007.
[31] Yan Huang, Tom Z. J. Fu, Dah-Ming Chiu, etl. "Challenges, Design and Analysis of a Large-scale P2P-VoD System", ACM Sigcomm 2008.
[32] Susu Xie, Bo Li, Gabriel Y. Keung, and Xinyan Zhang, "Coolstreaming: Design, Theory and Practice", in IEEE Transactions on Multimedia, 9(8): 1661-1671, December 2007.
[33] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher, "Availability in BitTorrent Systems," in Proc. Infocom, 2007.
[34] J. Xu, B. Li, and D.L. Li, "Placement Problems for Transparent Data Replication Proxy Services," IEEE J. Selected Areas in Comm., vol. 20, no. 7, pp. 1383-1398, 2002.