

Hi Application Software

The field of the invention is peer to peer systems and in particular systems that use VVoIP (Video and Voice over Internet Protocol).

BACKGROUND

Various applications allow users with Internet, Wi-Fi, and Bluetooth access to talk to each other via video, voice and text messaging. Many systems also provide one or more social networking options including e-gifts, games, online shopping, radio, and video/TV elements, selling of music and music videos and many more.

Some of the applications use Peer to Peer (P2P) technology to users including connecting users (sometimes referred to as peers/nodes which are the computers being used by those users) via largely ad hoc connections. Such networks are useful for many purposes. Sharing content files containing audio, video, data or anything in digital format is very common, and real time data, such as telephony traffic, is also passed using P2P technology.

A pure P2P network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client-server model where communication is usually to and from a central server. A typical example of a file transfer that is not P2P is an FTP server where the client and server programs are quite distinct: the clients initiate the download/uploads, and the servers react to and satisfy these requests.

In 'pure' P2P networks, peers act as equals, merging the roles of clients and server. In such networks, there is no central server managing the network, neither is there a central router. Some examples of pure P2P application layer networks designed for file sharing are Gnutella (pre v0.4) and Freenet. P2P networks have a distributed network architecture wherein the hardware of each user allows a portion of their resources (including disk storage and network bandwidth) available to others in the network, without the need for central coordination. A peer is both a supplier and consumer of the collective resources.

There also exist countless hybrid P2P systems, which distribute their clients into two groups: client nodes and overlay nodes. Typically, each client is able to act according to the momentary need of the network and can become part of the respective overlay used to coordinate the P2P structure. This division between normal and 'better' nodes is done in order to address the scaling problems on early pure P2P networks. Examples for such networks are for example Gnutella (after v0.4) or G2.

Another type of hybrid P2P network includes networks using on the one hand central server(s) or bootstrapping mechanisms, on the other hand P2P for their data transfers. These networks are in general called 'centralized networks' because of their lack of ability to work without their central server(s). An example for such a network is the eDonkey network (eD2k).

The P2P overlay network consists of all the participating peers as network nodes. There are links between any two nodes that know each other: i.e. if a participating peer knows the location of another peer in the P2P network, then there is a directed connection from the former node to the latter in the overlay network. Based on how the nodes in the overlay network are linked to each other, the P2P networks are unstructured or structured.

An unstructured P2P network is formed when the overlay links are established arbitrarily. Such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another node and then form its own links over time. In an unstructured P2P network, if a peer wants to find a desired piece of data in the network, the query has to be flooded through the network to find as many peers as possible that share the data. The main disadvantage with such networks is that the queries may not always be resolved. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful. Since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding will find a peer that has the desired data. Flooding also causes a high amount of signalling traffic in the network and hence such networks typically have very poor search efficiency. Most of the popular P2P networks are unstructured.

Structured P2P network employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired file, even if the file is extremely rare. Such a guarantee necessitates a more structured pattern of overlay links. By far the most common type of structured P2P network is the distributed hash table (DHT), in which a variant of consistent is used to assign ownership of each file to a particular peer, in a way analogous to a traditional hash table's assignment of each key to a particular array slot.

A system like Skype TM is a peer-to-peer network with three main entities: supernodes, ordinary nodes and the login server. It is an overlay network: each client builds and refreshes a list of reachable nodes known as the host cache. The host cache contains IP address and port numbers of supernodes. Signalling is encrypted using RC4; Voice data is encrypted with AES.

Supernodes relay communications to other clients behind a firewall. Any Skype client can become a supernode if it has good bandwidth, no firewall, and adequate processing power. Supernodes are grouped into slots (9-10 supernodes). Slots are grouped into blocks (8 slots).

Skype also routes calls through other Skype peers on the network to ease the crossing of Symmetric NATs and firewalls. This, however, puts an extra burden on those users who connect to the Internet without NAT, as their computers and network bandwidth may be used to route the calls of other users.

The Skype client's application programming interface (API) opens the network to software developers. The Skype API allows other programs to use the Skype network to get "white pages" information, and manage calls. The Skype code is closed source, and the protocol is not standard.

There are limitations to the functionality provided by the type of P2P networks described above. One of them is the inability to provide conference video calls. The invention described and defined in this specification provides such functionality or at least provides an alternative.

BRIEF DESCRIPTION OF THE INVENTION

In a broad aspect of the invention a peer to peer method for creating a multimedia connection, each peer having video and audio sources, and the connection between at least three peers, the method includes the steps:

processing both audio and video sources of each peer provided in digital form with a packetizer; and streaming those sources to multiple outputs to other of the peers using a Peer to Peer protocol.

In an aspect of the invention a method of facilitating data communications between peers in a peer to peer network environment having video and audio sources associated with each peer, and the data communication being between at least three peers, the method includes the steps of:

processing with a packetizer both audio and video sources of each peer provided in data form; and streaming those sources after packetization to multiple outputs to other of the peers using a Peer to Peer protocol.

In an aspect of the invention a system that facilitates data communications between peers in a peer to peer network environment having video and audio sources associated with each peer, and the data communication being between at least three peers includes:

multiple computer hardware executing peer to peer software including a communications component having a packetizer for processing both audio and video sources of each peer provided in data form; and streaming those sources after packetization to multiple outputs to other of the peers using a Peer to Peer protocol.

A detailed description of one or more preferred embodiments of the invention is provided below along with accompanying figures that illustrate by way of example the principles of the invention. While the invention is described in connection with such embodiments, it should be understood that the invention is not limited to any

embodiment. On the contrary, the scope of the invention is limited only by the appended claims and the invention encompasses numerous alternatives, modifications, and equivalents. For the purpose of example, numerous specific details are set forth in the following description in order to provide a thorough understanding of the present invention. The present invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the present invention is not unnecessarily obscured.

Throughout this specification and the claims that follow unless the context requires otherwise, the words 'comprise' and 'include' and variations such as 'comprising' and 'including' will be understood to imply the inclusion of a stated integer or group of integers but not the exclusion of any other integer or group of integers.

The reference to any prior art in this specification is not, and should not be taken as, an acknowledgment or any form of suggestion that such prior art forms part of the common general knowledge.

BRIEF DESCRIPTION OF THE FIGURES

Figure 1 depicts the creation of three key values which are created by applying a hash function to the data;

Figure 2 depicts the establishment of a secure communication exchange;

Figure 3 depicts an embodiment of the architecture of the subject system;

Figure 4 depicts the establishment of a call when direct UDP communication is possible;

Figure 5 depicts a call establishment when UDP is not firewalled and a direct link between users cannot be established;

Figure 6 discloses the hardware configuration of a three way call scheme;

Figure 7 depicts a relay multicasting arrangement according to the subject system;

Figure 8 depicts a more detailed illustration of the workings of elements of Figure 6;

Figure 9 depicts the process of creating a three way audio/video interaction between three participants is illustrated; and

Figure 10 depicts a pictorial representation of a P2P mesh for video and audio distribution in multi-way conference call between peers.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Unlike traditional client-server models, which most of modern Instant Messaging (IM) software follow, the arrangement described herein uses a Modified P2P model.

All nodes in the Modified P2P Network are divided into three main groups

- Ordinary nodes (these are ordinary users, typically connected using ADSL lines, located behind the Network Address Translation (NAT) boxes;

- Supernodes (these nodes have public IP address, plenty of available bandwidth, a lot of processing power (CPU, Memory)). To join the network a user connects to any of the online super-nodes in the network;

- Special purpose services (login servers, location servers, advertising servers, radio/TV content distributors, etc.)

Information lookup in P2P network is achieved using all the supernodes, the subject network uses a structure, called a Distributed Hash Table (DHT). Every supernode participating in subject network DHT is assigned a 160-bit hash value which is calculated from node IP:port pair.

- 5 When a peer wants to store data in the DHT the data hash of the data is calculated and data is stored on $N > 1$ nodes. Figure 1 discloses the creation of three key values (10, 12, and 14) which are created by applying a hash function 16 to the data (18, 20 and 22). The key values are stored in a respective (all) peers.

In the subject system the DHT is used for the following operations:

- 1) User search
- 2) User super-node discovery
- 10 3) Content provider discovery

Login

- Yet further a Login server is used for registering new users and authenticating existing ones. After a user is authenticated within the subject systems' login server they are given a certificate which is used later to establish a secure communication link. All traffic (voice, video, messages) are encrypted using the encryption key in the certificate (a public/private paired key is used) where the public key can be distributed freely with a certificate.
- 15

Security phases:

- 1) User registration
 - Register a username which is obtained by interacting with the login server
- 1) User login
 - 20 - Get the one time public key for the user certified by login server
- 1) User to User authentication
- 2) User to User communication

Encrypted P2P Communication

- 1) After mutual authentication, Alice and Bob establish a 256-bits common session key K_s (AES) for encryption. Alice and Bob are names used in the security field to describe two unrelated users;
- 25 2) Each user contributes 128-bits for the 256-bits long K_s ;
- 3) Each user sends its contribution to the other side, encrypted with the latter's public key;
- 4) Two 128-bits contributions are combined to generate the 256-bits secret session key K_s ;
- 5) All traffic (voice, video and text) is encrypted.
- 30 The process described above is illustrated in Figure 2 which illustrates the establishment of a secure communication exchange. The formulae disclosed in the figure are typical for public/private encryption and decryption arrangements.

Location server

- In the subject system it is possible to control the connection behaviour between users which need to use a server.
- 35 In general the closer the server is located to the clients the smaller the connection latency, which is important for

VoIP system. Also network packets exchanged between users stay within the same ISP network which lowers ISP expense. As is stated above, users should connect to a supernode in order to join the subject network. When an application associated with the subject network is first installed on a users' hardware it not does not know anything about supernode locations. This is the time when user is connected by the application to a location server in order to obtain one or more supernode addresses preferably located geographically close to the user. If there are no supernodes in the users' network of geographic region then a location server can be contacted after some time (several days) passed to check whether new supernodes appeared in the user region. Supernodes notify their presence to the system by contacting a location server every few hours. This helps the location server to select closer supernodes which are most likely to be online. Location servers can exchange information to update their records of the location of supernodes.

Each supernode knows about neighbouring users and common nodes, so it is not required for ordinary nodes to connect to a location server often, because when a user connects to supernode, the supernode sends some of it's online neighbours to newly a connected user. So even if all location servers are unavailable for some period of time most of the users will still be able to log-in and communicate. The architecture of the subject system is depicted in Figure 3. Ordinary hosts are depicted as 30, supernodes as 32 and neighbour relationships 34 between peers within the subject network. The location server 36 and login server 38 are accessed as needed by one or more user/host devices.

The media transfer and codecs the subject system uses include the ability to provide a (simultaneous both way) audio/video conference having a minimum number of participant users equal to 3.

One of the subject system features is that it can easily cross NATs using a supernode with a public IP address as media proxy (relay) when it is not possible to connect users directly. About 80% of home users have "P2P friendly" routers which allow direct communication thus lowering bandwidth utilization of relay supernodes.

Call establishment and NAT traversal is also provided by the subject system. The following steps are performed for three different cases a), b) and c).

a) Connection between two users where direct communication could be established

(Users have router ports opened, full cone NATs or restricted cone NATs where UDP hole punching can be done). Figure 4 depicts the establishment of a call when direct UDP communication is possible.

(1) User A finds their buddy (user B) location in the subject modified P2P network;

(2) User A establishes TCP connection to user B supernode;

(3) User A sends INVITE message to user B over TCP link;

(4) User B replies with acknowledgement (ACK) over TCP and sends several probe packets to user B IP:port;

(5) User A sends several UDP packets to user B IP:port in order to build direct UDP link;

(6) User A receives ACK from user B over UDP; and

(7) A direct UDP communication is established between the users and the user can now send and receive media data.

b) Connection between 2 users where UDP is not firewalled but direct UDP link cannot be established (symmetric NAT, when the router does not support UDP hole punching)

Figure 5 depicts a call establishment when UDP is not firewalled and a direct link between users cannot be established.

1) - 5) as in case (a) above

(6) User A does not receive ACK from user B

(7) User A sends FIND_RELAY request to user B to tell user B to find some relay super nodes

(8) User A selects some relay super nodes which are close to user A

5 (9) User B receives a list of super nodes found by user B

(10) User A merges both supernodes lists (selecting the most appropriate ones) and creates the relay supernode list for this call

(11) User A sends the merged list to user B

10 (12) User A and user B select the supernodes which will act as media proxy for user A and user B correspondingly

(13) Call is established.

c) Connection between 2 users when UDP is firewalled and direct communication is impossible

15 All call establishment steps are identical to case (b) with the only difference being that one or both users use TCP to send media data. In that case communication is limited to audio-only (no video data is sent over a TCP link)

20 Three-way audio/video calling is a feature of the subject system and specific arrangements are made to support a 3-way audio and video call for users who have UDP protocol enabled (not firewalled). Currently full custom mesh topology is being used to transfer both video and audio, which means that audio/video stream from each peer (user), is being duplicated. Due to low bandwidth requirements for the subject system it is possible for both audio and video 3-way call to be established between most peers who have a broadband internet connection. Typically video conferencing involves low relative motion of the users included in the conference so high picture quality on low bit rate connections. The nearest future plans include implementation of automatic video encoder bit rate to bandwidth tuning which will allow the system to automatically raise video quality if the user has sufficient available bandwidth.

25 Figure 6 discloses the hardware configuration of a three way call scheme showing peer 1 60 generating both video 62a from webcam/camera 62 and audio 66a from microphone 66, which is digitized respective video encoder 64 and audio encoder 68 and transmitted via packetizer 70 in two audio/video (AV) streams 72 and 74. Peers 2 and 3 receives respective A/V streams 72 and 74 which are divided into audio and video streams by a splitter 76 and forwarded to respective audio 78 and video 80 decoders. The output of the decoders are further processed, wherein the decoded audio is processed by a jitter buffer 82 which transforms the audio data into an analogue audio signal for input to a speaker 86, and the decoded video is processed by video module 84 transforming the video data into data suitable for display on a monitor 88.

Relay multicasting

35 If peer behind symmetric NAT places a call a relay node is being used to route traffic to other call participants. If user is in 3-way call traffic is not duplicated on his/her side. Instead it is being delivered to relay node as a single stream and relay node does traffic duplication. Such technique saves bandwidth on peer as well as the relay side. The subject system, in a preferred embodiment uses Speex and Intel IPP wideband audio codecs for audio compression/decompression and Intel IPP H.264 (high profile) for video. Audio bandwidth usage is about 40 3 KB/sec. Video bandwidth usage depends on the webcam being used and even CPU load (when CPU load is too high application automatically lowers the frame rate, so the bandwidth usage also becomes lower). It

actually ranges from 4KB/sec to 20 KB/sec. It is planned also that application will detect when users have plenty of available bandwidth and raises video quality/bandwidth usage in such case

Figure 7 depicts a relay multicasting arrangement according to the subject system which supports multi-way audio and video calls for all users who have UDP protocol enabled (not firewalled). Full mesh topology is being used to transfer video, which means that video stream from each peer, is being duplicated (except the case when relay multicasting is used).

Fig. 8 depicts a more detailed illustration of the workings of elements of Figure 6. The video encoder 64 (Figure 6) is shown as an Intel IPPh264 encoder and a portion of the software functionality of the packetizer 70 is shown as including a Microsoft Direct X multimedia API for Microsoft Windows wherein DirectShow is a component of DirectX used for streaming media, for creating a multiple virtual video output (two or more).

Further depicted in Figure 8 is the Audio encoder 68 (Figure 6) as a Speex Audio encoder and a portion of the software functionality of the packetizer 70 is shown as including an Intel IPP echo cancellation function. Yet further included in the packetizer 70 is a multiplexer function to merge audio/video streams for synchronisation and packetization 70a. The output of this function is provided to a Peer to Peer transport protocol 70b.

Referring to Figure 9, the process of creating a three way audio/video interaction between three participants is illustrated.

Peer 1 is the initiator in the sequence shown.

Peer 1, using a web cam 62 (Figure 6) acts a fully utilized capture source and calls Peer 2.

Upon capture, Peer1's web cam 62 will send the signals via a video encoder 64 to the packetizer 70 which creates a multiple virtual video source output from the video from the web cam 63.

The video stream from Peer 1 is sent to Peer 2 (Figure 6).

When Peer1 wants to establish a second video call, Peer 1 repeats the previously described steps, to output a further of the multiple virtual video source outputs.

Depending on the bandwidth available the number of simultaneous P2P audio/video links is not otherwise limited. Although the processing power available may be limiting.

Figure 10 depicts a basic P2P Mesh configuration, wherein a full mesh is when video data from each peer is being duplicated:

Consider users A, B, and C

A sends to B and C

C sends to B and A

B sends to C and A

A Packet (P) is sent to each other peer, in this example B and C.

A (10.0.0.61:9099) sends (P) to 10.0.0.62:1065 (B)

A (10.0.0.61:9099) sends (P) to 10.0.0.64:20653 (C)

Users A actually duplicate the video data and send to the connected users.

Another issue is the mixing of audio, in particular efficient mixing of audio which can conserve CPU effort and may reduce bandwidth requirements.

Nodes A, B, and C where node B is a mixer node (which has processing power that can be used)

Step 1) A sends audio packet to B

Step 2) C sends audio packet to B

Step 3) B:

- 5 a) decodes audio packet from A
- b) puts audio packet from A to it's jitter buffer (scheduled playback buffer)
- c) mixes decoded audio packet from A with it's own audio packet captured from its own microphone.

Mixing in PCM (Audio data) is just summing of the audio sample values

- d) encodes mixed packet and sends it to C
- 10 (e)-(g) the same is done with packet received from node (C)
- h) mixed packet is encoded and sent to A

the above arrangement of method steps can be implemented in software, which forms part of the application downloaded and used by each peer/node.

- 1) nodes A and C have less bandwidth requirements (actually the same as in 1-1 talk)
- 15 2) in case of using chargeable communication services like GPRS/3G, peers/nodes A and C also save money
- 3) B has to separately encode 2 audio streams instead of 1 in a full-mesh topology as described herein. This requires B to have sufficient processing power
- 4) If node (B) leaves the conference then the conference no longer exists or it is possible to reconfigure the topology to compensate which may use one or more other of the peers/nodes that have the capacity to perform the functionality required.
- 20

As mentioned above the arrangements, methods and configurations can be used for multi-way peer to peer connections, so that more than 3 peers can be involved.

Media Data Streaming sequence

- 25 The video/audio data flow is shown in Figure 6, previously described, and referred to which includes processing both audio and video sources of each peer provided in digital form with a packetizer; and streaming those sources to multiple outputs to other of the peers using a Peer to Peer protocol.

The sequence of events is the following:

video source -> encoder -> call context -> command router -> network façade service

- 30 **Call context** defines how to deliver the multimedia data to one or more connected peers. It creates commands which contains the multimedia data and routing information.

Packetizer as shown in Figure 6 has many functions some of which have been described previously, and repeated and added to below to include:

- 1) Parse command routing table (created by call context) and send command to appropriate destination

2) Split large datagrams (a datagram is a unit of data and associated header data which is the basis of connectionless networking) into smaller ones

5 **Command routing** information may contain multiple directions to which media data should be sent. The Command router is used to route commands to appropriate destinations. This component parses routing information in the command header of the datagrams and the call network facade specifies the destination (IP:port) to which data should be sent. If multiple destinations are specified in command route then the network facade is called multiple times.

Network facade service component does the following tasks while also being responsible for validating the request and authenticating.

10 1) Sends and receives command over TCP and UDP protocols

2) In case a command should be sent over UDP and a datagram is too large it splits the datagram into smaller chunks

Audio / Video Synchronized

DirectShow is used only to capture and render video.

15 Two DirectShow filters are implemented in this preferred embodiment.

The first one captures video from webcam and sends media frames to the application, while the second one takes frames from the application and sends them to a video renderer.

The filters are trivial or of low complexity in this embodiment although others of greater complexity could be used. No processing is done in filters or anywhere in the DirectShow graph being constructed.

20 Video to audio synchronisation is done by the receiving peer by means of a library look up or time sequence according to the included time stamps. DirectShow is not used for this task.

On the peer sender side audio and video are initially synchronized. No buffering is done before sending audio or video packets into the connectionless network. On the peer receiver side audio and video may desynchronize because of:

25 a) Network issues. Packets can arrive arbitrary

b) Audio jitter buffer which introduces latency to audio stream

30 Preferably in this embodiment, audio/video synchronisation is being implemented using timestamps which are inserted or present in both audio and video packets. In this embodiment and by way of preference only the audio stream is a master stream. Thus before displaying the video component there is a check of the timestamp of the last played audio packet and in the case where the video packet has the equal or earlier timestamp application it is displayed immediately, otherwise display of that video packet payload is delayed

The preferred P2P protocol for this embodiment of the invention is Kademlia but with a modification previously described however briefly

35 a) A Location server is required to perform initial super-node selection. Client should somehow get the SN IP:port to connect to P2P network. This is where location server being used

b) This is a way to efficiently refresh Kademlia nodes routing tables. Instead of pinging all neighbour nodes of which the supernode is aware (there could be thousands of them), each Kademlia node listens for information from other nodes telling indicating that certain of the content of their own routing table could not be contacted. In this case a Kademlia supernode schedules a ping request to this identified node and if there is not response

within a predetermined ping conditions and fails the pinged node is removed from that supernodes' routing table. This helps to keep routing tables fresh.

"Logic," as used here in, includes but is not limited to hardware, firmware, software, and/or combinations of each to perform a function(s) or an action(s), and/or to cause a function or action from another component. For example, based on a desired application or needs, logic may include a software controlled microprocessor, discrete logic such as an application specific integrated circuit (ASIC), or other programs are logic device. Logic may also be fully embodied as software.

"Software," as used here in, includes but is not limited to 1 or more computer readable and/or executable instructions that cause a computer or other electronic device to perform functions, actions, and/or behave in a desired manner. The instructions may be embodied in various forms such as routines, algorithms, modules, or programs including separate applications or code from dynamically linked libraries. Software may also be implemented in various forms such as a stand-alone program, a function call, a servlet, applet, instructions stored in a memory, part of an operating system or other type of executable instructions. It will be appreciated by one of ordinary skilled in the art that the form of software is dependent on, for example, requirements of a desired application, the environment it runs on, and/or the desires of a designer/programmer or the like.

Those of skill in the art would understand that information and signals may be represented using any of a variety of technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

Those of skill in the art would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

The steps of a method or algorithm described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. For a hardware implementation, processing may be implemented within one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), processors, controllers, micro-controllers, microprocessors, other electronic units designed to perform the functions described herein, or a combination thereof. Software modules, also known as computer programs, computer codes, or instructions, may contain a number a number of source code or object code segments or instructions, and may reside in any computer readable medium such as a RAM memory, flash memory, ROM memory, EPROM memory, registers, hard disk, a removable disk, a CD-ROM, a DVD-ROM or any other form of computer readable medium. In the alternative, the computer readable medium may be integral to the processor. The processor and the computer readable medium may reside in an ASIC or related device. The software codes may be stored in a memory unit and executed by a processor. The memory unit may be implemented within the processor or external to the processor, in which case it can be communicatively coupled to the processor via various means as is known in the art.

It will be appreciated by those skilled in the art that the invention is not restricted in its use to the particular application described. Neither is the present invention restricted in its preferred embodiment with regard to the particular elements and/or features described or depicted herein. It will be appreciated that various modifications can be made without departing from the principles of the invention. Therefore, the invention should be understood to include all such modifications within its scope.

Claim defining the invention:

1. A method for creating a multimedia connection, each peer having video and audio sources, and the connection between at least three peers, the method includes the steps:

processing both audio and video sources of each peer provided in digital form with a packetizer; and

5 streaming those sources to multiple outputs to other of the peers using a Peer to Peer protocol.

2. A method of facilitating data communications between peers in a peer to peer network environment having video and audio sources associated with each peer, and the data communication being between at least three peers, the method includes the steps:

processing with a packetizer both audio and video sources of each peer provided in data form; and

10 streaming those sources after packetization to multiple outputs to other of the peers using a Peer to Peer protocol.

3. A system that facilitates data communications between peers in a peer to peer network environment having video and audio sources associated with each peer, and the data communication being between at least three peers including,

15 multiple computer hardware executing peer to peer software including a communications component having a packetizer for processing both audio and video sources of each peer provided in data form; and streaming those sources after packetization to multiple outputs to other of the peers using a Peer to Peer protocol.