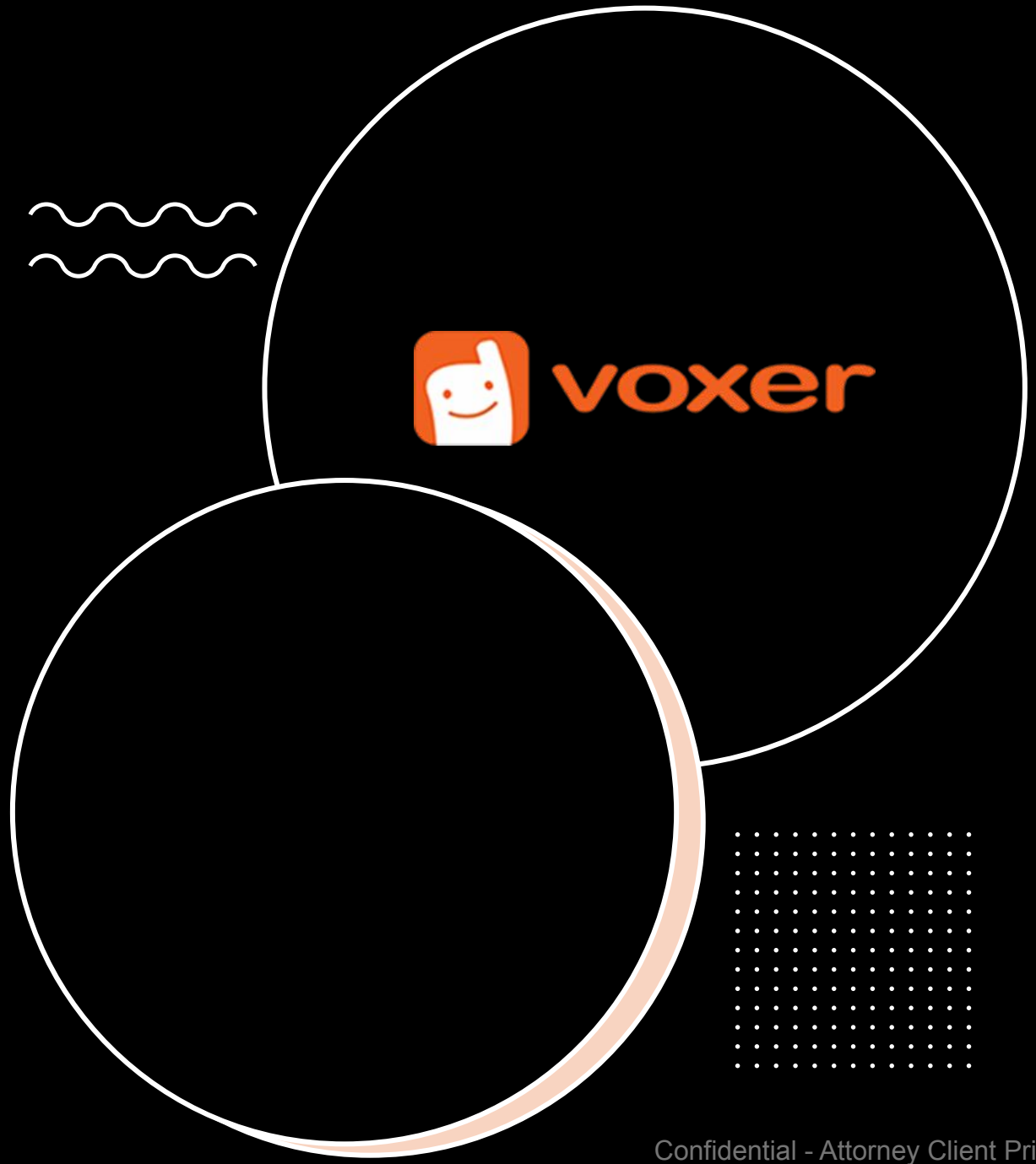


- Voxer is a communication protocol and architecture that combines the benefits of both synchronous communications (i.e., real time or “live”) and store and forward messaging, without the drawbacks of each (e.g., synchronous communications require end-to-end connections, while messaging is incapable of live communication).

- Wrapping your head around Voxer requires a clear understanding of both synchronous communication and store and forward messaging.

Synchronous Communication - Attributes & Examples

Requires end-to-end connection before communication may begin

Supports **live only** communication

Media is transitory – meaning it disappears once transmitted over the end-to-end connection.

No communication if connection:

- Cannot be established
- Breaks

"POTS" - Circuit connections

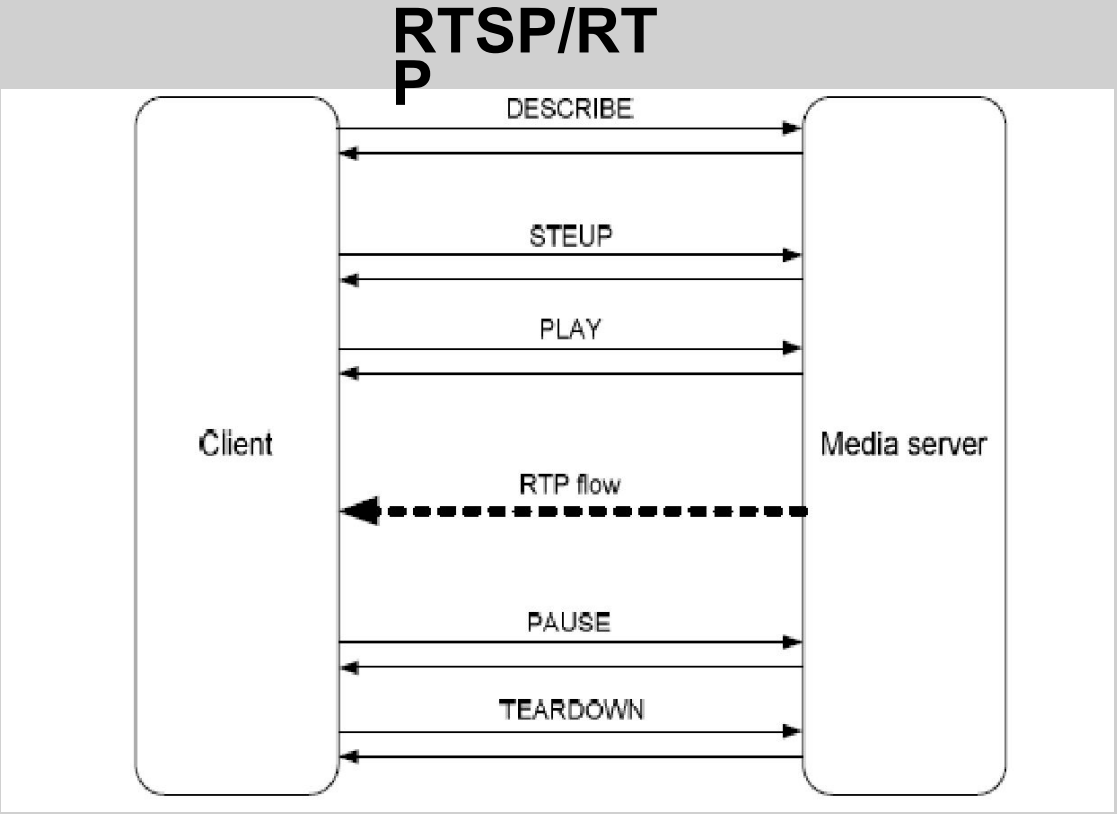
With radios, an end-to-end connection is established if both parties tune to the same channel/frequency

Synchronous Communication Over IP Networks

Requires a Virtual Connection Before Media Exchange

Virtual connection established using a setup protocol (e.g., RTSP) & sharing/agreeing on:

- Network locations
- Type of media to be exchanged
- Codecs to be used
- Transport protocol (e.g., RTP)

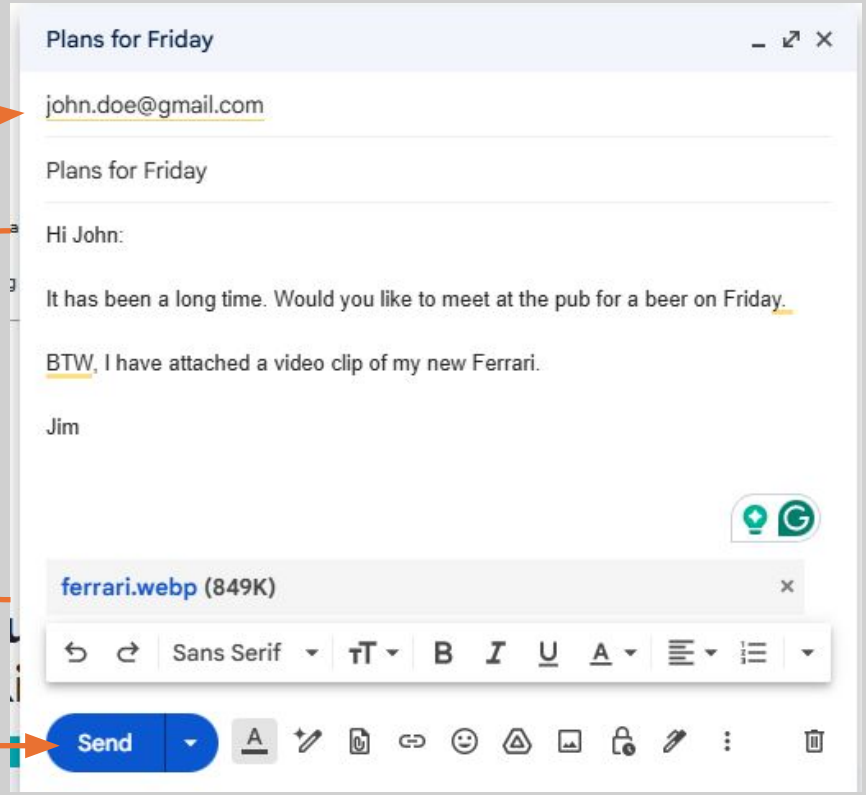


A Message is a Unit of Communication

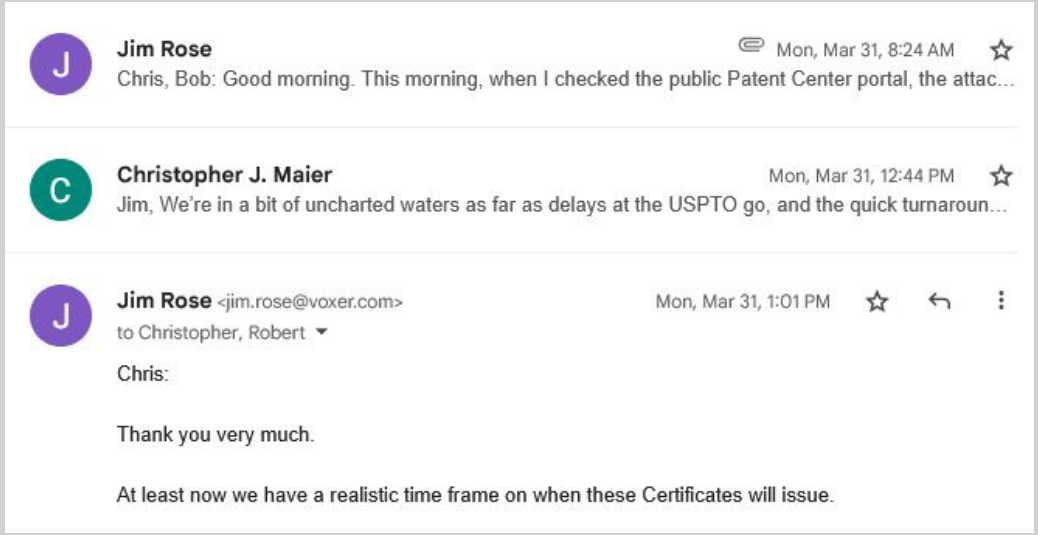
The unit of communication encapsulates:

- Header including Identifier info for addressing
- Body containing media

Transmission of unit of communication out of storage



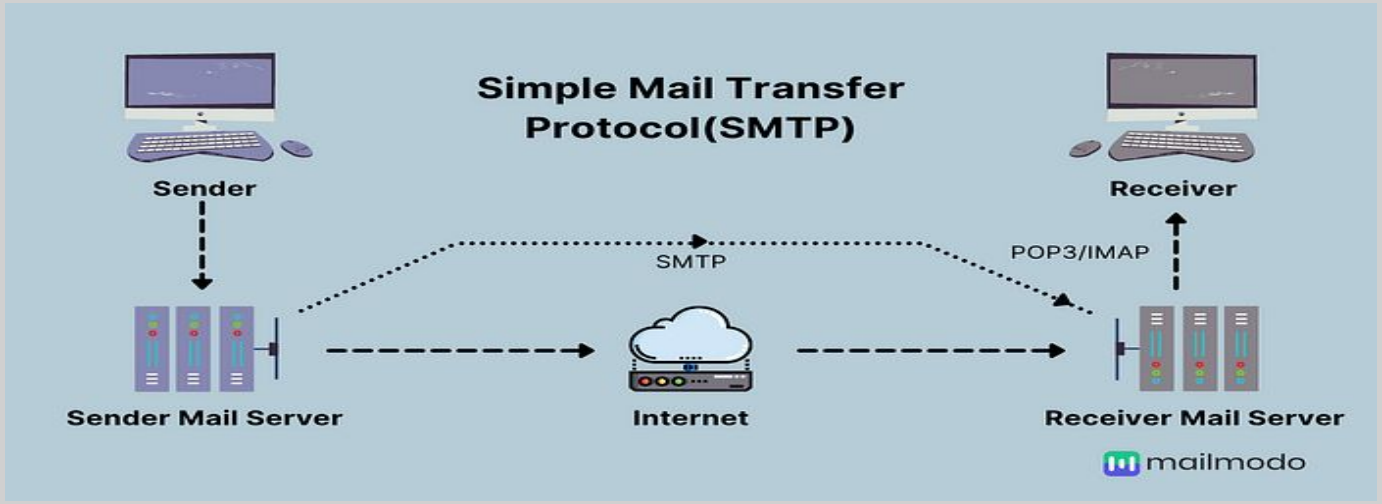
Messages can be threaded into a conversation



Messaging is Store & Forward

Sender

- Compose & optionally attach video/voice clip
- Transmission of unit of communication out of storage
- No end-to-end connection needed.
- The network relies on the embedded identifier information to ascertain recipient(s), route and deliver the unit of communication.



Network Server(s)

a. Store message
b. Check integrity
c. Use identifier(s) to ascertain next hop(s)
d. Forward out of storage

Repeat (a) – (d) at each server

Reliable delivery out of storage if recipient no immediately available.

Receiver

- Render out of storage
- Time-shifted only
- Never live

- **Voxer is messaging, but LIVE** (i.e., a recipient may optionally render while a sender is transmitting), like synchronous communication, ***but no end-to-end connection is first required***.

- **Vox messages share all the same attributes as conventional messaging**:
  - Vox messages are units of communication encapsulating identifier information & media.
  - The network relies on the embedded identifier information to ascertain recipient(s) & their location(s) for delivery.
  - Vox messages are stored – so can be rendered LIVE or in a time-shifted mode.
  - Reliable delivery - when no or poor network connectivity, messages can be later sent or delivered out of storage.
  - Message integrity - when network conditions are poor, media is gracefully degraded to maintain liveness, while a background process makes sure complete copies are eventually received.

‘270 Patent - Messages & Conversations Definitions

Message attributes (Col 8, lines 2-8)

- Conversation ID
- ID of sender
- Media (e.g., voice or video)

All messages are sent within a conversation (Col 8, lines 11-18)

Message: An individual unit of communication from one User to another. Each Message consists of some sort of Media, such as voice or video. Each Message is assigned certain attributes, including: (i) the User sending the message; (ii) the Conversation it belongs to; (iii) an optional or user created Importance Tag; (iv) a time stamp; and (v) the Media payload.

Conversation: A thread of Messages (identified, persistently stored, grouped, and prioritized) between two or more Users on their Devices. Users generally participate in a Conversation using their Devices by either Reviewing Messages in real time or in a time-shifted mode, or creating and sending Messages of a Conversation as desired. When new Messages are created, they either define a new Conversation, or they are added to an existing Conversation.

All Participants of a Conversation are Known

A Client Define Conversation Participants – See Col 37 line 55- Col 38 line 5:

Create a Conversation **1104***b*—A User creates a Conversation through the interface **110** by inputting a Conversation name, a list of Contacts, and an optional scheduled start time. If no start time is designated, it is assumed the start time is immediate. In response, the MCMS application **20** creates a new Conversation in the database **22**, associating records for each Participant on the Contacts list. The MCMS application **20** also creates in the database **22** Participant records for each User on the Contact list, allowing the caller to receive the presence information of the others on the Contact list. If the Conversation is scheduled, the MCMS application **20** starts the Conversation at the designated time. Otherwise, the Conversation starts right away.

Update Conversation Details **1104***c*—The User may make changes to a Conversation through the user interface **110**. For example, Participants may be added or removed. Any change in status of the participants is updated in the MCMS database **22**.

Within the network infrastructure, the MCMS database 82 on Server(s) 16 ***mirrors*** the MCMS database 22 on individual client devices, but for multiple clients. See Col. 22, lines 2-6:

~~embodiments,~~ the MCMS server application **80** and Store and Stream module **84**, including the MCMS database **82**, is configured to support many Users in one instance of the

How Voxer Ascertains Targets on the Network

- Since all messages identify (i) the conversation to which it belongs, and (ii) the sender, ***then all other participants of the conversation are ascertained as the targets***.

- Consider a conversation that includes Jack, Jill, and Mary, then:
  - When Jack sends a message, Jill and Mary are the targets
  - When Jill sends a message, Jack and Mary are the targets
  - When Mary sends a message, Jack and Jill are the targets

How Voxer Delivers Messages that can be Consumed **LIVE** without First Setting up End-to-End Connection(s)

- The Voxer infrastructure maintains connectivity with all clients connected to the network (i.e., knows their location).

- A sender merely identifies a conversation and transmits – The sender does not know or care if the conversation participants (i.e., the targets) are connected to or available on the network.

- On the network, the Voxer infrastructure uses the conversation ID and sender ID (i.e., identifier information) contained in Vox messages to immediately ascertain:
  - The target(s)
  - The IP address(es) of the client device(s) associated wit the target(s).

- The use of a ***STORE and STREAM*** protocol (as opposed to **store and *forward***) to ***continuously*** route media across the network to a recipient as the sender is transmitting

How Connectivity is Maintained with Clients

Clients run an “app” 12, which includes a store and stream module 24, which maintains connectivity with the servers 16 on the network

‘270 - Col 12, lines 5-22

D. Client Architecture

Referring to FIGS. **2**A and **2**B, a block diagram of a Client **12** running on a Device **13** is illustrated. As illustrated in FIG. **2**A, the Client **12** includes Multiple Conversation Management System (MCMS) application **20**, a rendering and encoding module **21**, and an MCMS applications database **22**. As illustrated in FIG. **2**B, the Client **12** further includes a Store and Stream (SAS) module **24** with a Persistent Infinite Message Buffer (PIMB) reader **26**, a

‘270 - Col 17, lines 47-49

The Store and Stream module **24** maintains connectivity with all target recipients (e.g., Servers **16** or other Devices **13**) on the underlying network **18**, manages all message,

What does Connectivity Mean?

Connectivity is maintained by running a background process between the Voxer infrastructure and each client device running the Voxer app:

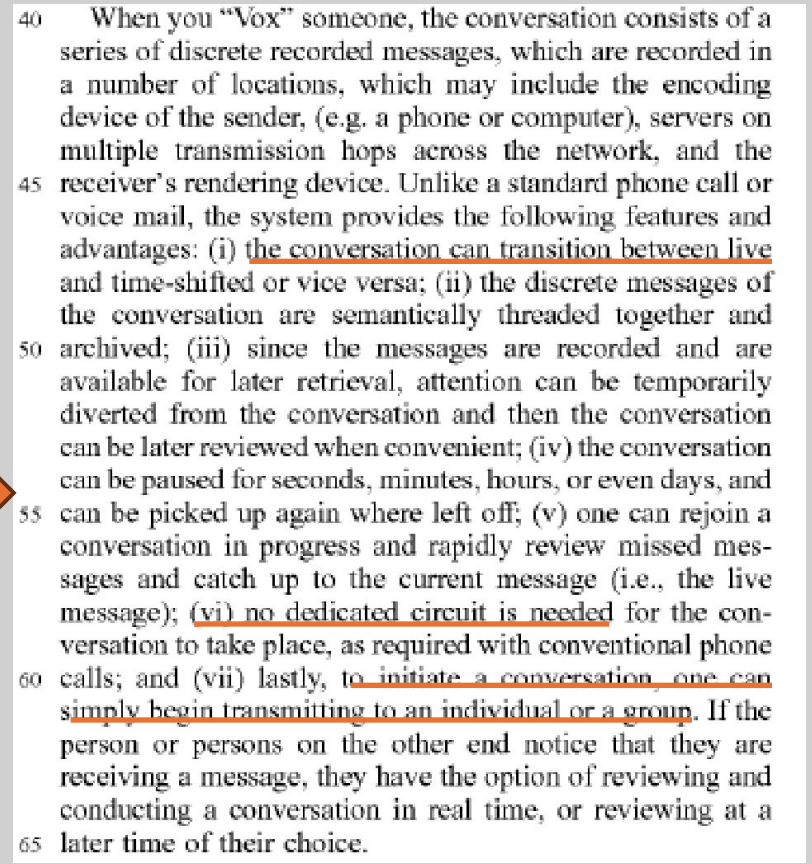
- IP addresses are dynamically assigned as mobile devices roam from network to network.
- The Voxer infrastructure periodically “pings” each client.
- In response, each client provides its ***current*** IP address.
- The Voxer network infrastructure thus knows the current IP address of all the client devices connected to the Voxer infrastructure.

A Sender Simply Selects a Conversation & Transmits

The sending client does not know or care if recipients are connected or available on the network because the server infrastructure on the network is responsible for ascertaining recipient(s) and their availability on the network using the identifier information contained in a Vox message. See:

Summary of the Invention in the '270 patent, **Col. 4, lines 40-65**, explains among other attributes that sending a "Vox" means a recipient may optionally (i) render live, (iv) no dedicated circuit connection is required, and (vii) a sender can simply begin transmitting to an individual or group.

Hence, messages are sent without first establishing an end-to-end connection.



When you "Vox" someone, the conversation consists of a series of discrete recorded messages, which are recorded in a number of locations, which may include the encoding device of the sender, (e.g. a phone or computer), servers on multiple transmission hops across the network, and the receiver's rendering device. Unlike a standard phone call or voice mail, the system provides the following features and advantages: (i) the conversation can transition between live and time-shifted or vice versa; (ii) the discrete messages of the conversation are semantically threaded together and archived; (iii) since the messages are recorded and are available for later retrieval, attention can be temporarily diverted from the conversation and then the conversation can be later reviewed when convenient; (iv) the conversation can be paused for seconds, minutes, hours, or even days, and can be picked up again where left off; (v) one can rejoin a conversation in progress and rapidly review missed messages and catch up to the current message (i.e., the live message); (vi) no dedicated circuit is needed for the conversation to take place, as required with conventional phone calls; and (vii) lastly, to initiate a conversation, one can simply begin transmitting to an individual or a group. If the person or persons on the other end notice that they are receiving a message, they have the option of reviewing and conducting a conversation in real time, or reviewing at a later time of their choice.

The Voxer Infrastructure on the Network Resolves Identifier Information into IP address(es) by:

1. Ascertaining the target(s) from the conversation ID and sender ID in the message

'270 - Col 22, lines 41-49

~~Clients **12** and transmit Messages to other Clients **12**~~. When a Client **12** transmits a message, the message contains an identifier associated with the target recipient. In response, the Server **16** finds a delivery route to a Client **12** associated with the intended or targeted recipient. As Messages are received, they are stored in the PIMB **85** and transmitted to the next Server **16** (i.e., the net "hop") of the network layer **14** along the path to the intended recipient(s), or to the recipient(s) directly depending on the system configuration.

2. Relying on a Domain Name System (DNS) to resolve the IP address(es) of client device(s) belonging to the target(s).

'270 - Col 26, lines 5-17

The address of a target Client **12** needs to be known so that the underlying network delivers the Vox packet **95** to the correct location. With IPv4 networks, the address is typically an IPv4 Address, which is a 32-bit number that uniquely identifies a host within the network. For other networking technologies, the address could be some other type of identifier. IP networks use the Domain Name System (DNS) to resolve human-readable names into IP addresses, and the Address Resolution Protocol (ARP) to resolve IP addresses into physical addresses. Regardless of the underlying networking technology, the system **10** uses one of the above-mentioned or other known addressing schemes for delivery of Vox packets **95** to the correct location.

Voxer's Store and ***STREAM*** Protocol

Voxer's protocol ***progressively*** streams and stores media as it is created, allowing a recipient to render a Vox message as the media is created by a sender and transmitted across the network.

Also see in the '270 patent the detailed discussion of the store and stream protocol respect to Figs 8A through 8F, Col. 27 line 1 through Col. 29, lines 49. In this section of the '270 patent, a complete discussion is provided explaining how media on an originating device is continuously or progressively processed as it is created, continuously transmitted, and processed by or more servers 16 on the network and continuously delivered to recipient(s) for optional rendering.

'270 -Col 15, lines 11-29

The Conversation/Message management service **20***f* processes the transmission and receipt of Messages in a progressive fashion. When transmitting, Media may be created while Messages are simultaneously encoded, stored and transmitted. In other words, the transmission of Messages may occur simultaneously with the generation of Media by the User (i.e., while speaking into their Device **13** or generating video). On the receiving side, the receipt, storage, and Rendering of Messages also all occur progressively. Messages do not need to be completely received before they can be Rendered. The Rendering of Messages may occur at the same time Messages are being delivered, right up to the MTSD. Further, the service **20***f* is also capable of the simultaneous transmission of outgoing Messages and Rendering of incoming Messages. The progressive nature of the service **20***f* allows a User to be engaged in a live Conversation while storing and streaming the media of the Conversation for later retrieval and review as well other functions described herein.

How Voxer Solves Conventional Synchronous Communication Problems

Problem	The Voxer Solution
Live Only	Voxer can be **LIVE** and **TIME-SHIFTED**. Since messages are stored, recipients can render in a time-shifted mode at a time of their convenience.
Requires first setting up an end-to-end connection	Voxer does **NOT** require an end-to-end connection. A sender can simply transmit regardless whether the recipients is/are connected to the network or not.
Requires the connectivity & attention of target recipient(s)	A sender can simply transmit regardless of whether the recipient connected and/or paying attention or not.
Media is transitory	Since all messages are stored on the network and/or locally, no message is lost or missed if rendered out of storage.
Simultaneous participation in multiple conversations is challenging	Vox messages are units of communication that can be stored and strung together in conversation strings. These attributes mean recipients can readily participate in multiple conversations by transitioning their attention from one conversation to another and participate in a given conversation either live or in a time-shifted mode.
No connectivity means no communication is possible	Messages can still be created or delivered out of storage when a sending device or a receiving device later reconnects to the network.
Poor connectivity possibly means degraded or no	Messages are transmitted as fast as the network permits to maintain liveness. A complete copy is eventually

Voxer Videos Worth Watching

https://vimeo.com/82119419

https://vimeo.com/74328527