Instant Messaging Systems: *talk*

Text messaging is not a new concept – it dates back to the earliest days of the ARPAnet (forerunner to the Internet). There were concerns initially with the "legality" of such a feature used for non-research purposes since the use of the network was limited and excluded "personal use".

There were several problems with early text message systems – you essentially had to know which specific machine the person was using in order to initiate communication using a program such as *talk*. There was some support for finding people online such as *finger*. The finger system was primary designed to provide information about a specific user on a specific machine. Extensions to this idea had a single machine in a domain keep up to date information about users, but there was still the need to know the domain in which the person was working.

Once you knew that the person with whom you wanted to text message was online at a specific terminal, you could send a *talk* request to that user at that machine. You could not simply type something such as *talk egolub* and have it "find" the desired user for you – instead you would need to type something such as *talk egolub@rac3.wam.umd.edu* (user name and specific machine name). The *talk* protocol required you to know the exact hostname or IP address of the machine the target user was using in order to have a conversation with them.
Instant Messaging Systems: RELAY chat

Over the years, several systems have been created where users connect to a main server regardless of where they are locally connected. Some of these were telnet-based chat rooms (which have mostly vanished). One classic example of this was the RELAY chat system. Users would connect to one of several RELAY servers, and all users connected to that server could chat in public or private “rooms”. If two users connected to two different RELAY servers, they would not be able to talk with one another, or even know that the other was online.

This resource was limited to certain machines on certain networks (there was still no single Internet unifying all individual networks).

Later came IRC (Internet Relay Chat) and then other systems such as ICQ (“I Seek You”) and AOL Instant Messenger. All of these rely on a centralized server keeping track of all the users currently connected to the system. However, similar to the RELAY servers, if you are logged into your AOL messaging account and your friend is logged into their MSN messaging account, you will not be able to communicate, or even see that they are currently online. There are some client applications (such as Trillian) that allow you to sign into all of your accounts at once so that you can interact over several systems through a single front-end application.
Modern Instant Messaging Overview

Instant messaging: Systems like AOL Instant Messenger, Microsoft Messenger, and Google's Gmail chat enable real-time communication combining capabilities such as one-on-one chat, chatrooms, file transfers, audio conversations, and even video conversations. The clients you run are often used to provide secondary information such as streaming stock market reports (such as within the buddy list window).

Components:

IM User Application – This allows users to connect either to the central IM server or directly to another user, and to send and receive data (ie: text messages, images, files, audio, video).

IM Server – Allows clients to “register” themselves and their current Internet location (ie: IP address), coordinates alerts such as buddies logging on and off and their status, supports message delivery for regular IM, facilitates direct connections between users.
Instant Messaging Modes

“Standard” IM text messages all go through a central **server**. For example, with AIM, each text message you send goes to one of several central AOL servers and then gets sent to the application running on the recipient’s machine. In this scenario, the IM application on your machine is running as a **client**.

If your IM user application supports it, you can elect to make a direct connection to another user’s computer to send text, images, audio, video, or files “directly” to that user. In this scenario, both your IM application and your buddy’s are running as peers to one another.
Security

As we've discussed, when you send information across the Internet, it is divided into individual packets of information that are then passed from machine to machine towards their final destination. Your IM conversation is potentially being sent past dozens of machines on its way to the AOL server and then dozens more on the way to your buddy's computer. Even if you make a "direct connection" with your buddy, the packets will pass by many other machines on the Internet – just not the AOL central servers.

In order to have private conversations, you need the information in those packets to be encrypted so that only your buddy's computer is able to decode and present the original information. There are a variety of clients that provide encryption services. Trillian has this ability built in. There is a plug-in available for GAIM called OTR ("Off the Record") messaging. One interesting issue addressed by OTR is that while you are able to be sure that the messages are private and unaltered during the conversation, once a conversation is complete, any logs kept can not be proven to be unaltered transcripts.

What do you consider part of "security"? Privacy? Accuracy? Deniability?
File Transfers via IM clients

One of the features commonly provided by IM software is the ability to make a direct connection with your buddy's application to exchange information "directly" rather than through the central servers.

There are some very practical reasons that organizations such as AOL don't want files transferred using them as an intermediary. One of these is size. A simple digital image of fair quality can be around 1 MB in size. This equates to approximately 1 million bytes of information. A single text character can be stored in a single byte. One million characters is quite a lot. Consider that there were fewer than 30k characters in the United States Constitution. Chapter 5 of our textbook contains fewer than 90k characters (and it's not a short chapter).
Direct Connections

While there are many scenarios where using a "direct" connection has advantages (or is even mandatory) they can present a technical challenge.

If you have a home network you might also have a software firewall in place. In the simplest sense, a software firewall can prevent "unsolicited" incoming packets of information. This means that if you initiated communications with a remote machine, it will be allowed to send information back but no foreign machine's packets will be allowed to reach your machine to ask you to start communicating with it. In the case of "direct" connections, one of the two machines (if not both) will need to be able to allow an initial incoming request for communication.

With regular IM, your machine always initiates communication with the IM server, and all incoming messages are routed through the server. Here you do not have that situation.