The Systems of Instant Messaging

Devashish Purandare
University of California, Santa Cruz

Submission Type: Survey

Abstract

Instant Messaging has captured the imagination of humanity for centuries. With better and faster modes of communication, we have been able to improve the quality of communication and reduce the delay. Improvement in communication has indeed brought the world a bit closer. In this paper I wish to trace the beginning of electronic instant communication, mainly digital communication. Later this paper presents the state of the art in Instant Messaging (IM). With a background on the development of communication, we will focus on modern protocols for IM and how they tackle some of the biggest problems faced by Instant Messaging.

After analyzing the state of the art, the paper presents some of the extensions to these systems as well as privacy and security aspects.

1 Introduction

We can trace back the first instant messaging system as early as the year 1792 with the invention of the electric telegraph. Since then we have been through generations of technologies, each better than the one it succeeded. While the core principles remain the same, each year more and more things are made possible with IM, and on a scale never seen before. There are 4.7 billion cell phone users worldwide [1]. This explosion in cell phone usage, coupled with improvement in network coverage with high speed data access has caused instant messaging to grow on a massive scale. Several popular instant messaging apps boast of more than a billion unique users.

This scale presents several challenges which are unlike what are faced by most distributed systems. Each client is unique and the connectivity to each client is unreliable. There could be multiple instances of the same client on different architectures. While consistency is best-effort, there's a need to maintain at least causality. With clients located at geographically different locations, in different timezones, the need for security and privacy, and network, battery constrains make IM systems very challenging.

2 Motivation

While a lot of work is work has been done on development of instant messaging services, little has been done on aggregation of all the techniques into a single cohesive compilation. This paper aims to trace the history of instant messaging, the design decisions that went into the instant messaging systems of each era according to the constraints, the exponential growth of IM, and what the future holds. The paper will focus on 5 main areas, Chronological timeline of IM, the challenges each protocol faced and the way those were addressed, what is the state of the art. The next section is on new technologies and the modern extensions to IM with a special focus on privacy and security in IM. The aim is to create a definitive document tracing the history of IM and the modern techniques.

3 Organization of the paper :

The paper is organized chiefly as follows section 4 will focus on historical systems, section 5 will address the arrival of Internet and subsequent systems and section 6 will be about current systems. In the next section, privacy and security aspects are discussed. The paper written around three main events which gave a boost to messaging, invention of the Telegraph, the arrival of Internet and the arrival of Smartphones. While internet created a world-wide network of connectivity to allow for instant communication, smartphones helped improve the reach of the web, with internet connectivity reaching every person as well as lowering the technological barrier for using IM.

4 Before the World Wide Web

The need for instant communication has been around ever since the dawn of civilization. Smoke signals were used by the Greeks to communicate in wars, and as we will see, the need to communicate in wars has given birth to a lot of systems of communications as well as some of the most fundamental problems faced by distributed systems [18]. The first successful telegraph network was operated in France as early as 1793. While the early Telegraph systems used optical communications, with the arrival of electricity, electric telegraphs using the Morse code became the first major worldwide communication system. The research on Hertz-waves gave rise to radio communication, allowing wireless transport of communication. Telegraph lines were laid under the sea in the first ever world-wide network for instant messaging [21].

4.1 The Electric Telegraph

Electric telegraphs worked much like telephone line. The sending station requested a connection to the switching center, the switching center would set up a session and a line between the sender and the receiver, and data could be transmitted over it. The switching did not allow for communications to a node when it was a part of another session. Because of the centralized nature of Telegraph system and the low number of messages, this was a feasible and practical solution for Telegraph. Early communications over phone line, such as dial up internet and fax would later the same technique.

It might seem tangential to include telegraph in this survey, but Telegraph provides surprising similarities and for much of its lifetime, faced the same challenges that messaging systems today face and gave rise to many of the methods that are used widely today.

As Telegrapgh gained popularity, the centralized switching network soon began to create hot-spots at busy switching hubs like London. This led to slowdowns and congestions, with messages getting delayed or lost due to the flood of messages at the switching centers. Telegraph companies found a rather igneous way to get around around this. They set up steam powered pneumatic lines, which forwarded messages to nearby centres from a busy center, who then processed and sent the message over the telegraph line.

Each message had a header space, where it was "Timestamped" by the receiver with the source and destination address, and by the time it reached destination, it had the complete history of the path it had been routed through. More than a century later, email would use the exact same technique. The Telegram centers which spoke frequently, set up a Nickname system to reduce the size of this header.

It is fascinating to see the similarities these physical systems share with the modern digital systems. The transmission of messages, which involved human operators reading and retransmitting the message, gave rise to codebooks, ciphers and encryption, not too dissimilar from the techniques that are used today. Paul Reuters (founder of the Reuters agency) famously used to send 3 homing pigeons with the same message to guarantee reliable delivery. The cost being dependent on the length of message gave rise to techniques for data compression and shorthand techniques.

There are many parallels between the uses of Telegraph and the current messaging systems. Telegraphs were famously used for playing games, transferring money, exchange of information, and even for 'online' marriage services. The operators at telegraph decoding stations would use the service for informal meetups and chats after working hours were completed. [47]

As we move along the time, we will see, much of the progress in messaging, is just reinventions to adapt the same technology to a newer medium of information flow. With telegraph and telephone, it was electricity, with email and IRC it was Internet, and with modern messaging systems, it is smartphones.

4.2 The Telephone

The discovery of Telephone was no accident. Alexander Graham Bell, working on a harmonic telegraph, a device which used frequency of sound to transmit messages, realized it would be possible to transmit sound over electrical lines directly, eliminating the need for the message to be decoded at the other end. The first working Telephone in fact was marketed as a speaking Telegraph machine which required to manual interpretation. Telephone became an instant success, building on available telegraph infrastructure, it largely led to the decline of Telegraph.

The telephone followed a very similar path to the Telegraph, and its development was led by many of the same people who worked on Telegraph systems - Thomas Edison, Alexander Graham Bell. From transcontinental telephone lines being set up to a trans-Atlantic line, it followed the same pattern as Telegraph, and the same path that the internet would follow decades later. One of the biggest achievements of the Telephone was that it managed to free communication carried by and developed by mostly postal and railway services to something which private citizens could own.

Much of the research on reducing congestion in Telegraph was carried over to telephones, with duplexing and quadraplexing to boost bandwidth and modulation to reduce the amount of data to be transmitted.

The development of Hertzian Radio waves, which could be used to transmit sound over short distances, soon gave way for wireless radio communication, typically over short distances. This led to the invention of one of the biggest catalysts for instant messaging development, the mobile phone.

4.3 The Mobile Telephone and Wireless Communication

First introduced in 1973, mobile phones took the world by storm. They led to some of the most important events which would shape the future of communication. Interoperability requirements led to creation of standards and governing bodies which would work on open protocols. Mobile telephony also made communication cheap and personal with every person having their unique communication device.

Mobile telephony led to the development of GSM, CDMA, EDGE, GPRS and several other technologies which modern systems use. Data communication improved leaps and bounds every generation to introduce 2G services, faster 3G services and the modern 4G or 4^{th} generation services.

4.4 The Short Message Service

4.4.1 The Technology

In the year 1984, while working with GSM networks, Friedhelm Hillebrand and Bernard Ghillebaert realized that, the control signal mechanism to diagnose and configure GSM could be used to relay short messages over mobile networks. The medium, optimized for telephone put restrictions on the payload such that only 160 characters of 7 bits each could be transmitted. SMS was one of the first open platforms for application and vendor independent messaging. While initially introduced for GSM networks, it was soon ported to work with GPRS and CDMA networks. The big appeal of SMS was its global compatibility, SMS could be sent over virtually every network worldwide with just a software update to the existing infrastructure.

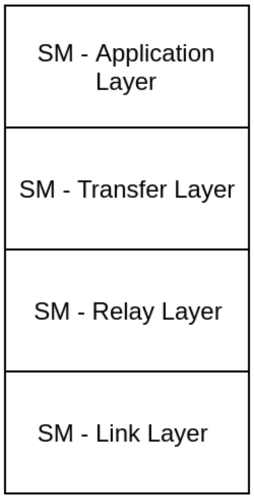

Figure 1: The architecture of SMS

SMS allowed peer to peer communication, passed through a wireless carrier, with each message limited to 160 characters.
Terminology :

- Short Message Entity - SME - application that sends and recieves SMS
- Service Centre - SC/SMSC - route, store and forward messages.

The SMS protocol has a set of features : SMS can get and generate delivery reports, add a reply path, execute an application and act as a command. Due of the intermittent and bursty nature of mobile networks, particularly 2G networks, the SMS protocol allows store and forward messaging. When a SME sends a message, the SMSC keeps a copy of the message and forwards it to the next base station. This goes on until the message reaches the station that has access to the receiving SME. If the device is not connected to the network, the SMSC holds the message for a specific amount of time waiting for it to reconnect before discarding the message. This specific amount of time can either be specified by the message originator, but to simplify communication, most companies use a validity period of 48 hours (2 days).[20]

SMS has a big limitation that it is restricted to a size of 160 characters * 7 bits. SMS uses UCS2 unicode's precursor to UTF-16, a character set which allowed complex characters such as Chinese and Arabic Script encoded in a small size [15]. However this makes it particularly limited when sending messages in complex scripts which require more space - such as indic scripts or emoji, limiting the characters to 72 or lower. While there was disagreement on whether the length of SMS was actually limiting, concatenating multiple SMSs allowed operators to offer greater character limits. Concatenating was introduced later, where carriers used multiple messages with sequence numbers and metadata to send a bigger SMS [13].

4.4.2 Layers of SMS architecture

Application Layer

This includes all the applications that make use of SMS including games, banking, rental, value added services.

Transport Layer

Message is encoded in a sequence of octets and made ready for transmission with details such as return address.

Relay Layer

The relay layer is responsible for the 'store and forward' routing of messages.

Link Layer

Link layer is the actual physical medium over which messages are transported.

4.4.3 Impact of SMS

SMS gained immense popularity because of its global compatibility and interoperability. The low cost of sending a message, the ability to receive a message even in adverse network conditions, the simplicity of sending a text, and the low barrier of entry made SMS the most successful forms of instant communications in the world. It is estimated that 6.1 trillion messages were sent in the year 2010, that is about 200,000 messages every second. 5.3 billion subscribers use cellular networks and wireless networks have improved the reach of networks where wired networks could not [17]. This provided billions of people worldwide a cheap and easy way of communication.

SMS spawned several products, directly or indirectly which had great impact on the world. Due to the size limitation on early cell phones, most phones had a 12 key keypad which made it very hard to type. Several algorithms were invented, such as T9, to allow predictive typing which would make it much easier to type of cell phones. Much like the telegraph, SMS length limitations soon spawned a language of there own, simplifying and shortening many words to make typing them faster and sending them easier. The human emotion is not limited by words, soon using expressions like `:)` to denote a smile or `<3` to denote a heart became popular.

As the popularity of ASCII emotion icons (emoticons) grew, companies began to parse and integrate animated, colored or custom versions of emoticons in their applications. To distinguish their service, Japanese network carriers started offering custom emoticons (Emojis) to users. Noticing the popularity of emoticons, unicode started adding emojis to the specification in an effort to standardize offerings. Unicode now offers over 2000 emojis, with options for skin tone, gender and professions [14].

Emojis were another instant success, in 2015 "😂" was declared as the word of the year by Oxford dictionary, noting it was the most searched word that year. [9]. As with telegraph, an industry soon appeared around SMS, SMS stock updates, sports updates, offers, SMS payments, SMS verification.

SMS allowed two-factor authentication, a security measure that would make evrything much more secure and tedious. The 160 character limit of SMS was adopted by Twitter, limiting tweets to 140 characters. The idea of limited text, instead of being restrictive, spawned a new generation of social interaction. Many companies modified their services so that they could be completely accessed using only SMS. Today, every phone plan in the US comes with unlimited text messages.

4.5 Enhanced Messaging Service and Multimedia Messaging Service

4.5.1 Enhanced Messaging Service

Enhanced Messaging Service (EMS) was introduced as an improvement to SMS. It was backwards compatible, so it could fall back to SMS if the device did not have EMS capabilities. EMS was a joint effort by several mobile phone manufacturers and it was completely implemented in the application layer of SMS so that the data was sent over SMS. EMS added several capabilities to SMS, adding the ability to send and receive bitmaps. EMS also added rich text formatting capabilities such as making text **bold** or *Italic*, changing the font, changing the font color and size. EMS also allowed the capability to send sounds, either predefined (where the actual tone was not transmitted) or custom. Custom sounds used the iMelody format to specify notes, composer and other metadata, allowing additional operations such as controlling lights and vibrations [4]. EMS also allowed specifying 4 8x8 or 2 16x16 images to serve as an animation.

EMS was further improved upon to Extended-EMS. Unlike EMS, it was not backwards compatible, but it extended several capabilities of EMS by adding options to send bigger images, 64 bit color images, polyphonic ringtones and vector images. While Extended EMS offered a lot of versatility, the underlying medium was still SMS.

4.5.2 Multimedia Messaging Service

With improvement of networks to 2G and 3G, transmitting data became easier, faster and more reliable. There was a need to ensure interoperability between the internet and phones. SMS and EMS were defined completely in the bounds of the network architecture and development of MMS required a lot of standardization, mainly with Wireless Application Protocol [25] and internet protocols. MMS added interoperability between SMS and email. Development of MMS required extensive collaboration between 3GPP which defined the content structure and format of MMS and WAP forum which adapted it for the internet. MMS added several important capabilities to instant messaging : the ability to send pictures and videos, the ability to send documents and email and the capability of voicemail.

In MMSE (MMS Environment) the MMS server stores all the messages which are sent over MMS and the subscriber gets a notification when a message is sent to him. The subscriber can then retrieve the message from what is stored on the MMS server message store. The MMSE includes several components and interfaces :

- MM1 - Interface between the user and MMSC (Multimedia Message Service Centre).
- MM2 - Interface between MMS Relay and MMS Server.
- MM3 - Interface between MMSC and external servers - email/SMS
- MM4 - Interface between two MMSCs for forwarding messages.
- MM5 - Interface between MMSC and routing information.
- MM6 - Interface between MMSC and User databases.
- MM7 - Interface between MMSC and Value Added Services - VAS.
- MM8 - Interface between MMSC and billing services.

Out of these MM2, MM6 and MM8 were never implemented or standardized. MMS was transmitted over TCP with a WAP proxy on top allowing interoperability. Hence MMS routing followed the handshake, data security and reliability guarantees offered by TCP.

MMS followed the specifications of [RFC822] [22] and [RFC 2822] [41] which were defined for email and later adapted to MMS. MMS followed and gave rise to the popularity of several W3C standards such as XML, SGML, HTML, XHTML, WML. MMS are routed through two different methods either by email eg. `user@mms.serviceprovider.com` or via Multimedia SISDN with unique identifying numbers.

MMS works in a similar way to SMS however it makes certain modifications for delivery. Because network connectivity is not guaranteed to be fast, the user just receives a notification for a MMS and can defer the retrieval of the actual message later. Deferring retrieval makes the MMS server store the messages for a longer amount of time. For the files which are too large, The Real Time Transport Protocol (RTP) and the Real Time Streaming Protocols (RTSP) were used to stream data bit by bit to the device [45, 46]. RTP and RTSP use time stamps and sequence numbers to order data and provide streaming.

While it did not gain the popularity that SMS did, MMS gave rise to many important standards and ways of communication of data. MMS gave rise to concerns of sharing copyrighted materials and gave rise to standards of Digital Rights Management (DRM). MMS also gave a boost to various established standards of data exchanges such as SOAP and XML. Most importantly, MMS bridged the gap between the Internet standards and Cell phone standards allowing for easy exchange of data.

4.6 ARPANET and the dawn of the Internet

Advanced Research Projects Agency Network (ARPANET) began as a research project in a joint initiative by the US department of defense and universities across the US. It was the first packet switched network implementing the TCP/IP protocol on such a large scale. Soon the ability to send message was added to ARPANET via various RFCs [19, 23]. Ray Tomlinson, regarded as the inventor of email used the "@" operator to differentiate between user and host name in the address. This was further extended to become `user@host.domain`, the standard email address formats that are widely used these days.

4.7 Other technologies :

Various other technologies existed for instant transmission of messages, such as messaging over Bluetooth in a peer-to-peer local network. Various technologies also allow sending messages over infrared signals, or through a local network such as LAN or WiFi.

5 The World Wide Web

The Internet, due to its scale, nature and openness soon became the ultimate medium for transport of messages. Much like telegraph, it became a way to transport data across the world at an unprecedented speed and a very low cost. This gave rise to the acceptance of instant messaging as form of formal communication and it became the de-facto medium to do so.

5.1 Email

With the ability to send messages over the ARPANET, a new application was devised, a way to send messages electronically [35, 39]. The format for email was later standardized [41]. Email uses a header field to keep track of routing. When an email message is received, it contains a complete history of the path it followed in a routed network to reach the destination. The ability to send images and files through email was added later when Multipurpose Internet Mail Extensions (MIME) were introduced [28]. This improved the versatility of email, and allowed fast and reliable exchange of data.

Emails soon became the de-facto method of communication, including communication for official and government purposes. The nature of email kept communication formal, allowing for much wider use along the lines of letters which could be both used for formal and informal communication.

Emails faced problems with serializability as messages arriving out of order can cause problems. The way emails get around this is a rather simple and straightforward way. Emails are managed in 'threads.' That means each incoming mail keeps a complete history of all the mails that preceded it and led to it. This allows receiving email applications to quickly fill in the gaps and allow seamless communication which follows a sequential order. This however has a major drawback, with each new email in a thread, the size of the payload goes on increasing. This can be further optimized by keeping a recent k history instead of the entire history.

Emails had a huge influence on instant messaging and remain one of the most used instant communications in the world. Email led to many problems - viruses, scams, spam messages, illegal deals, leaks and hacking and many solutions to these problems. The most popular email services are used by billions of users, and remain the primary information required to subscribe to the services of any website. Many businesses are built around email, such sending, forwarding and managing emails for large organizations and setting up an infrastructure. Emails are used for official purposes more than any other instant communication platform.

5.2 Chatrooms

The rise of popularity of Internet and its widespread use gave rise to message boards for people interested in a certain topic. Message boards would be websites, discussion forums, news and marketplaces. Soon websites started allowing users to participate in real-time conversations giving rise to “chat rooms.”

5.2.1 Internet Relay Chat (IRC)

IRC was created as an open text-only messaging protocol which would use TCP to connect to IRC servers creating an IRC network. IRC does not specify the text encoding and the RFCs [30, 31, 32, 33, 37] used for reference are rarely used which gave rise to many issues initially.

IRC Server protocol [33] pings all its connections periodically polling them, if a connection is unresponsive, a termination procedure is followed. Connection of two servers is a critical and error prone area in IRC. IRC was designed for slow networks, often dial up networks over telephone lines. To speed up data transmission, IRC servers support compression of data streams being sent. However this causes further problems in connection and forwarding of messages since all servers in a network, particularly external servers may not support compression.

To ensure consistency, IRC uses state information, mainly the information about the states of clients, servers and channels. This state machine information can then be used to determine where the connections should be cut off as IRC can only work in acyclic graph formation and formation of cycles is considered a collision. Collision can be because of two servers connecting to each other despite there being a path between them or because server nicknames turn out to be the same in a large enough network. When a server is terminated, the server sends a SQUIT message to all other servers to ensure that the network remains consistent.

Another measure that the IRC protocol takes is that each server maintains a record of all the recent nicknames so that in case of a server split or nickname change race condition, the network would continue to function and have information to resolve the conflict.

IRC also offers a way to deal with rogue or malicious client flooding the servers with messages by keeping a track of timestamps of the messages. After a certain threshold, each message from the client is penalized with a time delay (initially 2 seconds) for each messages, the delay can be incremented with time.

IRC had many problems from start such different standards of implementation causing interoperability issues. IRC is not very scalable fundamentally due to the requirement that each server should know about every other server and network can only be an acyclic graph. This also presents several privacy issues. IRC allows the users to pick labels for the nickname, channel name and service name. With no duplicates allowed, this often results in collisions. The channel and server lookup causes great problems with scaling with implementation algorithm having complexity $O(n^2)$.

IRC became very popular with over 1 million unique users every day, but its popularity was restricted to certain demographics such as developer communications, inter company collaboration or discussions about a topic. While it never gained the popularity enjoyed by the likes of SMS, it gave rise to collaborative messaging and data interchange apps like Slack and Yammer which took the core concept and implemented it with modern technologies. As of 2016, IRCv3 is in works, adding protocols for user presence, file, image, audio and video exchange and adapting IRC for modern networks.

5.3 eXtensible Messaging and Presence Protocol (XMPP)

XMPP was originally open sourced as Jabber protocol and is one of the most widely used protocols for instant messaging. As its name suggest XMPP is based on XML format. XMPP is a part of open standards and is maintained by the XMPP work group a part of IETF.

XMPP was defined as an open internet standard with RFC6120 [44] in 2004 and was developed upon to create an open standard for messaging and presence.

5.3.1 Presence

Presence allows clients to subscribe to and ask about the status of a particular client. This allows messaging applications to provide richer experience to users, showing the other user as "online", "busy", "away." This can be a very useful feature for users, and helps improve communication by providing additional information.

5.3.2 IQ

IQ in XMPP stands for Info/Query which is useful for setting up connections, getting the status of the network/service, and troubleshooting.

```
<stream>
<presence>
    <value/>
</presence>
<message to=‘recepient’>
    <body/>
</message>
<iq to=‘server’>
    <query/>
</iq>
...
</stream>
```

Figure 2: XMPP stream

5.4 Messaging with XMPP

All the exchange of information in XMPP happens in XML streams. XMPP defines two types of XML communications : XML streams and XML stanzas. XMPP also allows definition of a gateway to other protocols which would perform the translation. This has allowed many messaging applications to offer support or partial support for XML. Clients connect to XMPP servers over TCP, here multiple resources (such as multiple devices) can connect with the same client id to the XMPP servers on behalf of the same client. This is a very important feature, considering the fact that it allows users to use messaging over different devices like laptops and phones at the same time.

Messaging involves asynchronous exchange of data with relatively small payloads. To work with this XMPP defines XML Streams and XML stanzas :

5.4.1 XML Streams

The XML stream starts with a `<stream>` tag and can send an unbounded number of XML elements between this tag and the `</stream>` tag, which denotes the end of the stream. A response stream can be set up immediately to send responses.

5.4.2 XML Stanzas

An XML stanza is the first child of the root `<stream>` and can contain several sub tags and attributes. It is used for the presence, iq, and other features offered by XMPP.

The structure of XMPP data exchange is usually something like Figure 2 [44]

XMPP allows encryption of streams using TLS or SASL and requires response stream to have a different key than the stream it responds to. However the server should ensure that the same scheme is supported by the receiving server.

If the receiver detects an error it may send with an `<error/>` response specifying the error, errors in XMPP are considered unrecoverable and the data needs to be retransmitted. XMPP by default uses UTF-8 making it easy to have a set encoding style which is useful globally. XMPP manages a roster of client's contacts and allows only a user-approved subset of the contacts to subscribe to presence data [42]. For security and privacy, XMPP also maintains privacy lists, allowing users to block other users or communications.

XMPP battles with forging addresses in its Server-to-Server authentication protocol which stamps ‘from’ and ‘to’ at every step of communication. However it is still possible to forge address in this system by using a malicious server or by attacking the DNS. XMPP also suffers from the address mimicking flaw that email suffers from, partially due to the use of UTF-8 [43].

5.5 Messengers

The 1990s and early 2000s soon became the golden age of desktop based instant messaging with dozens of competing messengers fighting for a share of the market.

5.5.1 AOL Instant Messenger

AIM was at one time the most widely used messaging app. AIM used the OSCAR (Open System for CommunicAtion in Realtime) which AOL developed. Despite its name, OSCAR was a proprietary protocol, and AOL went great lengths to keep the protocol from being used by competitors [8].

AIM ‘running man’ icon became one of the most recognized the symbol of early 2000s, however its popularity fell sharply with the rise of social networking sites and Google-talk. AOL tried porting AIM to XMPP, but the effort was abandoned later. AIM also supported sending small files, audio-videos and real time games.

5.5.2 ICQ

ICQ (read I seek you) was released as an open protocol for messaging in 1996. ICQ quickly gained popularity with their method of assigning Unique Identification Numbers (UINs) to users allowing them to share them for connections. Blackberry messenger and Snapchat would later follow this model.

5.5.3 Yahoo Messenger

Yahoo introduced its own messenger and accompanying protocol for instant messaging. The protocol would include presence, file sharing, gaming and ability to send stickers and have custom avatars.

5.5.4 Google Talk

Google introduced an implementation of XMPP known as Google-talk alongside their email. Google Talk had all the features included in XMPP such as presence and multi-device support and some extensions such as the ability to send images, and files. With the launch of Google+, Google later replaced this protocol with its own proprietary Hangouts protocol.

5.5.5 Skype

Skype introduced one of the first server-less Peer-to-Peer messaging and video-audio calling protocol. This protocol was very successful initially, as it ensured smooth communication even with poor network connectivity because there was no server connection involved, allowing optimum utilization of the bandwidth. However, after acquisition by Microsoft, the protocol was converted into a server based protocol because it would not work well on smartphones, where battery life and processing power is constrained [12].

5.5.6 MSN Messenger

MSN Messenger added several features to messaging such as social network integration, ability to send and view albums, offline messaging. It was one of the first messengers to be available for the mobile platforms. It was later discontinued when Microsoft acquired Skype.

6 The Smartphone Era

In the last few years, the processing power and connectivity of phones has improved exponentially. Better hardware and more capable software to accompany turned the cell phone into a smart phone - a powerful device with abilities comparable to a computer. Improvements such as 3G and 4G as well as fast WiFi networks have improved the connectivity to phones significantly. With more and more people using smart phones, instant messaging got a huge push with millions of people experiencing the Internet for the first time. This growth was particularly high in countries such as India and China, which had poor network connectivity and awareness before low cost smartphones flooded the markets.

It is no surprise therefore that Instant Messaging is the biggest it ever was with trillions of messages sent every year, hundreds of thousands of them every second.

6.1 Social Networking

Smartphones gave a helping hand to the already growing social networks and the most popular social network - Facebook has more than a billion unique users. Twitter has millions of users who use it everyday. These networks added instant messaging abilities from day 1, including ability to send one-to-one personal messages.

Facebook introduced the ability to chat from the first version of the website. This ability was soon bundled into a standalone product - Messenger for smartphones. Facebook Messenger uses the MQTT protocol : see 6.2. To improve user experience Facebook made several modifications to the messenger, including ability to send SMS and continue the conversation if user is not connected to the internet, and releasing an optimized, light version of the app for phones with low memory, processing power and for areas with poor connectivity [7].

Facebook then acquired Whatsapp see 6.6.1 to bolster its smartphone offerings and has built a platform around the messenger for people to develop on allowing chat-bots, games, transactions over IM [2].

6.2 Message Queue Telemetry Transport (MQTT)

MQTT is a lightweight protocol, specially designed for constrained environment which provides lossless bidirectional ordered communication of messages. MQTT is an open standard, defined as ISO/IEC 20922:2016 [5]. MQTT uses websockets and other web based protocols such as WebRTC for communication. It is optimized for small payloads and delivery constraints [26]. MQTT gives the application control of message delivery allowing 3 different mechanisms using different methods :

1. At most once - Messages are sent only once, and message loss may occur.

2. At least once - Messages can be resent and received multiple times, but all messages are delivered.

3. Exactly once - used for transactions.

MQTT focuses on simplicity with just 5 methods, and binary payloads with no message properties and a small header.

6.3 Advanced Message Queuing Protocol (AMQP)

AMQP offers many more features as compared to other protocols with a focus on business features - reliable, secure, peer to peer messaging. AMQP sets up a unidirectional link between two nodes allowing for a decentralized message passing. These nodes are grouped in containers, with brokers to link to client operations. AMQP sets up a duplex connection and sends an ordered set of frames. A frame is only accepted if all preceding frames have arrived successfully. AMQP uses TCP to ensure these guarantees. AMQP allows both synchronous and asynchronous communication, as well as an option to treat messages as database transactions ensuring atomicity.

Frames in AMQP have no size limit and can carry arbitrary amount of data. AMQP sets up sessions between two nodes which can be terminated by a negotiation mechanism built in the protocol. Each frame has a fixed size header, a variable header and content. Some nodes are defined as 'Distribution Nodes,' they store and forward the messages [49]. See figure 3.

The states of a message in AMQP are as follows:

1. Accepted : Message was valid and forwarded.

2. Rejected : Invalid Message.

3. Release : Message was not processed.

4. Modified : Message was modified but not accepted.

This state can be used to debug and find out what went wrong in each operation. AMQP allows transactional processing allowing at most once delivery.

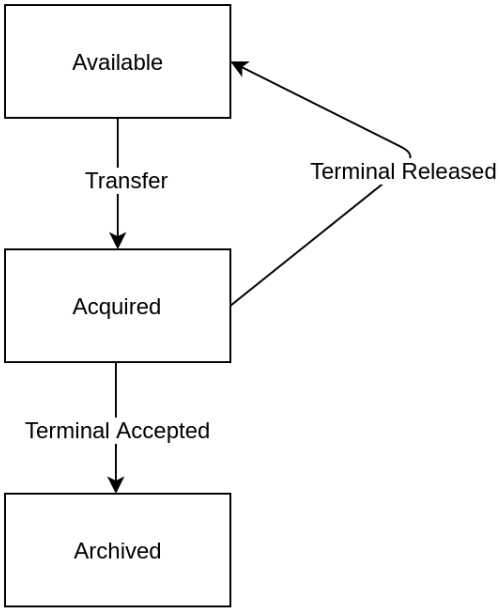

Figure 3: Routing in AMQP

AMQP places a lot of importance on security, with support for TLS and SASL built in. AMQP has several implementations such as Apache Qpid - an enterprise messaging system [40]. RabbitMQ offers a middleware to use multiple protocols for business messaging [48].

6.4 Streaming Text Oriented Message Protocol (STOMP)

STOMP is a messaging protocol that is similar to HTTP with commands working over TCP. STOMP allows a wire format which can be used by brokers written in any language and on any architecture. STOMP follows HTTP design with simple commands which make client implementation easy. Commands such as SEND, RECEIVE, CONNECT can be used with STOMP. STOMP uses heartbeats (periodic pings) to check the liveness of of the underlying TCP network. Stomp allows a set of limited commands

1. SEND

2. SUBSCRIBE

3. UNSUBSCRIBE

4. BEGIN

5. COMMIT

6. ABORT
7. ACK
8. NACK
9. DISCONNECT

Server sends 3 types of frames :

1. MESSAGE
2. RECEIPT
3. ERROR

A STOMP session can be completely described in Backus-Naur Form (BNF) like HTTP [27, 11].

6.5 MTProto

MTProto is a protocol developed and used by Telegram Messenger. The Client-Server protocol is open source, however the Server-Server protocol is proprietary. MTProto is designed around encryption and security with security being a core part of the protocol. MTProto allows the use of TCP, UDP, HTTP to transport messages. MTProto uses SHA-1 and AES encryption to allow complete end-to-end encryption [29].

6.6 Smartphone only Messengers

With the growth of Smartphones, messengers designed for smartphones first arrived with special features designed and optimized for the use with smartphones.

6.6.1 WhatsApp

The most successful of these has been WhatsApp, with over 1 billion users as of 2016 and 42 billion messages sent every day [16]. WhatsApp uses FunXMPP, a custom and proprietary modification of the XMPP protocol discussed in section 5.3. One of the reasons for the unprecedented success of WhatsApp has been the low barrier for entry : no need to have username and password registration, you can just start with a valid phone number. Over the years WhatsApp has added many features including audio-video calls and end-to-end encryption (see 7.1).

WhatsApp uses a highly optimized version of XMPP, which can be used in poor network environment, guaranteeing eventual (though ordering may differ) delivery of messages. This has led to its immense popularity in countries such as India with insufficient network infrastructure.

6.6.2 iMessage

Apple introduced iMessage with iPhone. iMessage seamlessly merges SMS and messages over Internet to create an illusion of unified inbox. This idea became very popular and was later used by Skype, Facebook Messenger and Google Hangouts.

6.6.3 WeChat

WeChat is a messaging app by Tenecent which is one of the largest messaging apps with over a billion users, a majority of them from China. WeChat provides integration with many different services, including ordering food, transfer of money, and calling a cab.

6.6.4 Slack

Slack built up on the channels idea by IRC and made a productivity and collaboration tool around it. Slack offers a broker for both IRC and XMPP. Slack soon gained in popularity and is one of the most used collaboration tools these days, all because of the ease of use.

7 Privacy and Security

With the growing fear of communication interception by governments, companies, and private entities there is an increasing demand for secure and encrypted communication. Large scale leaks of personal data and business communications have made security an extremely valuable feature for messengers. Several messengers came up with the explicit goal of encrypted point to point communication that was guaranteed to be private. Signal, Telegram and others came up with stronger encryption and authentication.

The following are the main techniques used by companies to ensure security,

7.1 End-to-End Encryption

End-t-End-Encryption promises to offer total security from any eavesdropping, including the carriers and app developers. E2EE works with a Pre-shared secret or a PGP key between two parties which is then used to derive a one time verification key to set up session [50]. This operation, completely done at both clients allows the clients to authenticate each other and ensure that messages that arrive are from verified source and secure. The most popular implementation of E2EE used by the Signal and WhatsApp messengers remains the one by OpenWhisperSystems – a not for profit security organization [38, 34].

E2EE is susceptible to various attacks despite its promises, such as Man-in-the-middle and backdoors. Access to the physical device can render E2EE ineffectual.

7.2 Self Destructing Messages

Popularized by Snapchat, self destructing messages have a timer set to them. When opened the message deletes itself after the given time, leaving no trace behind. This can be particularly useful to share secrets or for most messages where archival is unnecessary. However without an open platform to ensure this, current implementations are at the mercy of the companies which provide this service with no guarantee whether messages are actually deleted along with all replicas.

7.3 Access Control

With privacy in mind, most messengers provide some sort of access control needing approval from both the sender and receiver to set up communication. Messengers also allow blocking communication, limited access and a set of controls to control the access to safeguard privacy.

8 Extensions

Ever since the telegram, people have been building services which used the underlying message transfer to provide service and real-time information. Most of these services have been live sports scores, stock tickers, money transfer, value added services, subscription for news or updates, and a host of things that clients can do. Over the years, the services remain the same, just the underlying medium of exchange changes. The speed and reach improves and a few new services are added as extensions.

We will now focus on some of the applications that modern messaging has given rise to.

8.1 Chatbots

With the growth in Artificial Intelligence and Natural Language research, companies have utilized the powerful processing capabilities to create chat-bots - programs that can chat with humans to provide service. Through natural language, the bots can hold conversations with users to offer services such as ordering a pizza or hailing a cab without human intervention. This being available in natural language lowers the barrier for adoption and has attracted a lot of attention at companies like Facebook which have built a platform around their messenger to allow this [3].

Google recently released Allo a messenger created for the exact purpose of being your personal assistant. Google Home, Amazon Echo carry the same idea forward.

8.2 Audio/Video Calling

Most messengers these days integrate the ability to call people either as a voice call or a video call for free. Utilizing VoIP techniques they have made messengers a complete set of communication tools.

8.3 Live Video Broadcast

Live video broadcast is offered by Periscope (Twitter) and Facebook allowing instant broadcast of any video across the world. This has led to a revolution in news gathering and media, allowing instant reporting from scenes of crime and disasters.

8.4 Emojis, GIFs and Stickers

Each messenger offers its selection of stickers, animations and reactions using faces, movements and stickers.

8.5 Read Receipts

While delivery receipt service for sent messages existed from the time of Telegraph, modern apps can actually keep a track of whether the receiver has read the message. This allows better feedback for the sender and can give an assurance of the delivery of the message.

9 Future Work

This document tries to cover the topic of instant communication starting from history to modern interfaces. While it covers certain things in detail, it is not a technical document, nor it is a guide for developers to develop their own system.

Future work would involve implementing and analyzing the modern protocols for Instant Messaging and creating a guide which works on each use case and suggest what should the approach be. Modern messaging protocols offer complex techniques to maintain basic properties, however it being a best effort service, these techniques have not been analyzed critically. These protocols must be formally verified to show that they are fault tolerant and accurate.

Further, a lot of the techniques used in distributed systems and these protocols are very similar, their needs to be a survey of which techniques each community could borrow from another to improve reliability and availability of the systems.

9.1 Upcoming Technologies

Being an ever changing landscape, the work in this survey is in no way comprehensive, several new protocols are coming up and they need to be put through the same exercise to achieve completeness, a few upcoming protocols are discussed in brief :

9.1.1 Rich Communication Services

Rich Communication Services is an ongoing effort by the GSM association to add all the features offered by modern messaging systems such as sharing photos, location, presence, files directly into the phone services offered by the carriers. RCS promises to improve messaging, calling, and add several features which can only be used in third party apps directly into the bundled services. RCS, named 'Joyn' promises to create an open platform for theses services [10].

9.1.2 IRCv3

The decline of IRC has led to the IRC working group to work on IRCv3, an open protocol based on IRC that would allow all the modern messaging extensions with the simplicity and openness of the IRC protocol. The protocol is still in works, with new versions released this year, but it is not currently used by any application [6].

10 Conclusion

The nature of communications is cyclic, a new medium comes up, improving over the existing medium, and it simplifies access, makes it cheaper and better. Services are built around the medium, causing new companies to boom, until the next breakthrough is achieved and the process begins all over again. An entire industry was built around telegraph stock tickers which ended abruptly with the invention of the telephone, and often we see this pattern repeated over the ages, companies and products built over a certain medium do not last as the medium gets replaced with something better.

The study of instant messaging is a study of distributed, networked communication. The hope with each iteration that better communication would bring peace rarely holds, but at the same time it is true better communication has improved life of humanity and made the world a less lonely place.

The aim of this paper was to trace history of instant communication and present the modern techniques. There are several lessons to be learned from history and some key takeaways. It is also interesting to observe that the modern protocols are geared towards simplicity and low overhead communication rather than more features.

Acknowledgements

I would like to take this opportunity to thank Dr. Peter Alvaro, Michael Shavlovsky, and the entire CMPS 232 Fall 16 class, discussions with whom inspired the idea of this exploration. This document wouldn't have been possible without the infrastructure provided by UC Santa Cruz and UC Berkeley libraries.

References

[1] Cell phone usage statistics. https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/.

[2] Facebook messenger platform. https://developers.facebook.com/docs/messenger-platform.

[3] Facebook messenger platform. https://developers.facebook.com/docs/messenger-platform/.

[4] imelody format specification. http://merwin.bespin.org/t4a/specs/ems_imelody.txt.

[5] Information technology – message queuing telemetry transport (mqtt) v3.1.1.

http://www.iso.org/iso/catalogue_detail.htm?csnumber=69466.

[6] Ircv3. http://ircv3.net/irc/.

[7] Messenger lite. FB newsroom.

[8] Open system for communication in realtime - unofficial documentation. http://iserverd.khstu.ru/oscar/.

[9] Oxford 2015 word of the year. http://blog.oxforddictionaries.com/2015/11/word-of-the-year-2015-emoji/.

[10] Rich communication services. http://www.gsma.com/network2020/rcs-2/.

[11] Simple (or streaming) text oriented messaging protocol. https://stomp.github.io/stomp-specification-1.2.html.

[12] Skype moving to servers for scalability. https://www.listbox.com/member/archive/247/2013/06/sort/time_rev/page/1/entry/6:271/20130623090855:0B714E0A-DC06-11E2-9F35-8CD4CCA160A2/.

[13] Sms point to point specification, 3gpp ts 23.040. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=747.

[14] Unicode emoji list. http://unicode.org/emoji/charts/full-emoji-list.html.

[15] The unidcode universal coded character set. http://www.columbia.edu/kermit/ucs2.html.

[16] Whatsapp statistics. WhatsApp stats.

[17] The world in 2010. http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf.

[18] Akkoyunlu, E. A., Ekanadham, K., and Huber, R. V. Some constraints and tradeoffs in the design of network communications. *SIGOPS Oper. Syst. Rev. 9*, 5 (Nov. 1975), 67–74.

[19] Bhushan, A., Pogran, K., Tomlinson, R., and White, J. Standardizing network mail headers. Tech. rep., 1973.

[20] Bodic, G. L. *Mobile Messaging Technologies and Services: SMS, EMS and MMS*. John Wiley & Sons, 2003.

[21] Burns, R. W. *Communications: An International History of the Formative Years*. IET, 2004.

[22] Crocker, D. Standard for the format of arpa internet text messages.

[23] Crocker, D., Vittal, J., Pogran, K. T., and Henderson, D. A. Standard for the format of arpa network text messages. Tech. rep., 1977.

[24] Dierks, T. The transport layer security (tls) protocol version 1.2.

[25] Erlandson, C., and Ocklind, P. Wap—the wireless application protocol. In *Mobile Networking with WAP*. Springer, 2000, pp. 165–173.

[26] Fette, I. The websocket protocol.

[27] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. Hypertext transfer protocol–http/1.1. Tech. rep., 1999.

[28] Freed, N., and Borenstein, N. Multipurpose internet mail extensions (mime) part one: Format of internet message bodies. Tech. rep., 1996.

[29] Jakobsen, J., and Orlandi, C. *A practical cryptanalysis of the Telegram messaging protocol*. PhD thesis, Master Thesis, Aarhus University (Available on request), 2015.

[30] Kalt, C. Internet relay chat: Architecture. Tech. rep., 2000.

[31] Kalt, C. Internet relay chat: Channel management.

[32] Kalt, C. Internet relay chat: Client protocol.

[33] Kalt, C. Internet relay chat: Server protocol.

[34] Marlinspike, M., and Perrin, T. The x3dh key agreement protocol.

[35] Myer, T., and Henderson, D. Message transmission protocol. Tech. rep., 1975.

[36] Myers, J. Simple authentication and security layer (sasl). Tech. rep., 1997.

[37] Oikarinen, J., and Reed, D. Internet relay chat protocol.

[38] Perrin, T. The xeddsa and vxeddsa signature schemes.

[39] Pogran, K., Henderson, D., Crocker, D., and Vittal, J. Proposed official standard for the format of arpa network messages.

[40] Qpid, A. Open source amqp messaging. *URL: http://qpid. apache. org* (2013).

[41] Resnick, P. W. Internet message format.

[42] Saint-Andre, P. Extensible messaging and presence protocol (xmpp): Instant messaging and presence. Tech. rep., 2004.

[43] Saint-Andre, P. Extensible messaging and presence protocol (xmpp): Address format.

[44] Saint-Andre, P. Extensible messaging and presence protocol (xmpp): Core.

[45] Schulzrinne, H. Real time streaming protocol (rtsp).

[46] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. Real-time transport protocol. *RFC1899* (2003).

[47] Standage, T. *The Victorian Internet: The Remarkable Story of the Telegraph and the Nineteenth Century's On-line Pioneers*. Macmillan, 1998.

[48] Videla, A., and Williams, J. J. *RabbitMQ in action*. Manning, 2012.

[49] Vinoski, S. Advanced message queuing protocol. *IEEE Internet Computing 10*, 6 (2006), 87.

[50] Zeidler, H. M. End-to-end encryption system and method of operation, Mar. 25 1986. US Patent 4,578,530.