MSN Protocol Analysis

- Special thanks for Mr. Gin-Yuan Jai to prepare this material

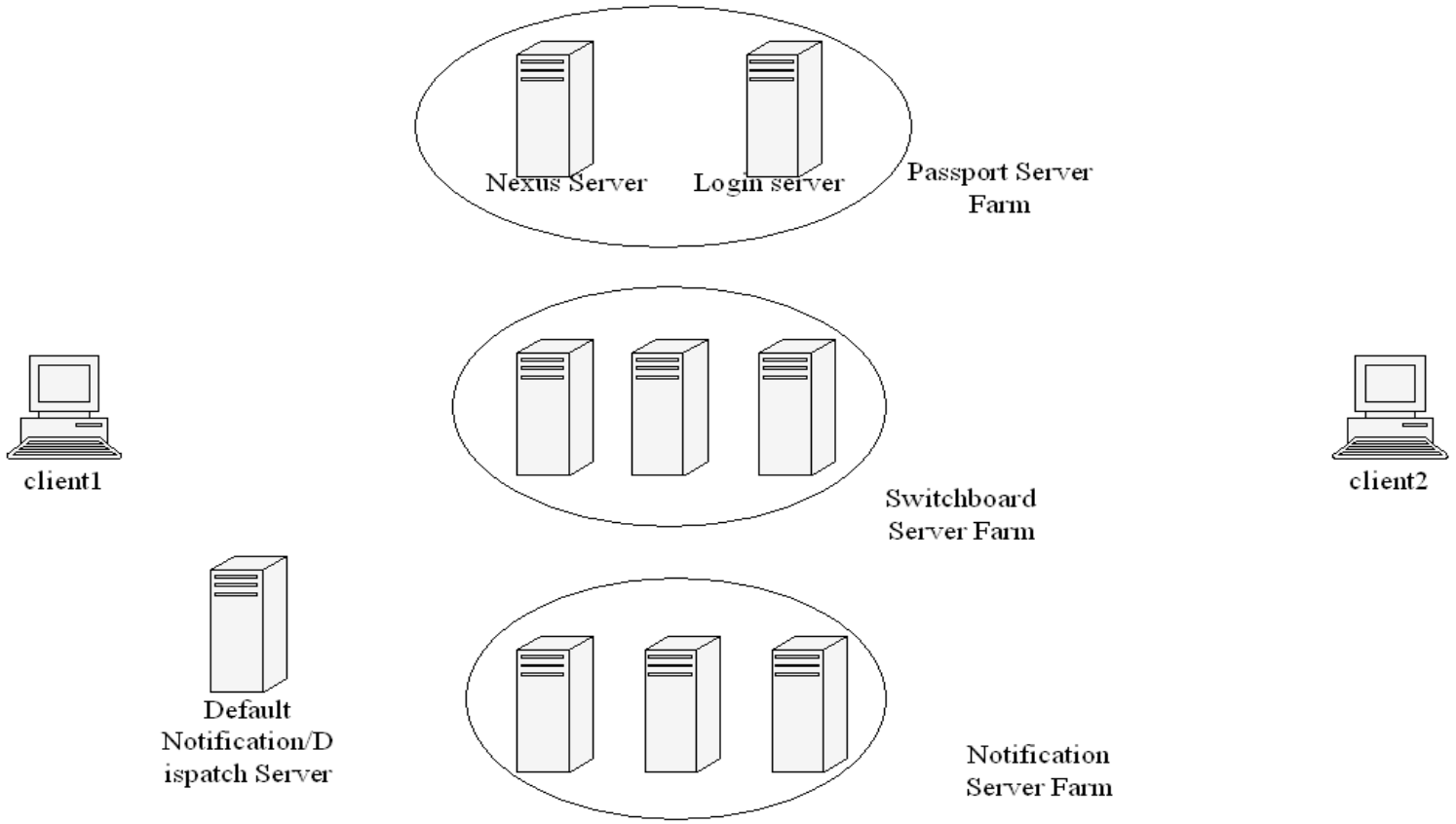
Outline

- MSN messenger network
- MSN protocol
- Communication & Command Features
- Login
- Switchboard session
- Message
- Invitation Message

MSN Messenger Network

- The MSN Messenger network is a *presence* and *instant* messaging network from Microsoft.
- The authentication/login stage is .NET Passport single sign-in service.
- An MSN Messenger session(from login to logout) involves a connection to a "notification server" (or "NS"), which provides a presence service.
- The notification server allows you to connect to "switchboard servers" ("SB"s) which provide an instant messaging service.

MSN Messenger Network(cont.)



Server Farm

- .NET Passport Server Farm
  - Nexus server (for retrieving the URL of a login server)
  - Login server (for authentication under HTTPS connection)
- Notification Server (NS)
  - Default notification server(Dispatch server)
  - Notification server
- Switchboard (SB)
  - For File/Chat sessions

Notification Server (NS)

- Notification server handles presence information about users and the principals whose presence they have subscribed to.
- Presence information, in its broadest sense, is information about the present state of the principal and/or the client.(status, display name)
- Notification server also performs some other services
  - New e-mail in hotmail inbox
  - To create new (or join existing) switchboard sessions
- Dispatch Server is just treated as the default notification server.

Switchboard (SB)

- Directly connected conversations between principals are not used in MSN Messenger,
- The switchboard handles instant messaging sessions between principals. The switchboard acts as a proxy between you and those you are chatting with.
- Each person in an MSN chat corresponds to a connection to a shared switchboard session.
- Being in two conversations at once means connecting to two switchboard servers at once.

Switchboard (SB) (cont.)

- The SB is also where invitations to other services such as ***file transfer*** and ***NetMeeting*** are sent and received
- Mobile paging is one of the only forms of communication that does not take place over a switchboard server.
- The SB and the NS are not very tightly integrated.
  - When a principal disconnects from the NS, all switchboard sessions still remain open until the client explicitly closes them.

MSN Client(official client)

- The User-end application setup at the user's PC for retrieving chatting service.
- MSN Messenger clients: MSN Messenger (also known as ".NET Messenger") and Windows Messenger.
- Recently released : MSN Messenger 7.0 (Mostly used is MSN Messenger 6.2, 6.1)
- Default build-in Windows XP : Windows Messenger 4.7

MSN Client/Server(third-party)

- Gaim - Support MSN, Yahoo, AOL, ICQ at Linux/Windows.

  http://gaim.sourceforge.net/
- Miranda IM

  http://www.miranda-im.org/
- MSN2GO -> Java Applet Client

  http://www.msn2go.com/

MSN Protocol Generation

- MSN Messenger protocol :The MSN Messenger protocol consists of a series of commands sent between the client and the server.
- MSNP8 : MSN messenger version 5.0
- MSNP9 : Introduced with MSN messenger 6.0

  (Most of the changes in MSNP9 exist solely to support MSNC1.)
- MSNP10 : Introduced with MSN messenger 6.1
- MSNP11 : introduced with MSN messenger 7.0

MSN Protocol Generation(cont.)

- MSN Client protocol :The MSN Client protocol consists of messages sent between clients.
- Grew quite organically - one version of the official client would behave differently to another
- MSNC0(MSNFTP)
- MSNC1(表情符號傳送)
- MSNC2

Communication Features

- All connections to MSN servers take place over *TCP/IP*
- Dispatch server
  - **messenger.hotmail.com(**207.46.104.20), port 1863 as the dispatch server for direct and SOCKS-based connections
  - **gateway.messenger.hotmail.com(207.46.110.249)**, port 80 as the dispatch server for HTTP connections.
- The connection between client and server must be considered asynchronous except for login stage.(I.e. not one request and then one response)
- .Net passport login stage is based on SSLv3 (HTTPS)

Communication Features(cont.)

- Text-based protocol with *UTF-8*, *URL-encoding*(to make sure a particular parameter does not contain any spaces, newlines, or otherwise invalid characters.), *XML-encoding*
- **messenger.hotmail.com** always sends **XFR(Transfer to another Notification Server)**, but **gateway.messenger.hotmail.com** never does. Microsoft's other notification servers very rarely send **XFR** - presumably, they send it when they are overloaded or going down for maintenance.
- **(from server 207.46.104.20)<<< XFR 36 NS 207.46.106.112:1863 0 207.46.104.20:1863\r\n**

Command Features

- The command code is a three all-caps letter.(except for Error Command)
- Most commands have at least one parameter.
- Some commands also have a Transaction ID (TrIDs) immediately after the command and before the parameters.
- Normal Command
  - Every normal command ends with a newline(\r\n)
  - All normal commands have a transaction ID (explained below) and end with a newline (also explained below).
  - Commands sent by the client will generally cause the server to respond with one or more commands.

Command Features(cont.)

- Payload Command : span over multiple lines
  - No "\r\n" end string
  - Special types of commands specify the length of the body in bytes
  - The payload commands in MSNP8 are **QRY**, **PAG**, **NOT** and (most importantly) **MSG**
  - **>>> (from client)QRY 1049 msmsgs@msnmsgr.com 32\r\n**
    8f2f5a91b72102cd28355e9fc9000d6e (no newline)
- Asynchronous commands
  - Commands sent by the server without the client explicitly requesting them,(e.g. **NLN**, **FLN**, and **BPR**)
  - Some commands do not contain TrIDs, other always contain TrID 0.(e.g. **ADD** and **REM**)

Command Features(cont.)

- Error commands have error numbers with TrID instead of letters in their command code.
  - Clients never send error codes to the server.
  - The server never sends error codes that aren't in response to any particular client command.
    - Ex. *try to ADD an invalid email address*

      (from client)>>> ADD 20 FL a@b a@b\r\n
      (from server)<<< 201 20\r\n
- The (client ping to server)PNG command and some SYN responses don't contain transaction IDs.
- In most cases, there will be just one response to each client command.

Command Order

- Replies to commands may come out of the order. TrID can be used to recognize the relation.
- There are no real rules on the order of unrelated commands.
- Some commands have multiple responses from the server. These responses contain the same TrID as the original command .

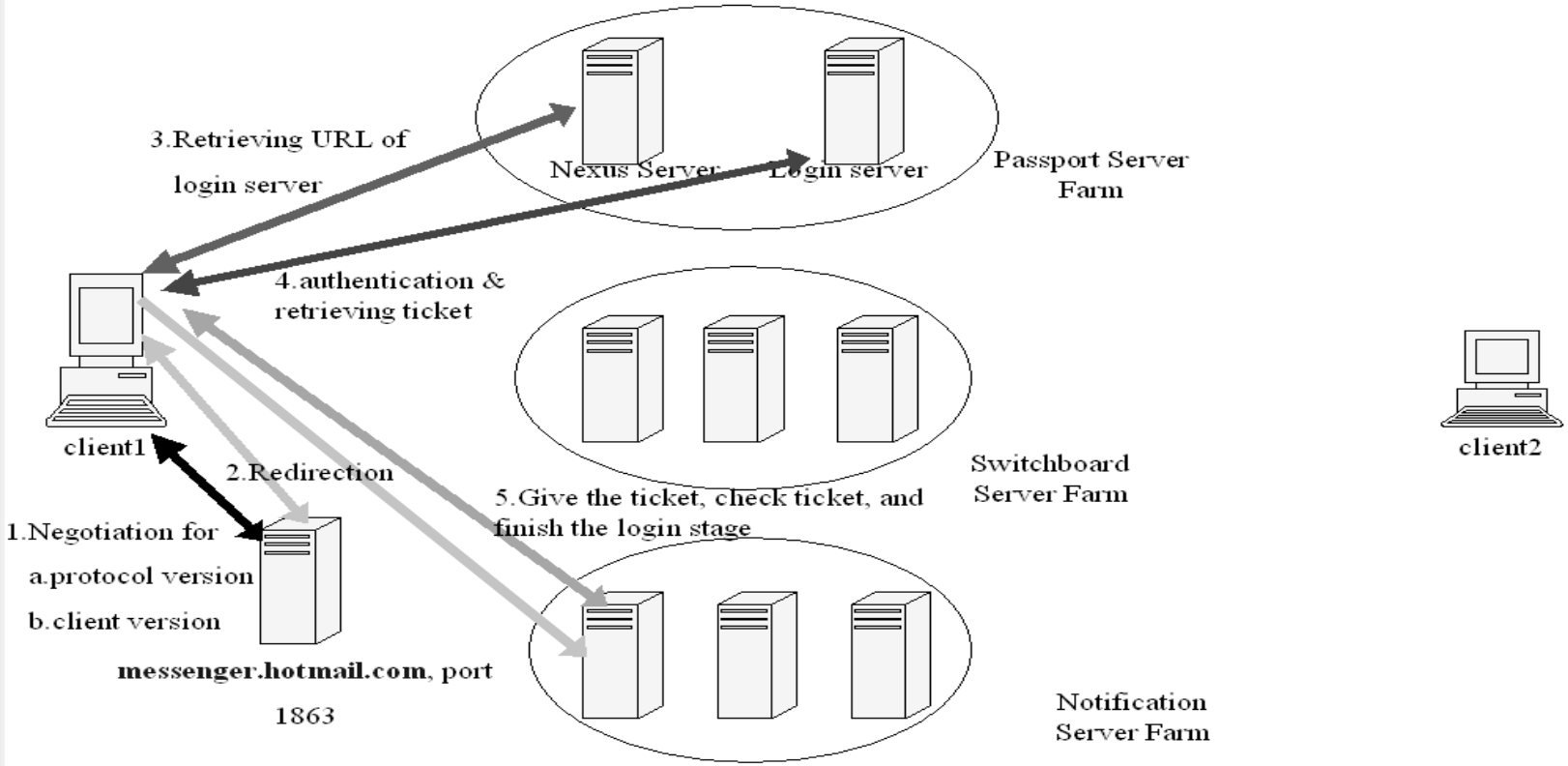
Transaction ID(TrID)

- TrIDs are used to match a client command with a server response.
- Commands sent by the client are required to contain a TrID, and commands from the server sent in direct response to that command will contain the same TrID
- The official client always increments the TrID by one after sending each command.
- Number between 0 and 4294967295 (2 to the power 32, minus 1).
- Some asynchronous commands use 0 as the TrID
- If your TrID is greater than 4294967295, the server will behave unpredictably.

Login stage

- When you first connect to a notification server, you are in the "login stage", which involves
  - Agreeing on a protocol version to use
  - Authenticating yourself to the MSN server,
  - Possibly being redirected to another notification server if the current one is overloaded.
- After Redirection to new NS, the client must to proceed the authentication at .NET Passport Login server to get the service entrance ticket.
- The ticket will be give to NS to finish the login stage(NS will check the validation of that ticket, if the ticket is invalid, the service will response error 911 and close the connection)

Login stage(cont.)



Switchboard Session

- Create a SB Session
  - Client sends XFR(with TrID)to NS for requesting a new switchboard session(XFR: transfer to SB)
  - The server will reply with another XFR (with the same TrID of Client's XFR command)with the following parameters:
    - Always SB
    - SB's IP & port
    - Type of authentication(always CKI)
    - Authentication string
  - Once connected to the switchboard, the client must send the USR command(with another TrID)
    - Account name
    - Authentication string

Switchboard Session(cont.)

- Create a SB Session(cont.)
  - If successful, the server will respond with a **USR** with the same TrID
    - **OK** as the first parameter
    - The user's account name
    - The user's display name

Switchboard Session(cont.)

- Create a SB Session(example)

(from client to NS)>>> XFR 31 SB\r\n

(from NS to client)<<< XFR 31 SB 207.46.108.39:1863 CKI 724871.1115613916.14547\r\n

(from client to SB) >>> USR 11 laplazshu@yahoo.com 724871.1115613916.14547\r\n

(from SB to client) <<< USR 11 OK laplazshu@yahoo.com laplazshu@yahoo.com\r\n

Switchboard Session(cont.)

- Invite others
  - To invite a principal into a switchboard session, send the CAL command (with TrID) to the switchboard.
    - Account name of the principal you wish to invite
    - The user's account name
    - The user's display name
  - If successful, the server will respond with another CAL with the same TrID
    - String RINGING
    - Session ID of this switchboard session (part of authentication string??)

Switchboard Session(cont.)

- Invite others(cont.)
  - If the CAL was successful, the specified principal will receive the RNG command from the NS.
  - When the principal join the session, every principal in the session (excluding the principal that just joined) will be sent the JOI command(no TrID).
    - Account name of the principal that just entered the session
    - URL-encoded display name of the principal that just entered the session

Switchboard Session(cont.)

- Invite others(example)

>>>(from client to SB) CAL 15 billgile@pchome.com.tw\r\n

<<< (from SB to client)CAL 15 RINGING 615268\r\n

<<< (from SB to every participants)JOI billgile@pchome.com.tw [billgile]\345\256\205\345\260\217\345\234\223(how%20ashame%20I%20am)\r\n

Switchboard Session(cont.)

- Invited by other principals
  - Receive an **RNG** command(no TrID) from the notification server
    - Unique Session ID
    - Address and port (has always been 1863) of the switchboard
    - Type of authentication—always CKI
    - Authentication string
    - Account name & URL-encoding display name
  - Client then connect to SB, and send an **ANS** command(with TrID)with the following parameters
    - Account name
    - Authentication string
    - Switchboard session ID

Switchboard Session(cont.)

- Invited by other principals(cont.)
  - If successful, the SB will respond with one or more **IRO**s(with the same TrID of ANS).
    - (Sequence)Number of the current **IRO** command in the list.
    - Total number of **IRO** commands that will be sent.
    - Participant's account name
    - Display name for the participant
  - The SB will send an **ANS** OK response(with the same TrID of ANS) to client.

Switchboard Session(cont.)

- Invited by other principals(example)

(from NS)<<< RNG 16781256 207.46.108.53:1863 CKI 1115614755.22305 billgile@pchome.com.tw [billgile]\345\256\205\345\260\217\345\234\223(how%20ashame%20I%20am)\r\n

<o> Client Connects to 207.46.108.53:1863 (Switchboard)

>>> ANS 13 laplazshu@yahoo.com 1115614755.22305 16781256\r\n

<<< IRO 13 1 2 billgile@pchome.com.tw [billgile]\345\256\205\345\260\217\345\234\223(how%20ashame%20I%20am)\r\n

<<< IRO 13 2 2 myname@msn.com My%20Name\r\n

<<< ANS 13 OK\r\n

- **<o> Continue SB Session . . .**

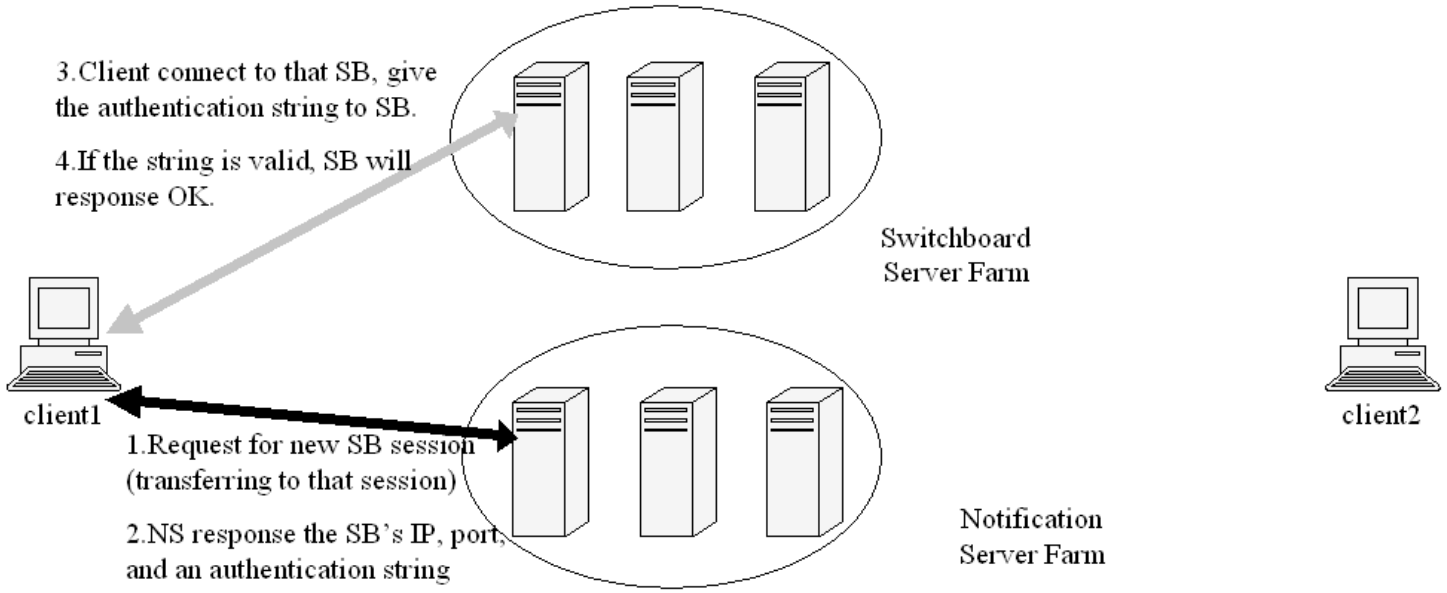
Switchboard Session(cont.)

- Leave the session
  - To leave a switchboard session, a client should send the OUT command with no TrID and no parameters.
  - All other principals in the session will receive the BYE command with no TrID and the account name of the principal that left as the first parameter.

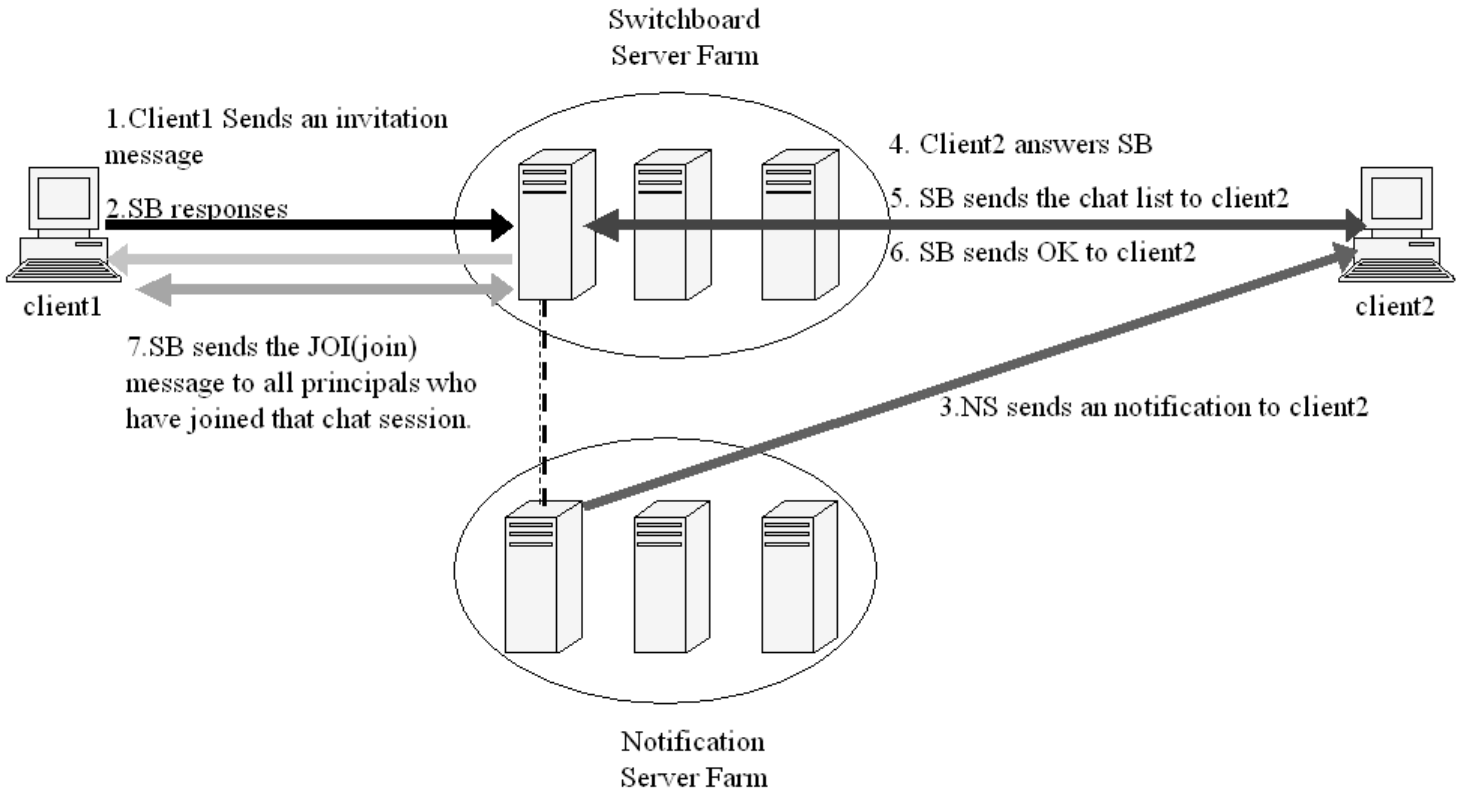
>>> OUT\r\n

<<< BYE laplazshu@yahoo.com\r\n

Create a SB session



Invite Others



Messages(MSG Commands)

- From server to client :
  - They always have no TrID
  - Two parameters(account name and URL-encoded display name) before the payload length parameter.
    - NS : they are always both Hotmail
    - SB : they are account name and display name of the source of the message
- From client to server(Only allowed in SB session) :
  - They must have a TrID ,
  - Acknowledgment type specifying the acknowledgment you wish to receive from the server
  - Length of the payload in bytes

Messages(cont.)

- Payload
  - Server does not interpret the payload
  - The payloads of messages are expected to be formatted in a particular way(MIME type)
  - Standard MIME header, unordered fields(Valid header must contain at least **Content-Type** field and **MIME-Version** field)
  - Body encoding depends on the information of header.
- Three standard types of messages (differentiated by a unique **Content-Type** field)
  - **Instant Messages (text/plain)**
  - **Typing Notifications (text/x-msmsgscontrol)**
  - **Application Invitations (text/x-msmsgsinvite)**

Acknowledgement

- Acknowledgement setting in client's MSG command
  - U : no acknowledgement
  - N : acknowledging only if the message was not properly received
  - A : acknowledging whether it was properly received or not. (ACK /NAK)
- Acknowledgement from the switchboard to Client :
  - ACK (successfully received by every participant)
  - NAK (one or more participants did not successfully receive your message)
  - TrID corresponding with the outgoing MSG

Instant Messages

- Content-Type: text/plain (for ISO 8895-1)or Content-Type: text/plain; charset=UTF-8,
- Acknowledgement type N

MSG 13 N 165\r\n

MIME-Version: 1.0\r\n

Content-Type: text/plain; charset=UTF-8\r\n

X-MMS-IM-Format:
FN=%E6%96%B0%E7%B4%B0%E6%98%8E%E9%AB%94;
EF=; CO=0; CS=88; PF=0\r\n

\r\n

nice to see you ^_^

Typing Notifications

- To periodically inform the other participants in a switchboard session that you are currently typing a message.
- **Content-Type: text/x-msmsgscontrol**
- Acknowledgement type **U**
- **TypingUser** field in header(account name)
- Body consists only of a single newline (\r\n).

MSG 15 U 92\r\n
MIME-Version: 1.0\r\n
Content-Type: text/x-msmsgscontrol\r\n
TypingUser: laplazshu@yahoo.com\r\n
\r\n
\r\n

X-MMS-IM-Format Field

- Font Name (FN): the font name must be URL-encoded. e.g. "MS Sans Serif", you would have to specify FN=MS%20Sans%20Serif.
- Effects (EF): optional style effects (bold, italic, underline, and strikethrough). e.g. EF=IB or EF=BI mean bold-italic text
- Color (CO): font color.
  - A six-character hexadecimal BGR (blue-green-red, the reverse of the standard RGB order seen in HTML) string.
  - The official client also cuts off all leading zeros in outgoing messages. E.g. 0000ff = ff

X-MMS-IM-Format Field(cont.)

- Character Set (CS): one or two hexadecimal digits e.g. 0 - ANSI_CHARSET
- Pitch and Family (PF): the PF family defines the category that the font specified in the **FN** parameter falls into.
  - The first digit of the value represents the font family. E.g. 1_ - **FF_ROMAN**
  - The second digit represents the pitch of the font - in other words, whether it is monospace or variable-width. E.g. _2 - **VARIABLE_PITCH**
- Right alignment (RL): whether a message should be right-aligned or not.(only 1 if the message is right-aligned, other values will be omitted)

Application Invitations

- To invite principals to join applications
- Voice conversation, video conferencing, NetMeeting, remote assistance, whiteboard, games, and more.
- Each invitation message is sent through a switchboard session as a MSG command
- **Content-Type: text/x-msmsgsinvite; charset=UTF-8**
- Acknowledgement type **N**
- The Negotiation behavior of File Transfer is different with other application

Application Invitations(cont.)

- Invitation and Negotiation(Except for File Transfer)
  - The inviter sends an INVITE command
  - the invitee ACCEPTs the invitation
  - The inviter opens a socket to listen for a connection, then ACCEPTs.
  - The inviter then sends some Context. (only in some invitation-types)

Command Fields

- Common Command Fields
  - Invitation-Command: The type of message being sent.
    - INVITE
    - ACCEPT
    - Context
    - CANCEL
  - Invitation-Cookie: An integer between 1 and 2^32 - 1, uniquely identifying a negotiation.
  - Session-ID : A unique identifier for the current running MSN Messenger client.

Command Fields(cont.)

- Invitation Command Fields
  - Application-Name: A natural-language description of the class.
  - Application-GUID: The unique identifier of the class.
  - Session-Protocol:Session protocol(s) supported by sending the client. The official client only supports SM1.
  - Application-URL *(optional)*:A URL with information about downloading the specified application.
  - Context-Data *(optional)*:A string used for an application-specific purpose (e.g. negotiating details of the protocol to use)

Command Fields(cont.)

- CANCEL Command Fields
  - FAIL : the receiving client cannot parse an invitation message you sent it
  - OUTBANDCANCEL : the switchboard window in which the INVITE message was sent is closing.
  - REJECT : The principal has declined the invitation
  - REJECT_NOT_INSTALLED : the client does not support that application GUID.
  - TIMEOUT : the client has got bored of waiting for your reply (or the principal has cancelled the invitation).

Command Fields(cont.)

- First ACCEPT Fields
  - Session-Protocol : Selected session protocol (taken from the list given in the INVITE message). If there are no acceptable protocols, the invitation is cancelled with a cancel-code of FAIL.
  - Context-Data *(optional)* : A string used for an application-specific purpose (e.g. negotiating details of the protocol to use)
  - Launch-Application : Instructs the other client (not) to launch an external application. Normally "TRUE". I don't know under what condition you would set this to "FALSE".

Command Fields(cont.)

- First ACCEPT Fields(cont.)
  - Request-Data : The value of this field must be "IP-Address:". Requests that the other client send its IP address.
  - IP-Address :The IP address (and optionally port number) of the computer which will act as client.
- Second ACCEPT Fields
  - Launch-Application : Instructs the other client (not) to launch an external application. Normally "TRUE". I don't know under what condition you would set this to "FALSE".
  - IP-Address : The IP address (and optionally port number) of the computer which will act as server.

Command Fields(cont.)

- Context Fields
  - Context-Data : A string used for an application-specific purpose (e.g. negotiating details of the protocol to use)

Voice Invitation

❖ From client 1 to SB

MSG 37 S 400\r\n

MIME-Version: 1.0\r\n

Content-Type: text/x-msmsgsinvite; charset=UTF-8\r\n

\r\n

Application-Name: \350\252\236\351\237\263\344\272\244\350\253\207\r\n

Application-GUID: {02D3C01F-BF30-4825-A83A-DE7AF41648AA}\r\n

Session-Protocol: SM1\r\n

Context-Data: Requested:SIP_A,;Capabilities:SIP_A,SIP_V,;\r\n

Invitation-Command: INVITE\r\n

Invitation-Cookie: 24993528\r\n

Session-ID: {5386E9F8-FF66-4C9B-A131-E885A1F83BC6}\r\n

Conn-Type: Direct-Connect\r\n

Sip-Capability: 1\r\n

\r\n

Voice Invitation (cont.)

- From SB to client1

```
MSG billgile@pchome.com.tw
    [billgile]\345\256\205\345\260\217\345\234\223(how%20ashame%20I%20am)
    (\345\200\222\346\225\270..)(\351\226\211\351\227\234\345\201\232\344\272\2
    13..) 404\r\n
  MIME-Version: 1.0\r\n
  Content-Type: text/x-msmsgsinvite; charset=UTF-8\r\n
  \r\n
  Invitation-Command: ACCEPT\r\n
  Context-Data: Requested:SIP_A,;\r\n
  Invitation-Cookie: 24993528\r\n
  Session-ID: {B9C721B1-6677-4FCE-837D-0287F3A6B990}\r\n
  Session-Protocol: SM1\r\n
  Conn-Type: Direct-Connect\r\n
  Sip-Capability: 1\r\n
  Launch-Application: TRUE\r\n
  Request-Data: IP-Address:\r\n
  IP-Address: 218.169.74.71\r\n
  IP-Address-Enc64: MjE4LjE2OS43NC43MQ==\r\n
  \r\n
```

Voice Invitation(cont.)

- From client1 to SB

```
MSG 38 A 335\r\n
 MIME-Version: 1.0\r\n
 Content-Type: text/x-msmsgsinvite; charset=UTF-8\r\n
 \r\n
 Invitation-Command: ACCEPT\r\n
 Invitation-Cookie: 24993528\r\n
 Session-ID: {5386E9F8-FF66-4C9B-A131-E885A1F83BC6}\r\n
 Conn-Type: Direct-Connect\r\n
 Sip-Capability: 1\r\n
 Launch-Application: TRUE\r\n
 IP-Address: 218.169.75.55:16713\r\n
 IP-Address-Enc64: MjE4LjE2OS43NS41NToxNjcxMw==\r\n
 \r\n
```

File Transfer

- P2P Message--With MSN Messenger 6 a new message type is introduced, the **"application/x-msnmsgrp2p"**.
- MSNSLP--with MSN Messenger 6, a new sort of protocol is introduced and is based on the SIP (Session Initiation Protocol, RFC 2543) Protocol.
- MSNSLP is pretty much the same as SIP, it only supports less request methods.
- MSNSLP uses only the "INVITE" and the "BYE" method, the ACK's are provided by the server.
- The file name will be encoded with base64
- The direct-connection negotiation will include IP, port.

File Transfer(cont.)

- P2P Messages
  - Header: The 48-byte binary header consists of 6 DWORDs and 3 QWORDS, which are all in Big Endian order
    - First field is a DWORD and is the SessionID
    - Second field is a DWORD and is the Identifier which identifies the message,
    - Third field is a QWORD and is the Offset which is only being used if the data which is being sent is larger then 1202 bytes.
    - The fourth field is a QDWORD and represents the **total size** of the data which is being sended,
    - The fifth field is a DWORD and is the length of the data which is being transferred in **this** message.
    - The sixth field is a DWORD and is the Flag field, 0x20 when the data is for User Display Images or Emoticons, 0x01000030 if it's the data of a file.

File Transfer(cont.)

- P2P Messages(cont.)
  - Optional Data: MSNSLP message
  - Footer: The 4-byte binary footer consists of 1 DWORD which is in Little Endian order and that field represents the ApplicationIdentifier (AppID).
    - 0x1 for User Display Images and Emoticons,
    - 0x2 for File Transfers,

File Transfer(cont.)

- Step by step negotiation:
  - Invitation Message: From Sending Client (SC)to the Receiving Client (RC)
  - BaseIdentifier Message: RC to SC and SC to RC
  - 200 OK /Error Message: RC to SC
  - 200 OK Acknowledged Message: SC to RC
  - Direct Connection Invitation Message: SC to RC
  - Direct Connection Invitation Acknowledged Message:RC to SC
  - Direct Connection 200 OK/Error Message: RC to SC

File Transfer(cont.)

- Step by step negotiation: (cont.)
  - Direct Connection Handshake: SC to RC
  - Direct Connection Handshake Reply Message: RC to SC
  - File Data Message(s)/Data Acknowledged Message: SC→RC/RC→SC
  - Bye Message: RC to SC
  - Bye Acknowledged Message: SC to RC

File Transfer(cont.)

- Client1 send the invitation for file transfer(tmp.txt on desktop)

MSG 18 D 1349\r\n
MIME-Version: 1.0\r\n
Content-Type: application/x-msnmsgrp2p\r\n
P2P-Dest: billgile@pchome.com.tw\r\n
\r\n
\000\000\000\000Akw\000\000\000\000\000\000\000\000\
000\267\004\000\000\000\000\000\000\262\004\000\000\00
0\000\000\000\006\253\327\001\000\000\000\000\000\000\
000\000\000\000\000\000INVITE
MSNMSGR:billgile@pchome.com.tw MSNSLP/1.0\r\n
To: <msnmsgr:billgile@pchome.com.tw>\r\n
From: <msnmsgr:laplazshu@yahoo.com>\r\n
Via: MSNSLP/1.0/TLP ;branch={250EDFDE-03F6-4036-BC77-
4BBF86F02C86}\r\n

File Transfer(cont.)

- Client1 send the invitation for file transfer(tmp.txt on desktop)(cont.)

CSeq: 0 \r\n
Call-ID: {A2800F01-88C7-4218-A89B-1415B871F1B9}\r\n
Max-Forwards: 0\r\n
Content-Type: application/x-msnmsgr-sessionreqbody\r\n
Content-Length: 863\r\n
\r\n
EUF-GUID: {5D3E02AB-6190-11D3-BBBB-00C04F795683}\r\n
SessionID: 30911209\r\n
AppID: 2\r\n
Context:
PgIAAAIAAABWBwAAAAAAAAEAAAB0AG0AcAAuAHQAeAB0AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA

File Transfer(cont.)

- Client2 response to Client 1

MSG billgile@pchome.com.tw
[billgile]\345\256\205\345\260\217\345\234\223(how%20ashame%20I%20am)(\345\200\222\346\225\270..)(\351\226\211\351\227\234\345\201\232\344\272\213..) 480\r\n
MIME-Version: 1.0\r\n
Content-Type: application/x-msnmsgrp2p\r\n
P2P-Dest: laplazshu@yahoo.com\r\n
\r\n

\000\000\000\000\210\322=\b\000\000\000\000\000\000\000\000P\001\000\000\000\000\000\000P\001\000\000\000\000\000\0006\035\241\t\000\000\000\000\000\000\000\000\000\000\000\000MSNSLP/1.0 200 OK\r\n
To: <msnmsgr:laplazshu@yahoo.com>\r\n
From: <msnmsgr:billgile@pchome.com.tw>\r\n

File Transfer(cont.)

- Client2 response to Client 1(cont.)

Via: MSNSLP/1.0/TLP ;branch={250EDFDE-03F6-4036-BC77-4BBF86F02C86}\r\n

CSeq: 1 \r\n

Call-ID: {A2800F01-88C7-4218-A89B-1415B871F1B9}\r\n

Max-Forwards: 0\r\n

Content-Type: application/x-msnmsgr-sessionreqbody\r\n

Content-Length: 24\r\n

\r\n

SessionID: 30911209\r\n

\r\n

\000\000\000\000\000

File Transfer(cont.)

- Client 1 request for direct connection to Client 2

MSG 21 D 580\r\n
MIME-Version: 1.0\r\n
Content-Type: application/x-msnmsgrp2p\r\n
P2P-Dest: billgile@pchome.com.tw\r\n
\r\n

\000\000\000\000Bkw\000\000\000\000\000\000\000\000\000\261
\001\000\000\000\000\000\000\261\001\000\000\000\000\000\000
\222\022\330\001\000\000\000\000\000\000\000\000\000\000\000
\000INVITE MSNMSGR:billgile@pchome.com.tw MSNSLP/1.0\r\n
To: <msnmsgr:billgile@pchome.com.tw>\r\n
From: <msnmsgr:laplazshu@yahoo.com>\r\n
Via: MSNSLP/1.0/TLP ;branch={54B595BF-3450-4D61-8E63-B4213349EAE7}\r\n
CSeq: 0 \r\n
Call-ID: {A2800F01-88C7-4218-A89B-1415B871F1B9}\r\n

File Transfer(cont.)

- Client 1 request for direct connection to Client 2(cont.)

Max-Forwards: 0\r\n
Content-Type: application/x-msnmsgr-transreqbody\r\n
Content-Length: 92\r\n
\r\n
Bridges: TRUDPv1 TCPv1\r\n
NetID: 0\r\n
Conn-Type: Direct-Connect\r\n
UPnPNat: false\r\n
ICF: false\r\n
\r\n
\000\000\000\000\000

File Transfer(cont.)

- Client 2 response to Client 1

MSG billgile@pchome.com.tw
[billgile]\345\256\205\345\260\217\345\234\223(how%20ashame%20I%20am)(\345\200\222\346\225\270..)(\351\226\211\351\227\234\345\201\232\344\272\213..) 692\r\n
MIME-Version: 1.0\r\n
Content-Type: application/x-msnmsgrp2p\r\n
P2P-Dest: laplazshu@yahoo.com\r\n
\r\n

\000\000\000\000\211\322=\b\000\000\000\000\000\000\000\000$\002\000\000\000\000\000\000$\002\000\000\000\000\000\000\000371!\241\t\000\000\000\000\000\000\000\000\000\000\000\000MSNSLP/1.0 200 OK\r\n
To: <msnmsgr:laplazshu@yahoo.com>\r\n
From: <msnmsgr:billgile@pchome.com.tw>\r\n

File Transfer(cont.)

- Client 2 response to Client 1(cont.)

Via: MSNSLP/1.0/TLP ;branch={54B595BF-3450-4D61-8E63-B4213349EAE7}\r\n
CSeq: 1 \r\n
Call-ID: {A2800F01-88C7-4218-A89B-1415B871F1B9}\r\n
Max-Forwards: 0\r\n
Content-Type: application/x-msnmsgr-transrespbody\r\n
Content-Length: 236\r\n
\r\n
Bridge: TCPv1\r\n
Listening: true\r\n
Nonce: {70531C96-FA19-4BA8-9E58-DC14531B7BB5}\r\n
IPv4Internal-Addrs: 192.168.47.1 192.168.50.1 192.168.0.4 218.169.74.71\r\n
IPv4Internal-Port: 3150\r\n
IPv6-Addrs: 2002:daa9:4a47::daa9:4a47\r\n
IPv6-Port: 1025\r\n
\r\n
\000\000\000\000\000

Port Alteration

- The port setting could happen at some possible stage
  - XFR commands(NS Transfer, Create SB sessions)
  - Accept stage of Application invitation
- The NS/SB server port is all 1863 (except port 80 of Dispatch server)
- The port number is not fixed only for application invitation.

References

- MSN Messenger Protocol
  http://www.hypothetic.org/docs/msn/index.php
- MSN Messenger Protocol version 9
  http://wisoftware.host.sk/msn6/msnp9/
- MSNP11

http://msnpiki.msnfanatic.com/index.php/Main_Page