

GLoboH

XMPP Proxy



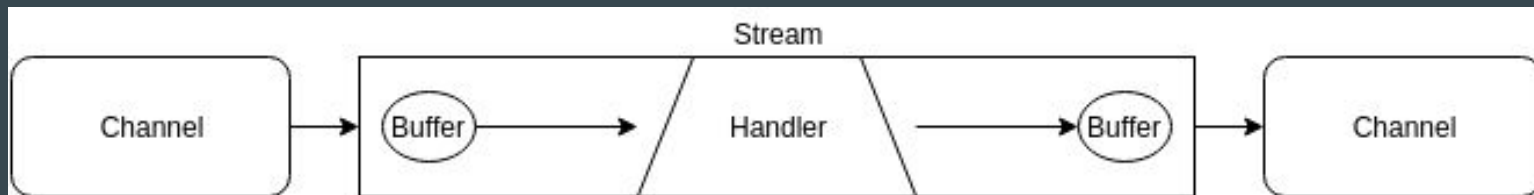
Goffan - Lobo - Horvat

55431 - 51171 - 55564

Grupo II

Planteo de la consigna - I

- Protocolos en capas:
 - TCP (XML (XMPP / GLoboH))
- Programacion contra interfaces
 - Nos permitio abstraernos facilmente y poder agregar funcionalidades.
- NIO
 - Dar los Terminals a Handlers, que en conjunto con NIO, nos permite manejar eventos concurrentes y multiplexación de conexiones.
 - Requiere una estructura compleja
- TCP Channels, Streams y Buffers



Planteo de la consigna - II

- TCPConversation: maneja el flujo de datos entre los terminales.
 - Hay una versión para protocolos en que el proxy es el intermediario, como XMPP y otra que el proxy va a procesar, como GLoboH.
 - Contiene al/ a los stream/s correspondiente/s.
- TCPProtocolHandler: maneja las opciones típicas: Accept, Connect, Read y Write.
 - Contiene a la TCPConversation correspondiente.
 - Al igual que la Conversation hay una versión para protocolos proxy-ados y otra para procesados.

Protocolo de administración - I

Saludo inicial y login

C: <globoh/>

S: <hello/>

C: <username>root</username>

S: <ok/>

C: <password>root</password>

S: <ok/>

Protocolo de administración - II

Comandos

<l33t> on/off </l33t>

<statistics> readbytes/writtenbytes/connections/readMessages/writtenMessages
</statistics>

<mute set="true/false"> username </mute>

<server_multiplex username="username"> address </server_multiplex>

<goodbye/>

Protocolos de administración - III

Los tipos de respuesta son:

Exitoso: <ok/>

Error: <error> <code>error code</code>

<message>error message</message> </error>

Métricas: <stats> <option>option</option>

<description>description</description>

<value>value</value> </stats>

Protocolos de administración - IV

400 BAD_XML_FORMAT

401 UNEXPECTED_COMMAND

402 BAD_COMMAND_FORMAT

403 BAD_COMMAND_OPTION

404 UNRECOGNIZED_COMMAND

405 WRONG_CREDENTIALS

911 UNKNOWN_ERROR

Librerías externas

- Parseo no bloqueante de XML: FasterXML - AaltoXML
 - <https://github.com/FasterXML/aalto-xml>
 - Apache License 2.0
- Validaciones, base64 encoding, y varias utilidades - Apache commons
 - <https://github.com/apache/commons-lang>
- Logging - SLF4J
 - <http://slf4j.org/docs.html>

Formato I33t

- Está funcional en TCP pero no en XMPP
- Falta poder codificar en utf-8 "<" para el "<"

Registros

- Se pueden ver:
 - Accesos
 - Requests que recibe el servidor
 - Responses que da el servidor
 - Bytes transferidos (recibidos y enviados - sólo para imágenes/archivos)
- `<statistics>CODIGO_DE_REGISTRO</statistics>`
- ¿Dónde se almacenan?
 - Se pueden ver en la consola.
 - Si el servidor se reinicia, se pierden los logs.
 - Posiblemente terminemos guardando en un archivo.
- Idem con las métricas.
- Soporta monitoreo remoto
- Son generales (no filtramos métricas).

Stream Initiation File Transfer

- Sin implementar por el momento.
- Divide el archivo en partes (de 8k por ej)
- La idea es que las partes no superen nuestro buffer interno.
- XEP-0095: Stream Initiation
- Negociación del stream:
 - Sender discovers if Receiver implements the desired profile.
 - Sender offers a stream initiation.
 - Receiver accepts stream initiation.
 - Sender and receiver prepare for using negotiated profile and stream, EUC
- Condición de errores:
 - The Receiver does not support the desired profile, EUC
 - Receiver rejects the stream initiation, EUC

Silenciar usuarios

- Idea: El usuario no puede enviar ni recibir mensajes.
- Nuestro protocolo lo que hace es setear una blacklist de usuarios por username.
- `Set<String> muteJIDs`

Dificultades

- Estructura compleja: mucha facilidad al momento de agregar funcionalidad pero ante el planteo inicial o cambios se pierde fácilmente la perspectiva del problema.
- Dificultades al momento de hacer tests por la complejidad del problema. Pudimos instalar un servidor XMPP en AWS para poder hacer las pruebas.
 - Va a ser testeado localmente también... ;)
- Dificultad al momento de encontrar causa de errores.

Posibles extensiones

Del proxy:

- Multithread.
- Manejo de protocolos de encriptación.
- Manejo de múltiples extensiones de XMPP.

Del protocolo GLoboH:

- Encriptación de datos.
- Mejores métricas.
- Mayor manejo de las propiedades de XML.