

Protocolo grupal de comunicación PIRO 2023

Esquema de comunicación

- Los servidores intermedios y de piezas deben estar en la misma red
- Los Servidores Intermedios y de Piezas usarán el protocolo HTTP
- Los Servidores Intermedios no se comunican con otros Servidores Intermedios
- Los servidores intermedios deben conectarse con los servidores de piezas y escuchar por solicitudes de clientes
- Los servidores intermedios contienen un mapa de ruta, el cual indica los servidores de piezas que contienen las piezas necesarias para construir cierta figura
- Los servidores de piezas indican a los servidores intermedios las figuras que contienen
- Los clientes establecen una conexión con los servidores intermedios, nunca con los servidores de piezas

URI

Los servidores intermedios redirigen las solicitudes de los clientes a los servidores de piezas, de modo que las URIs utilizadas deben ser estandarizadas, por lo cual se ha acordado utilizar la forma “/piece”, donde “piece” especifica la figura que desea el cliente. Por ejemplo si se desea la figura de Jirafa, la solicitud será de la siguiente manera:

GET /girafeo

Puertos

Se utilizarán 3 diferentes puertos para la comunicación en la red. Los servidores intermedios deben comunicarse con los clientes, así como con los servidores intermedios, para lo cual se ha acordado lo siguiente:

- En el puerto 4850, el Servidor Intermedio va a enviar un GET de broadcast, y los Servidores de Piezas van a mandar un 200 OK con su tabla de figuras si se encuentra levantado.
- En el puerto 4849 el Servidor Intermedio va a redirigir la request del cliente a los Servidores de Piezas, si es que lo encuentra en su tabla.
- En el caso en el que el Servidor Intermedio no encuentre la figura solicitada en dentro de su tabla de rutas con los Servidores de Piezas, entonces hará de nuevo un broadcast por si algún nuevo Servidor de Piezas se levanta y tiene la figura pero no estaba en la tabla de rutas del Servidor Intermedio

Conexiones

Los servidores intermedios deben esperar el envío de bytes por parte de los Servidores de Piezas. Si estos, luego de cierta cantidad de tiempo no reciben bytes, deben eliminar la dirección IP de ese servidor de su mapa de rutas, puesto que el Servidores de Piezas no está activo.

Si el cliente le solicita una figura al servidor intermedio, cuyo servidor no tiene en su tabla de rutas, el servidor debe enviar un mensaje de broadcast a través del puerto 4850, con tal de encontrar dicho servidor. El servidor de piezas, entonces, responde con un mensaje indicando su dirección IP y las figuras que contiene.

En caso de que otro servidor intermedio reciba el mensaje, este simplemente descarta la solicitud.

Mensajes

Los mensajes entre cliente y servidor intermedio, y entre servidor intermedio y de piezas, se realizan sobre el protocolo HTTP 1.0. El contenido de los mensajes es sensible a la naturaleza de la solicitud a un servidor, y el puerto donde se realiza la conexión.

En particular, existen estas categorías de mensajes:

- Mensajes de contenido entre cliente y servidor intermedio:
 - Los clientes pueden realizar solicitudes GET para obtener el listado de figuras, y el listado de piezas para alguna figura en particular.
 - Los servidores intermedios son responsables de responder con el listado de figuras en HTML al cliente.
 - La distinción entre una solicitud del listado de figuras y el listado de piezas de alguna figura se realiza mediante el URI asociado al método GET.
 - Un servidor intermedio debe devolver la respuesta HTTP con el listado de piezas de una figura que le respondió un servidor de piezas al cliente que las solicitó.
 - De no poder contactar un servidor de piezas, el servidor intermedio podrá responder con su propia respuesta HTTP.
 - De manera semejante, de obtener una solicitud mal formada, o una pieza inexistente, se debe responder con su propia respuesta HTTP.
 - Aquellas respuestas de los casos de error de redirección a un servidor de piezas deben indicar que no se pudo encontrar la pieza. Se sugieren las respuestas 404 o 500 para mensajes de error HTTP.
- Mensajes de contenido entre servidor intermedio y servidor de piezas:
 - Los servidores intermedios deben enviar la solicitud HTTP de un listado de piezas de una figura enviada por un cliente hacia el servidor de piezas correspondiente.
 - Al obtener la respuesta apropiada a esa solicitud, deben redirigirla hacia el cliente que la solicitó.
- Mensajes de descubrimiento desde el servidor intermedio hacia servidores de piezas:
 - Un servidor intermedio comienza desconociendo los servidores de piezas.

- Cuando un cliente solicita una pieza para la cual el servidor intermedio desconoce el servidor de piezas apropiado, éste debe preguntar mediante broadcast cuáles servidores contienen esa pieza.
- Un servidor de piezas al recibir una solicitud de descubrimiento de piezas de un servidor intermedio debe responder apropiadamente. Se le puede preguntar cuáles figura soporta, o si soporta una pieza o no.
- Cuando un cliente solicita una pieza, y el servidor intermedio conoce el servidor de piezas apropiado, pero éste no responde tras un tiempo de timeout, el servidor intermedio se desconoce de aquel servidor de piezas y responde al cliente con una respuesta indicando que la pieza no se encontró.

El formato de los mensajes HTTP es el siguiente:

- Solicitud de listado de figuras:
 - Método: "GET"
 - URI: "/"
 - Versión HTTP: "HTTP/1.0"
 - Cuerpo: Ignorado
- Respuesta del listado de figuras (encontrado):
 - Estado: "OK" 200
 - Versión HTTP: "HTTP/1.0"
 - Cuerpo: Dirección IP del servidor, un cambio de línea (CRLF), junto a lista separada por comas de figuras que maneja el servidor, terminando en un punto.
- Solicitud de listado de piezas para una figura:
 - Método: "GET"
 - URI: "/" seguido del nombre de la figura
 - Versión HTTP: "HTTP/1.0"
 - Cuerpo: Ignorado
- Respuesta del listado de piezas para una figura (encontrado):
 - Estado: "OK" 200
 - Versión HTTP: "HTTP/1.0"
 - Cuerpo: Listado de piezas para una figura en HTML
- Notas:
 - El listado de figuras y el listado de piezas para una figura se envían HTML.
 - El contenido de las respuestas y solicitudes no tienen encoding particular (ni chunked). Se asume el final del mensaje como el final de la respuesta / solicitud HTML. Se sugiere considerar un timeout en la lectura.
 - Ambos listados son libres de ser implementados según desee cada grupo, siempre y cuando le permita al cliente formar las solicitudes de figuras en el formato apropiado.

Observaciones:

- Formato de los paquetes: 1.2.3.4:1234 UP/DOWN
- Cada grupo tiene un servidor intermedio (sin comunicación entre ellos)
- I1 I2 I3 I4
- P3 P2 P1 P4
- Intermedio primero (intermediario ajusta sus tablas)
 - Broadcast UDP pto (formato)
 - Respuesta Piezas (formato, TCP, IP, pto, {conjunto de figuras}, +)
 - Servidor de piezas termina (formato)
 - Servidor de piezas muere (timeout, intento conectarme y falla)
 - ¿Reconexión?
- Piezas primero
 - Broadcast pto (formato IP, pto, +)
 - Respuesta (formato, IP, pto, +)
- IP pública: 10.1.35.x/16 (conexiones entre cliente/servidor)
- IP privada 172.16.123.x/28 (conexiones entre intermediarios/piezas)
 - 172.16.123.1 o 172.16.123.2 (VPN con firewall)
 - 172.16.123.255 pto
- Intermediario conoce a Piezas
 - ¿Cómo le indica cuál es la pieza que quiere? (formato figure=dragoncito)
 - Formato de la respuesta (cantidad, descripción de la pieza +)

Propuesta grupo 1

[Joseph, Johana, Esteban]

Establecer los esquemas de comunicación:

- Entre los clientes y los servidores intermedios se comunican por medio de una red pública con el puerto 80 (HTTP)
- Entre los servidores intermedios y los servidores de piezas se comunican por medio de una red privada, en un puerto diferente a 80

Cliente: Solicita el menú de figuras y las piezas necesarias a través de la solicitud GET de protocolo HTTP que se envía a través de la URL. Cuando el usuario desea armar una figura, envía una solicitud al servidor intermedio. Debe después recibir una respuesta negativa o positiva sobre si la figura pudo ser armada.

Servidor Intermedio: Contiene el mapa de rutas. Este mapa de rutas se debe actualizar cuando identifica que se agrega un nuevo servidor de piezas. Al recibir la solicitud de armar una figura busca entre todos los servidores de piezas que tenga en su mapa interno para poder encontrar si este puede ser armado o no. Debido a esto, necesita tener en memoria las piezas de las solicitudes pedidas, para poder armar figuras entre las piezas encontradas entre todos los servidores de piezas. De encontrarse todas las piezas en un único servidor, eso es suficiente.

Servidores de piezas: Realiza una revisión de los modelos que almacena y es quien brinda las piezas solicitadas por el cliente. Las piezas son cargadas de un documento txt al iniciar el servidor, y posteriormente pueden ser agregadas a través de una función ya facilitada.

- 4 hilos principales
 1. Escucha por peticiones del cliente
 2. Atiende peticiones del cliente
 3. Escucha por peticiones del browser (mostrar inventario únicamente)
 4. Atiende peticiones del browser (mostrar inventario únicamente)
 5. Se puede agregar un hilo extra para escuchar al standard in, a través de este se puede indicar que se agreguen las piezas de un nuevo o mismo documento y/o otros controles.

Protocolo de comunicación para adicionar o eliminar servidores de piezas a servidores intermedios o viceversa (interacción):

Puertos:

- **Puertos para intermediario:**
 - Puerto: 80(clientes)
 - Puerto: 2432 (otros servidores intermediarios)
 - Puerto: 2560 (servidores de piezas)
- **Puertos para servidor de partes:**
 - Puerto: 2816 (servidor intermediario)
 - Puerto: 2832 (conexión a browser para mostrar inventario únicamente)

Sucesión de eventos:

1. Primer Caso: Servidor intermediario se levanta antes que los servidores de pieza
 - a. Hace un broadcast a todos los puertos 2560 con su IP dentro del servidor y empieza a escuchar por el puerto 2304 a los clientes.
 - b. Al no haber ningún servidor de piezas entonces no recibe respuesta.
 - c. Se mantiene escuchando en el puerto 2304 a que algún servidor de piezas anuncie su levantamiento.
 - d. Al levantar un servidor de piezas, éste realiza un broadcast a todos los puertos 2304 dentro de la red, conjunto con su IP.
 - e. El servidor intermediario recibe el IP, y lo guarda en un mapa local para poder acceder cuando sea necesario.
2. Segundo Caso: Servidor de pieza se levanta primero que el servidor intermediario
 - a. El servidor de piezas realiza un broadcast con su IP a todos los puertos 2560 dentro de la red.
 - b. Al no encontrarse ningún servidor intermediario, no recibe respuesta.
 - c. Se levanta un servidor intermediario, y este realiza un broadcast a todos los puertos 2560 con su IP dentro del servidor y empieza a escuchar por el puerto 2304 a los clientes.
 - d. El servidor de piezas recibe el IP del servidor intermediario y este responde a este IP con su IP.
3. Tercer Caso: conexión de un servidor intermediario con otro servidor intermediario
 - a. El servidor intermediario se levanta y también emite un broadcast con su IP a todos los puertos 2432 dentro de la red.
 - b. El otro servidor intermediario recibe el IP, lo agrega a su mapa local de IPs y responde al IP del otro servidor intermediario con su IP.
 - c. El primer servidor intermediario recibe el IP del otro servidor y lo agrega al mapa local de servidores intermediarios.
4. Cuarto Caso: se borra un servidor intermediario
 - a. Se deja de recibir peticiones por el puerto 2304.
 - b. Se finalizan de procesar las peticiones del servidor.
 - c. Se le envían mensajes a los otros servidores intermediarios con su IP y código de borrado en el último byte.
 - d. Los otros servidores mandan un mensaje confirmación de borrado.
 - e. De recibir un mensaje de confirmación de todos, se borra.
 - f. De no ser así, se vuelve a mandar el mensaje al menos 10 veces más esperando respuesta antes de borrarse de todas maneras y reporta el error.
5. Quinto Caso: se borra un servidor de partes
 - a. Espera a recibir un mensaje del servidor intermediario que lo está utilizando confirmando que se finaliza el chequeo o utilización, en el caso que está siendo utilizado.
 - b. El servidor de partes realiza un broadcast con el código de borrado de servidor de piezas a todos los puertos 2560.
 - c. Todos los servidores intermediarios reciben el mensaje, borran el IP de sus mapas locales y mandan una confirmación de borrado.
 - d. Al recibir los mensajes de confirmación, el servidor local reduce un contador local de servidores intermediarios.
 - e. Al llegar este contador a 0 se borra.

- f. De no llegar a 0, en una cantidad arbitraria de tiempo, este reporta el error antes de borrarse.

Nota: si es necesario diferenciar entre un broadcast y un mensaje de respuesta, se puede agregar un byte al final de cada mensaje con IP y en este se puede tomar el 00000000 como un broadcast, el 00000001 como un mensaje normal, 00000010 cuando se manda un mensaje de borrado de servidor intermediario, y 00000011 cuando se manda un mensaje de borrado de servidor de partes, el resto de los dígitos pueden utilizarse para agregar más funcionalidad funcionalidades en el futuro. Otro uso podría ser reservar los primeros 32 valores para mensajes generales y después asignar 112 a mensajes de servidores de piezas y otros 112 a mensajes de servidores intermediarios, para mantener la organización.

Propuesta grupo 2

[Alonso, Carlos, Mauricio]

1. Definición

El protocolo va a facilitar la comunicación entre servidores para las operaciones del manejo de los recursos disponibles, también tendrá categorías de mensajes, donde un mensaje indicará si es de actualización, registro o petición de recursos, junto a la prioridad del mensaje, esto último para el caso de peticiones múltiples que modifican los recursos y hacen la petición

inalcanzable para el cliente.

Mantendrá la información consistente y verificable, como fechas y registro de la proveniencia del mensaje seguridad y autenticación de los mensajes enviados entre cualquier servidor.

2. Servidor Intermedio

El servidor intermedio deberá ser capaz de:

- Recibir distintos mensajes de los servidores de piezas como de intermedios, como:
 - / Cambio en los recursos
 - Reporte de disponibilidad
 - Reporte de otros servidores intermedios sobre sus servidores de piezas
- Registrar las rutas de los servidores intermedios con una tabla de enrutamiento hacia estos servidores
- Analizar los mensajes de petición en caso de éxito y fracaso, y si fracasa, volver a analizar su registro actualizado para comprobar si puede completar la solicitud del cliente
- Tener una cola de mensajes con los más importantes de primero, para cuando se reciba un mensaje de petición y otro de actualización, y no ocurran inconsistencias

3. Servidor de Piezas

El servidor de piezas deberá ser capaz de:

- Contactar los servidores intermedios para reportar recursos
- Recibir solicitudes de servidores intermedios como:
 - Reporte de disponibilidad
- Tener el control de las solicitudes de recursos como una cola para evitar problemas como condiciones de carrera

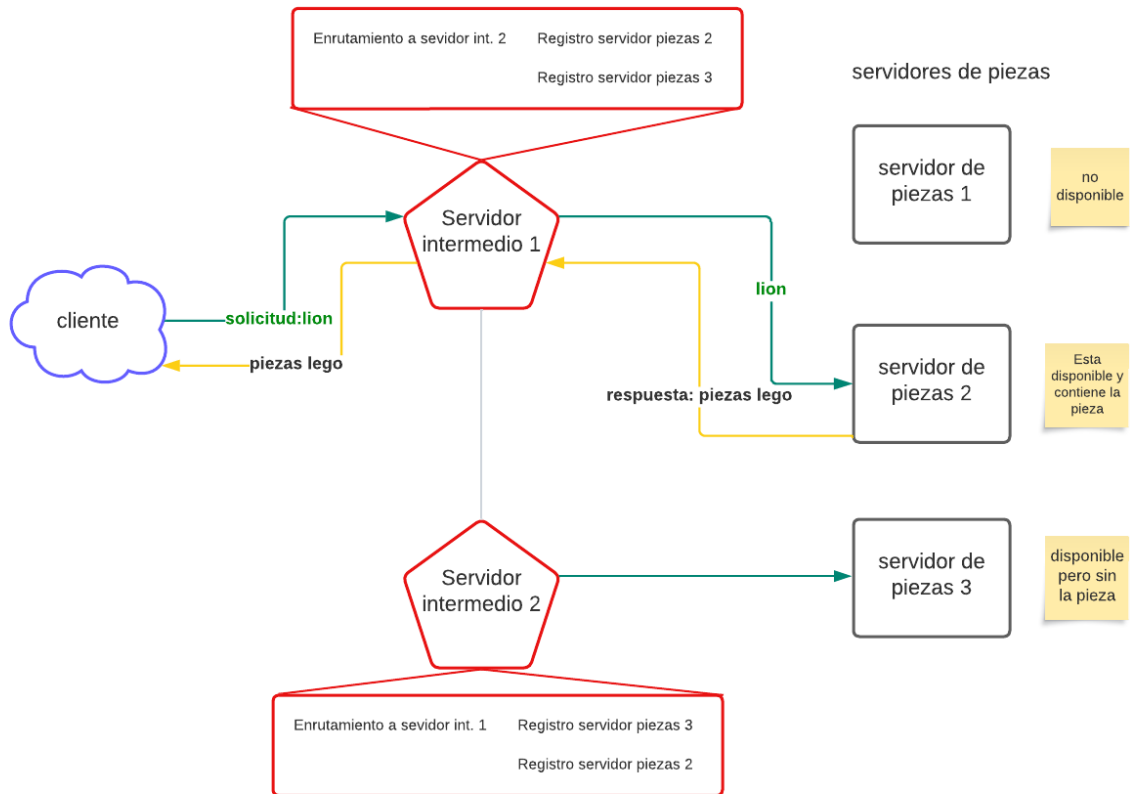
4. Cliente

El cliente va a actuar solicitando un recurso, en este caso una figura y espera recibir una respuesta con un desglose de piezas lego que componen la figura solicitada. Las solicitudes correctas que puede hacer el usuario y mandar el cliente al servidor son las siguientes:

- blacksheep
- dragon
- duck
- elephant
- fish
- giraffe
- horse
- lion
- monkey
- penguin
- roadrunner
- shark
- squid
- swan
- turtle
- whitesheep

Cualquier solicitud diferente a las opciones anteriores va a ser una solicitud incorrecta y se dará una respuesta de figura no disponible.

Se provee un diagrama sobre el funcionamiento del protocolo para su entendimiento.

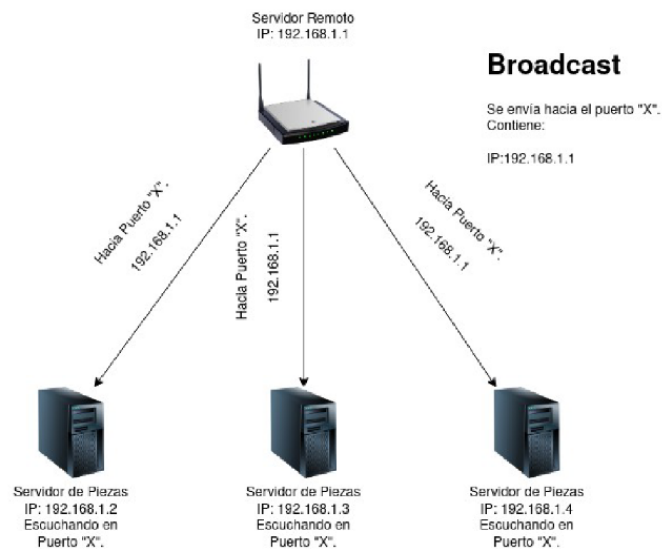


Propuesta grupo 3

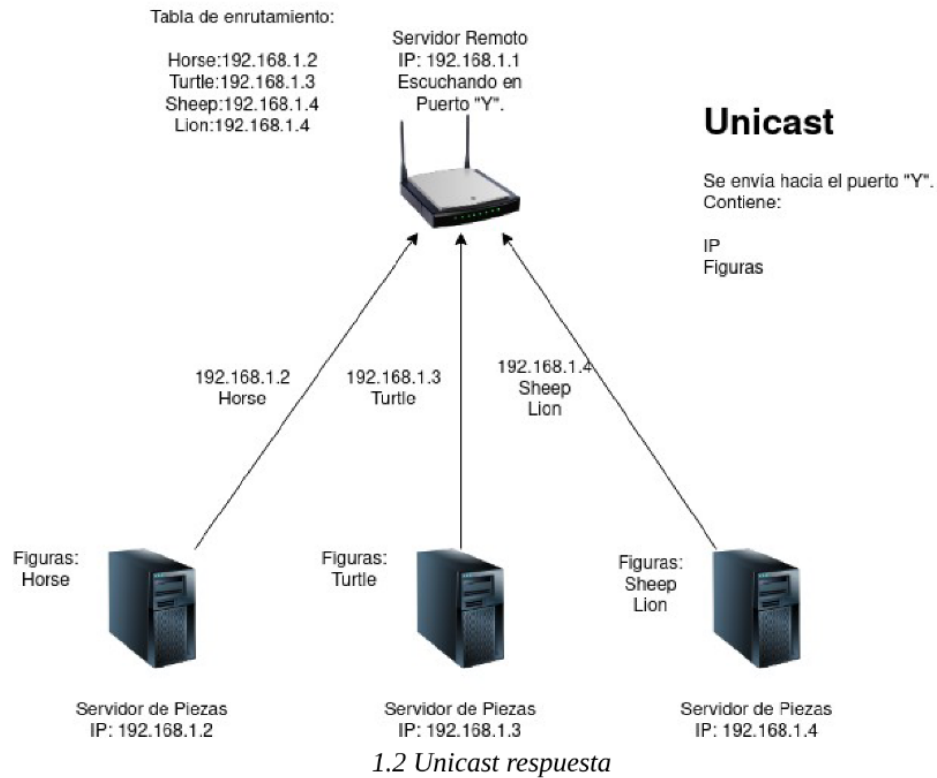
[Jeremy, Richard, Marcel]

1. Protocolo de Conexión Servidor Intermedio - Servidor de Piezas:

Cuando un servidor remoto enciende, realiza un broadcast hacia un puerto "X" en el que los servidores de piezas estarán escuchando, en el mensaje se incluye el ip del servidor remoto. Posteriormente, los servidores de pieza enviarán un mensaje unicast hacia el servidor remoto, en el que se incluye la ip del servidor de piezas así como la lista de figura que posee, el servidor remoto estará escuchando en un puerto "Y". El servidor remoto entonces almacena cada mensaje en una tabla de enrutamiento.

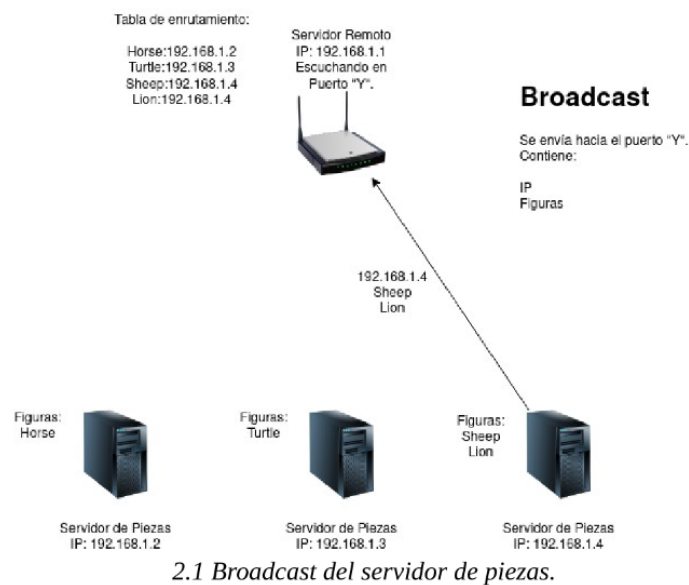


1.1 Broadcast del servidor.



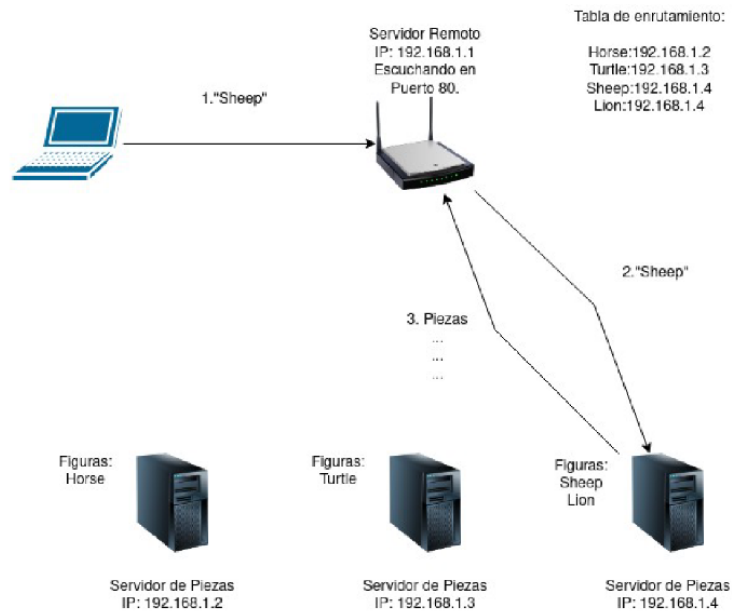
2. Protocolo de Conexión Servidor de Piezas - Servidor Intermedio:

Cuando un servidor de piezas enciende, hace un broadcast hacia un puerto "Y", dónde especifica su IP así cómo su lista de figuras. El servidor intermedio, que está escuchando en el puerto "Y", recibe el mensaje y lo añade a la tabla de enrutamiento.



3. Protocolo de Conexión Cliente-Servidor Intermedio:

Entre los clientes y los servidores intermedios se hace por medio de una red (pública) y se utilizará el puerto 80 (HTTP).



3.1 Proceso completo.

Propuesta grupo 4

[Alexander, Gabriel, Javier]

Protocolo de Adición y Remoción de Servidores

Como se explicó anteriormente es necesario el desarrollo de un protocolo para la adición y remoción de servidores (tanto servidores de piezas como servidores intermedios), para una implementación segura y confiable.

Para esto se debe tener claro el esquema de comunicación:

Esquema de Comunicación

- Los servidores intermedios y de piezas deben estar en la misma red
- Los servidores de piezas deben indicar a los servidores intermedios que un servidor será añadido o removido
- Los servidores intermedios deben indicar a los servidores de piezas que un servidor será añadido o removido
- Los clientes establecen una conexión con los servidores intermedios, nunca con los servidores de piezas
- Los servidores intermedios contienen un mapa de ruta, el cual indica los servidores de piezas que contienen las piezas necesarias para construir cierta figura
- Los servidores de piezas contienen el inventario sobre las piezas que contienen

Puertos

Para la comunicación entre dispositivos en una red se utilizan Sockets, estos Sockets son los encargados de establecer una conexión entre dos elementos. Con el fin de establecer esta conexión, el Socket debe recibir como parámetro una dirección IP (del dispositivo de destino) y un puerto de servicio a través del cual comunicarse.

Debido a que la dirección IP es variable, no es necesario establecerla en el protocolo a seguir, sin embargo el puerto de comunicación sí se puede especificar. Este puerto debe ser mayor a 1024, esto debido a que los puertos anteriores están reservados para servicios conocidos y bien establecidos, como HTTP, TCP, FTP, entre otros.

Por este motivo es necesario establecer un puerto mayor a 1024 para comunicación que no utilice estos protocolos, como es este caso. Debido a que se deben implementar servidores intermedios se implementarán 2 puertos de comunicación exclusivos para la comunicación entre servidores y otro puerto

exclusivo para la comunicación con el cliente.

- El puerto 4849 será utilizado para la comunicación entre los servidores intermedios y los servidores de piezas.
- El puerto 80 (para HTTP) será utilizado para la comunicación entre el cliente y el servidor intermedio.
- El puerto 4851 será utilizado para la comunicación entre servidores intermedios.

Escenarios a evaluar

Se deben analizar los posibles escenarios a los cuales pueden ser sometidos los servidores y cómo estos deben reaccionar en dicha situación.

1. Servidor intermediario es creado

- El nuevo servidor envía un mensaje de broadcast a través del puerto 4849 y 4851.
 - Los servidores de piezas y/o intermediarios (si los hay) reciben el mensaje de broadcast con la flag indicando conexión
 - El nuevo servidor escucha en el puerto 4849 esperando conexiones de servidores de piezas
 - Se establecen conexiones entre servidores
 - Los servidores intermediarios añaden el nuevo servidor a sus mapas de rutas
 - El nuevo servidor añade estas conexiones a su mapa de rutas
 - El nuevo servidor escucha en el puerto 80 para atender a conexiones de clientes
2. Servidor de piezas es creado
- El nuevo servidor envía un mensaje de broadcast a través del puerto 4849.
 - Los servidores intermedios (si los hay) reciben el mensaje de broadcast con la flag indicando conexión
 - El nuevo servidor escucha en el puerto 4849 esperando conexiones de servidores de piezas
 - Se establecen conexiones entre servidores
 - El servidor intermedio añade esta conexión a su mapa de rutas
3. Servidor intermedio es eliminado
- El servidor en cuestión captura un error o es obligado a desconectarse
 - El servidor envía un mensaje de broadcast a través del puerto 4849 y 4851.
 - Los servidores de piezas y/o intermedios (si los hay) reciben el mensaje de broadcast con la flag indicando desconexión
 - El servidor intermedio desconecta los clientes conectados a él en el puerto 4850
 - En caso de interrupción desconocida (interrupción de corriente), los servidores deben esperar cierto tiempo antes de desconexión automática.
 - Los servidores de piezas se desconectan del servidor de piezas especificado
 - Los servidores eliminan al servidor de piezas especificado de su mapa de rutas
 - Se elimina el servidor intermedio
4. Servidor de piezas es eliminado
- El servidor en cuestión captura un error o es obligado a desconectarse
 - El servidor envía un mensaje de broadcast a través del puerto 4849.
 - Los servidores intermedios (si los hay) reciben el mensaje de broadcast con la flag indicando desconexión
 - El servidor de piezas desconecta los clientes conectados a él en el puerto 4849
 - En caso de interrupción desconocida (interrupción de corriente), los servidores deben esperar cierto tiempo antes de desconexión automática.
 - Los servidores de piezas se desconectan del servidor de piezas especificado
 - El servidor intermedio elimina el servidor de piezas de su mapa de rutas

Formato de flags

Los flags pueden tener el formato *ip FLAG*, donde *FLAG* puede ser *UP* o *DOWN*. Estos flags son enviados a través de un puerto específico, de modo que puede utilizarse para ambos tipos de servidores. Las señales deben ser mayúsculas.

