UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Redes – Jorge Yass

Proyecto 1:

Implementación de un protocolo existente

José Alejandro Prince, 22087

Guatemala, 14 de septiembre de 2025

MCP desarrollados

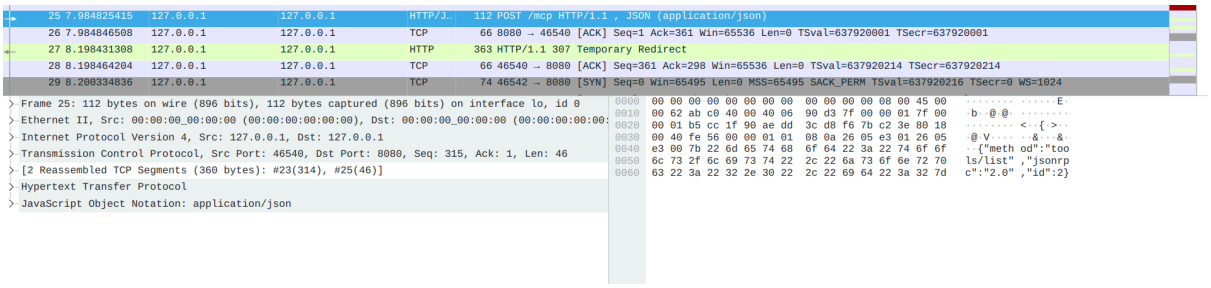
Para la creación del cliente mcp este se conecta con diferentes servidores mcp tanto oficiales como implementaciones propias. Para los servidores oficiales se hace conexión con los servidores mcp de git y filesystem como se indican en las instrucciones del proyecto.

Para la implementación de un servidor mcp propio local se desarrolló con la funcionalidad de poder predecir el “score” de un videojuego en base a las características que tenga este. Este servidor se conecta al cliente utilizando el SDK de stdio, facilitando la implementación del protocolo mcp tanto para el servidor como para el cliente.

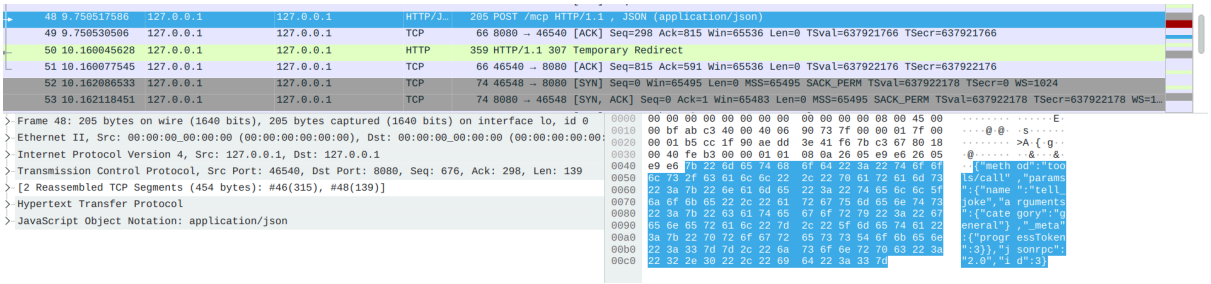
Con respecto al servidor mcp remoto se siguieron las instrucciones para montarlo en google cloud ofrecidas en el documento del proyecto. Las instrucciones para deployar y correrlo se encuentran en ese link. La implementación de este server se encuentra en el repositorio de “ChispitasGPT” para fácil accesibilidad.

Análisis de wireshark

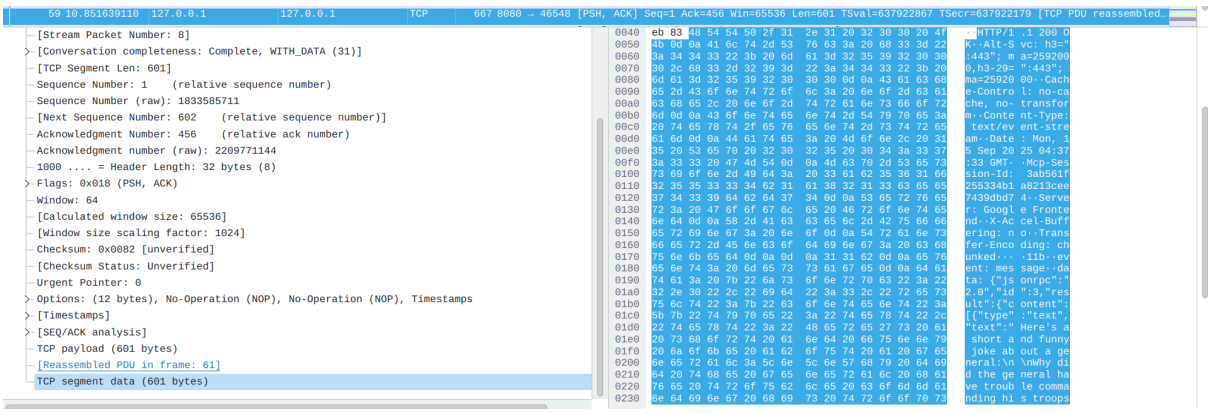
Captura de sincronización:

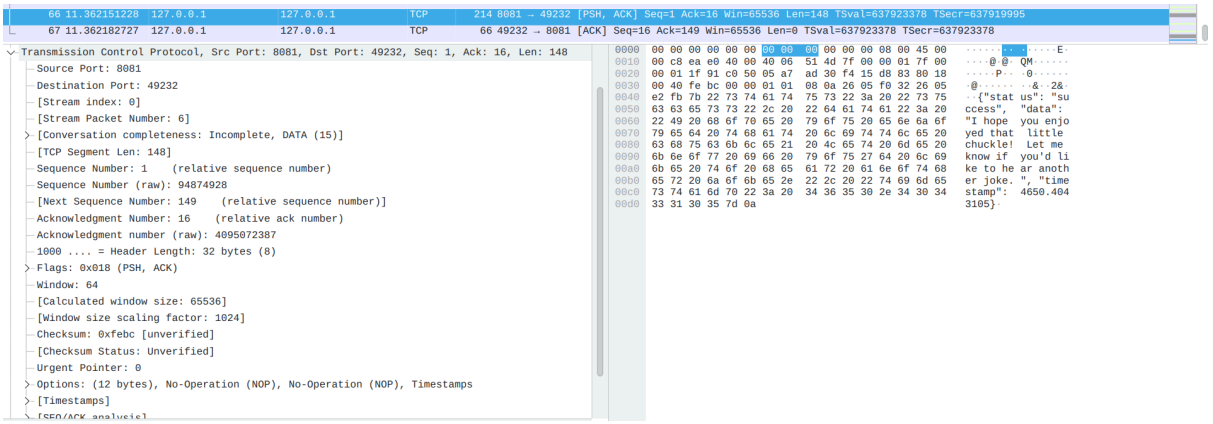


Captura de petición:



Capturas de respuestas:





A nivel de la capa de enlace lo que sucede es que se encapsulan los datos en tramas y se transmiten por el medio físico. En este caso el servidor remoto esta haciendo un forwarding por medio del puerto 8080 del localhost a la ip del servidor. Cada trama incluye la dirección de origen y destino.

En la capa de red se arma un paquete IP con la dirección IP de origen y destino, en este caso ambas son 127.0.0.1 (localhost) debido al forwarding que se hace. El paquete al llegar al puerto 8080 es enviado a la red para ser recibido por el servidor mcp remoto.

Para la capa de transporte, MCP usa TCP. Antes del POST que se ve en la captura de sincronización es que ocurre el three-way-handshake. Luego de esto el cliente abre un socket TCP hacia el puerto destinatario.

En la capa de aplicación se ve que el protocolo usado es HTTP/1.1, sobre el cual viaja el JSON-RPC de MCP. En este caso el servidor MCP puede responder de 3 formas: “initialize” hace el handshake de MCP, “list_tools” responde con las herramientas disponibles y “call_tool” ejecuta la tool y devuelve el resultado. La respuesta del servidor llega en formato JSON dentro del HTTP.

Conclusiones

- El protocolo MCP nos ayuda a realizar la comunicación para las herramientas de LLM, con este protocolo estamos generalizando la comunicación y ayudando a que los LLM obtengan más capacidades.
- El envió de datos del protocolo MCP es mediante JSON-RPC, este formato guarda datos útiles que son interpretados por los LLM y permite identificar la tool que se debe de utilizar y la respuesta que se envía.
- La implementación de SDK con el protocolo ya existente ayudó en el desarrollo, agilizando y permitiendo que se desarrollaran varias funcionalidades a nivel de frontend que mejoraron la calidad del proyecto.

Comentarios

- Seria interesante explorar realizar la implementación del protocolo sin la utilización de SDK, de esta forma se tiene un aprendizaje más certero sobre este protocolo.
- Para facilitar la comunicación es mejor si el lenguaje con el que se hace la GUI es el mismo con el que se hace el cliente MCP.