

Universidad del Valle de Guatemala

Facultad de Ingeniería Ciencias de la Computación



Proyecto 1

Uso de un protocolo existente

Redes

Angel Herrarte 22873

22 de septiembre del 2025

Análisis de la comunicación entre el servidor remoto y el cliente

1. Mensaje de Solicitud (Request) - Frame 229634

- Origen: Cliente (192.168.0.7)
- Destino: Servidor (66.33.22.59)
- Protocolo: HTTP/JSON sobre TLS
- Método HTTP: POST /mcp HTTP/1.1
- Contenido JSON-RPC:

```
json
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/call",
  "params": {
    "name": "get_weather",
    "arguments": {
      "city": "guatemala"
    }
  }
}
```

- Tipo: Este es un mensaje de solicitud o petición donde el cliente invoca el método `tools/call` con parámetros específicos.

2. Mensaje de Respuesta (Response) - Frame 230395

- Origen: Servidor (66.33.22.59)
- Destino: Cliente (192.168.0.7)
- Protocolo: HTTP/JSON sobre TLS
- Respuesta HTTP: HTTP/1.1 200 OK
- Contenido JSON-RPC:

```
json
{
  "success": true,
  "city": "Guatemala City",
  "temperature": 21.02,
  "humidity": 100,
  "pressure": 1013,
  "description": "nubes",
  "main_weather": "Clouds",
  "wind_speed": 3.58,
  "wind_direction": 200,
  "timestamp": "2025-09-21T20:35:16.848368"
}
```

- Tipo: Este es un mensaje de respuesta a la solicitud anterior, conteniendo los resultados de la operación solicitada.

3. Mensajes de Sincronización

En la captura se observan varios mensajes TCP de control que forman la sincronización de la conexión:

- Frames 228929, 229169, 229170: Handshake TCP (SYN, SYN-ACK, ACK)
- Frames 229198-229199: Handshake TLS (Client Hello)
- Frames 229592-229596: Negotiation TLS (Server Hello, Certificate, etc.)
- Frames 229630-229631: Finalización handshake TLS (Change Cipher Spec, Finished)

Documentación de servidores MCP

Filesystem MPC

https://github.com/aherrarte2019037/git-mcp-server/blob/main/docs/FILESYSTEM_MCP.md

Git MPC

https://github.com/aherrarte2019037/git-mcp-server/blob/main/docs/GIT_MCP.md

Git Analyzer MPC

https://github.com/aherrarte2019037/git-mcp-server/blob/main/docs/GIT_ANALYZER_MCP.md

Weather MPC

https://github.com/aherrarte2019037/git-mcp-server/blob/main/docs/WEATHER_MCP.md

Que sucede a nivel de la capa de enlace, red, transporte y aplicación.

Capa de Enlace (Ethernet)

- Dirección MAC destino: Router local (c0:89:ab:eb:d5:16)
- Dirección MAC origen: Cliente (06:34:e6:94:64:f2)
- Protocolo: IPv4 (0x0800)
- *Función*: Transmisión de frames entre dispositivos en la red local.

Capa de Red (IP)

- IP origen: 192.168.0.7 (cliente)
- IP destino: 66.33.22.59 (servidor remoto)

- TTL: 64 (cliente), 45 (servidor)
- *Función*: Enrutamiento de paquetes a través de Internet.

Capa de Transporte (TCP)

- Puertos: Cliente (51885) → Servidor (443/HTTPS)
- Handshake TCP: SYN → SYN-ACK → ACK
- ACKs: Confirmaciones de recepción (ej: frame 230396)
- *Función*: Comunicación confiable y ordenada entre aplicaciones.

Capa de Aplicación

- TLS: Cifrado de la comunicación (handshake en frames 229199, 229592)
- HTTP/JSON-RPC:

Solicitud (POST /mcp):

```
json
{"method":"tools/call","params":{"city":"guatemala"}}
```

Respuesta (200 OK):

```
json
{"temperature":21.02,"humidity":100,...}
```

- *Función*: Intercambio de datos específicos entre aplicaciones cliente-servidor.

Conclusiones y comentario sobre el proyecto

Implementación

- 4 servidores MCP (2 locales, 1 remoto, 1 custom)
- Chatbot inteligente con detección de intenciones
- Análisis técnico de comunicación de red

Demostración de Concepto

- MCP es viable en la práctica
- Integración exitosa de múltiples servidores
- Escalabilidad probada

Valor Educativo

- Comprensión de protocolos (JSON-RPC, HTTP, TLS)
- Análisis de red con Wireshark
- Arquitectura distribuida

Lecciones Aprendidas

Fortalezas del Protocolo MCP:

- Interoperabilidad independiente del LLM
- Modularidad y reutilización
- Estándar abierto bien documentado

Desafíos Técnicos:

- Inicialización compleja de servidores
- Manejo de errores específicos por servidor
- Análisis de tráfico encriptado

Soluciones Implementadas:

- Sistema de clientes modulares
- Detección híbrida de intenciones
- Logging detallado para debugging

El proyecto ha sido exitoso en demostrar la viabilidad del Model Context Protocol (MCP), estableciendo una base sólida para futuros desarrollos en integración de IA con herramientas externas.

El futuro de la IA está en la integración inteligente con herramientas externas, y MCP es el protocolo que lo hará posible.