Proyecto Vehículo Autónomo Telematica

Especificación del Protocolo de Comunicación

Camilo Andres Melo Alvarez,

Cesar Augusto Montoya

Isabela Mendoza

Universidad EAFIT

2025-10-05

Especificación del Protocolo de Comunicación

El presente apartado describe la especificación formal del protocolo de comunicación implementado en el proyecto de vehículo autónomo. Este protocolo define la forma en que el cliente de control y el servidor de telemetría intercambian información a través de una conexión TCP/IP, empleando la API de sockets de Berkeley.
Su propósito es garantizar un canal confiable, bidireccional y en tiempo real para la transmisión de comandos de control, datos de estado y mensajes de autenticación.

El modelo de comunicación adoptado responde a una arquitectura cliente–servidor persistente, en la cual el servidor mantiene múltiples sesiones simultáneas y cada cliente conserva su conexión mientras la sesión esté activa. Los mensajes se transmiten en formato de texto plano codificado en UTF-8, con los campos separados por espacios y terminados en salto de línea (\n). Este enfoque permite una comunicación sencilla, legible y fácilmente depurable durante el desarrollo o la prueba de red.

Estructura general de los mensajes

Cada intercambio entre el cliente y el servidor sigue la estructura:

<COMANDO> [argumentos] [opciones]

Las palabras clave se escriben en mayúsculas; los argumentos se separan por espacios y los parámetros opcionales se expresan mediante pares clave=valor.
Por ejemplo:

AUTH admin 1234
CMD SPEED UP
TELEMETRY v=3.2 battery=95 dir=N timestamp=2025-10-05T20:45:00Z

Este formato uniforme facilita el análisis de comandos y la extensión futura del protocolo sin alterar su compatibilidad.

Tipos de mensajes

El protocolo define tres grandes categorías de mensajes:

1. Mensajes de control (cliente → servidor): incluyen comandos de autenticación, suscripción y control del vehículo. Ejemplos:
    - AUTH <usuario> <contraseña> para iniciar sesión.
    - SUBSCRIBE OBSERVER para recibir telemetría.
    - CMD TURN LEFT o CMD SPEED UP para enviar acciones de manejo.
    - QUIT para finalizar la sesión.
2. Mensajes de respuesta (servidor → cliente): confirman o niegan las solicitudes recibidas. Ejemplos:
    - AUTH-OK token=<token> si la autenticación fue exitosa.
    - CMD-ACK action=SPEED_UP status=OK si la orden fue ejecutada.
    - CMD-ERR action=SPEED_UP reason=battery_low si ocurrió un error.
    - ERR not_authorized en caso de intentos no válidos.
3. Mensajes de telemetría (servidor → cliente): se envían automáticamente cada 10 segundos a todos los clientes suscritos.
   Tienen el formato:
4. TELEMETRY v=<velocidad> battery=<porcentaje> dir=<dirección> timestamp=<ISO8601>

   Por ejemplo:
   TELEMETRY v=2.5 battery=97 dir=E timestamp=2025-10-05T21:00:00Z.
   Este mensaje informa la velocidad actual del vehículo en m/s, el nivel de batería en porcentaje, la dirección cardinal (N, E, S, W) y la hora UTC del envío.

Reglas de procedimiento

1. Establecimiento de conexión.
   El cliente inicia una sesión mediante una conexión TCP hacia el puerto configurado del servidor. Una vez aceptada, el servidor crea un hilo dedicado para manejar la comunicación con ese cliente.
2. Autenticación y roles.
   - Los administradores deben autenticarse con AUTH <usuario> <contraseña>.
     El servidor compara el hash SHA-256 del valor salt + contraseña almacenado en el archivo credentials.txt.
     Si la verificación es exitosa, el servidor responde con AUTH-OK token=<token> y concede privilegios administrativos.
   - Los observadores pueden suscribirse sin autenticación mediante SUBSCRIBE OBSERVER.
3. Control del vehículo.
   El administrador autenticado puede enviar comandos CMD para modificar el estado del vehículo virtual (velocidad, dirección y orientación).
   El servidor valida la batería y los límites operativos antes de confirmar la orden con CMD-ACK o CMD-ERR.
4. Telemetría periódica.
   Un hilo secundario del servidor emite cada 10 segundos un mensaje TELEMETRY a todos los clientes activos.
   Los clientes actualizan sus paneles de control en tiempo real con los valores recibidos.
5. Finalización de la sesión.
   Cuando el cliente envía QUIT, el servidor responde con BYE, cierra el socket y elimina la referencia del cliente.

Ejemplo de secuencia de comunicación

Cliente → SUBSCRIBE OBSERVER
Servidor → SUBSCRIBE-OK role=OBSERVER
Servidor → TELEMETRY v=0.0 battery=100 dir=N timestamp=2025-10-05T20:00:00Z

Cliente → CMD SPEED UP
Servidor → CMD-ACK action=SPEED_UP status=OK
Servidor → TELEMETRY v=2.5 battery=99 dir=N timestamp=2025-10-05T20:00:10Z
Cliente → QUIT
Servidor → BYE

Consideraciones de seguridad

Las credenciales del administrador se almacenan de manera protegida utilizando el algoritmo SHA-256 con salt aleatorio, evitando el guardado de contraseñas en texto plano.
El servidor genera tokens de sesión únicos de 32 caracteres hexadecimales que permiten la reconexión autenticada.
Aunque la comunicación se basa en texto plano, se recomienda implementar el protocolo dentro de redes seguras o sobre túneles cifrados (VPN o SSH) para prevenir intercepciones.

Conclusión

El protocolo propuesto establece un marco de comunicación claro, extensible y confiable entre el cliente y el servidor del sistema de vehículo autónomo.
Su sintaxis basada en comandos de texto y su implementación sobre TCP permiten garantizar la integridad del intercambio y la interoperabilidad entre distintos lenguajes de programación.
Además, su diseño modular facilita la incorporación de nuevas funciones de control y monitoreo sin afectar la compatibilidad con las versiones actuales del software.