Revisión de Protocolos de Comunicación para Sistemas Multiagentes

1st Bryan Mendoza
Facultad de Ingeniería
Universidad de Cuenca
Cuenca, Ecuador
bsteven.mendoza7@ucuenca.edu.ec

***Abstract*—Este artículo presenta un análisis de protocolos de comunicación en sistemas multiagente, destacando tres enfoques principales. Primero, el uso de ontologías para facilitar la interoperabilidad semántica entre agentes heterogéneos. Segundo, un estudio comparativo de los diferentes protocolos de comunicación más utilizados, evaluando sus fortalezas y limitaciones en varios escenarios. Finalmente, se revisa un protocolo basado en activación por eventos y predicción para sistemas con retrasos de comunicación, que mejora la eficiencia y reduce el uso de recursos. Se concluye que la selección del protocolo debe basarse en la naturaleza del problema y que existen múltiples oportunidades para nuevas implementaciones y combinaciones de métodos en sistemas multiagente.**

***Keywords*—Protocolos de comunicación, Multi-Agent Systems, MAS, Ontología, FIPA-ACL.**

I. Introducción

En las últimas décadas, los sistemas multiagente (MAS) han ganado relevancia en aplicaciones de ingeniería, desde robótica colaborativa hasta redes de sensores inteligentes [1]. Una característica fundamental de estos sistemas es su capacidad para operar de manera distribuida, donde agentes autónomos interactúan mediante protocolos de comunicación estandarizados para alcanzar objetivos colectivos. Estos protocolos, que definen reglas para el intercambio de información, son críticos en la coordinación de acciones, el compartir estados y la toma de decisiones consensuadas [2].

Sin embargo, al trasladar estos sistemas a entornos realistas, surgen desafíos significativos: limitaciones de ancho de banda, retardos variables en las redes y restricciones energéticas [3]. Los métodos tradicionales basados en actualizaciones continuas, aunque efectivos en entornos ideales, resultan ineficientes en escenarios con recursos limitados debido a su alto costo computacional y energético [4]. Esto ha impulsado el desarrollo de estrategias de comunicación adaptativas, como los protocolos activados por eventos, que reducen la carga de transmisión sin comprometer la convergencia del sistema [5].

El objetivo principal de este trabajo es analizar críticamente los protocolos de comunicación en MAS, comparando sus ventajas, desventajas y casos de uso. Para ello, se estudian tres contribuciones recientes: la ontología FIPA-ACL, estándar para la interoperabilidad semántica entre agentes heterogéneos [6]; un análisis comparativo de protocolos existentes [7]; y nuestra propuesta de protocolo híbrido, que combina activación por eventos y predicción de estados [8].

Este estudio proporciona un marco de referencia para seleccionar protocolos según restricciones específicas (latencia, energía o complejidad computacional), contribuyendo así al diseño de MAS más robustos y eficientes.

II. A FIPA-ACL Ontology in Enhancing Interoperability Multi-Agent Communication

El artículo se desarrolla en el contexto de los sistemas multiagente (MAS), los cuales operan bajo un paradigma computacional distribuido, dinámico y abierto. En estos entornos, múltiples agentes heterogéneos deben interactuar y colaborar para alcanzar objetivos comunes, utilizando para ello un lenguaje de comunicación de agentes (ACL). En paralelo, la web semántica ha emergido como una extensión de la World Wide Web, añadiendo significado explícito al contenido de la información con el fin de que pueda ser procesada automáticamente por máquinas (agentes, en este caso).

Este artículo propone el desarrollo de una ontología basada en el estándar FIPA-ACL para optimizar la interoperabilidad en la comunicación entre agentes. La idea central es combinar las ventajas de FIPA-ACL, un lenguaje de comunicación estandarizado desarrollado por la Foundation for Intelligent Physical Agents (FIPA) con los beneficios de las ontologías de la web semántica mediante el uso de OWL 2 DL. Esta combinación permite tener una representación lógica decidible, aportando mayor precisión y eficiencia en la interpretación y el procesamiento de los mensajes intercambiados.

A. Desarrollo de la ontología FIPA-ACL

El desarrollo de la ontología se llevó a cabo siguiendo el enfoque metodológico de ingeniería ontológica, el cual implica iteraciones continuas para su refinamiento y evaluación. Los pasos a seguir del proceso son:

1) **Determinación del dominio y alcance:** Se establece como dominio la especificación FIPA-ACL, abarcando aspectos como actos comunicativos, protocolos de interacción, lenguaje de contenido y la estructura del mensaje. La ontología está destinada a ser empleada por agentes que intercambian información conforme a las normas FIPA. Además, permite inferir el performativo (acto comuniativo) adecuado en función del contenido y del contexto del mensaje, validando si el agente dispone

del conocimiento necesario para ejecutar determinadas acciones.

2) **Reutilización de ontologías existentes:** Tomaron como base tres ontologías previas: la desarrollada por Zou (centrada en OWL como lenguaje de contenido), la de Dickson (basada en el modelo mentalista) y la ontología de obligación de Fornara, la cual incorpora un modelo semántico basado en compromisos sociales derivados de la interacción. Esta estrategia no solo garantiza la coherencia del modelo, sino que también potencia la interoperabilidad al aprovechar desarrollos consolidados.
3) **Enumeración de términos relevantes:** Los términos se extrajeron de las especificaciones FIPA-ACL y de otras fuentes relevantes. Los sustantivos se modelaron como clases (ej. Agente, Mensaje, Protocolo), mientras que los verbos se convirtieron en propiedades que conectan las clases.
4) **Definición de clases y jerarquía de clases:** Se emplea un enfoque híbrido, comenzando por conceptos clave y aplicando generalización y especialización.
5) **Definición de propiedades de las clases:** Se identifican las propiedades de objeto y de datos que establecen las relaciones entre clases. Por ejemplo, hasSender relaciona un mensaje con su agente emisor, y hasProtocol indica el protocolo asociado a un mensaje.
6) **Definición de facetas de atributos:** Se establecen características como tipo de valor, cardinalidad, dominio y rango, utilizando tipos de datos definidos por XML Schema compatibles con OWL 2 DL.
7) **Creación de instancias:** Inicialmente no se definieron instancias, ya que se espera que estas se generen dinámicamente durante el intercambio de mensajes entre agentes.

El artículo presenta de manera detallada el proceso metodológico para la creación de una ontología FIPA-ACL completa basada en OWL 2 DL, con el objetivo de mejorar la interoperabilidad en la comunicación entre agentes. Esta combinación permite aprovechar el poder expresivo de OWL y garantiza la decidibilidad lógica, superando limitaciones de enfoques anteriores basados en lógica modal o de primer orden. Sin embargo, este artículo no incluye una implementación práctica, solo proporciona una guía metodológica sólida que puede ser reutilizada o extendida por otros desarrolladores de sistemas multiagente.

III. Analysis of Communication Protocols in Multi-Agent Systems

Este artículo presenta un análisis comparativo de los principales protocolos de comunicación utilizados en MAS, con el objetivo de orientar la selección del protocolo más adecuado según las necesidades del sistema, considerando factores como complejidad, eficiencia, recursos y seguridad.

A. Protocolos ACL (Agent Communication Languages)

- **FIPA-ACL:** Es un lenguaje de comunicación basado en actos performativos (como solicitud, propuesta o rechazo), que permite a los agentes interpretar la intención del mensaje y tomar decisiones según el contexto. Es ideal para entornos distribuidos complejos donde se requiere intercambio de información estructurada y semánticamente rica.
  Desventajas:
  - Complejidad: La estructura de los mensajes y los performativos pueden añadir complejidad al desarrollo.
  - Sobrecarga: La abstracción y la riqueza de la comunicación pueden generar sobrecarga, lo que puede no es adecuado para entornos en tiempo real o con recursos limitados.
  - Interoperabilidad: Diferencias en ontologías, lenguajes o plataformas pueden generar problemas si no se gestionan correctamente.
  - Rendimiento: El protocolo de comunicación de alto nivel puede introducir ineficiencias en sistemas que demandan alta velocidad.
- **KQML (Knowledge Query and Manipulation Language):** Ofrece una encapsulación flexible de mensajes y separa el contenido del mensaje de su estructura.
  Ventajas:
  - Simplicidad: Es más fácil de implementar que lenguajes de comunicación más complejos como FIPA-ACL.
  - Flexibilidad: Admite diversos lenguajes de contenido, lo que lo hace versátil para diferentes dominios.
  - Expresividad: Abarca una amplia gama de actos comunicativos.

  Desventajas:
  - Falta de semántica: KQML no aplica una semántica sólida, lo que puede causar inconsistencias en la interpretación.
  - Sin protocolos integrados: Carece de protocolos predefinidos para las interacciones entre agentes.
  - Adopción limitada: Su uso ha disminuido en favor de alternativas más estructuradas como FIPA-ACL.

  Aunque KQML es más simple y flexible, FIPA-ACL ofrece una comunicación más estructurada y con semántica formal, siendo más apropiado para sistemas multiagente complejos. KQML, en cambio, se adapta mejor a entornos ligeros o menos exigentes.

B. Protocolos de Publicación - Suscripción

Se utilizan para distribuir mensajes de forma eficiente y escalable en sistemas con múltiples emisores y receptores.

- **MQTT (Message Queuing Telemetry Transport):** Protocolo ligero ideal para IoT y redes con recursos limitados. Utiliza un intermediario que recibe mensajes de los publicadores y los reenvía a los suscriptores basándose en temas.
  Soporta tres niveles de calidad de servicio (QoS):
  - **Máximo una vez:** El mensaje se envía solo una vez y puede perderse.

  - **Al menos una vez:** Se garantiza la entrega, pero puede haber duplicados.
  - **Exactamente una vez:** Asegura que el mensaje se reciba solo una vez, ideal para datos críticos.

  Ventajas:
  - Baja sobrecarga: Ideal para dispositivos con recursos limitados.
  - Fácil de implementar.
  - Escalabilidad: Admite implementaciones a gran escala con un consumo mínimo de recursos.

  Desventajas:
  - Sin persistencia de mensajes integrada.
  - Seguridad básica.

- **AMQP (Advanced Message Queuing Protocol):** Protocolo robusto orientado a sistemas empresariales que requieren alta fiabilidad y funciones avanzadas. También usa un intermediario, pero permite patrones de mensajería más sofisticados y garantiza que los mensajes se almacenen y se entreguen sin pérdida.
  Ofrece dos garantías de entrega:
  - Al menos una vez: Garantiza la entrega del mensaje, aunque es posible que se dupliquen.
  - Exactamente una vez: Garantiza que cada mensaje se entregue exactamente una vez sin duplicaciones.

  Ventajas:
  - Fiabilidad: Proporciona mecanismos robustos para garantizar la entrega de mensajes.
  - Funciones avanzadas. Admite múltiples patrones de enrutamiento de mensajes.
  - Seguridad: Incluye mecanismos integrados para una comunicación segura.

  Desventajas:
  - Alto consumo de recursos.
  - Complejidad.

  Mientras que MQTT es ideal para entornos con recursos limitados y aplicaciones IoT gracias a su baja sobrecarga y simplicidad, AMQP se adapta mejor a sistemas empresariales que demandan alta fiabilidad, seguridad y patrones de mensajería complejos, pero con un mayor consumo de recursos y complejidad.

C. Protocolos de Solicitud - Respuesta

Son esenciales para la comunicación cliente-servidor.

- **HTTP/REST (Hypertext Transfer Protocol / Representational State Transfer):** HTTP es el protocolo base para la comunicación web, mientras que REST es un estilo arquitectónico que utiliza métodos HTTP (GET, POST, PUT, DELETE). Cada solicitud es independiente y auto contenida, lo que favorece un diseño sin estado muy utilizado en APIs web, generalmente utilizando JSON o XML para transferir datos. **Ventajas:**
  - Escalabilidad: La ausencia de estado facilita el escalado horizontal.
  - Simplicidad: HTTP al ser un protocolo consolidado, facilita su implementación.

  Desventajas:
  - Sobrecarga: Las cabeceras de HTTP pueden ser grandes, lo que lo hace ineficiente en entornos restringidos.
  - Latencia: La alta sobrecarga de la red y la falta de estado pueden generar retrasos, especialmente en aplicaciones en tiempo real.
- **CoAP (Constrained Application Protocol):** Diseñado para dispositivos y redes con restricciones, CoAP opera sobre UDP, lo que reduce considerablemente la sobrecarga. Utiliza un modelo de solicitud-respuesta similar a HTTP, pero está optimizado para entornos de bajo consumo y permite tanto comunicación unidifusiva como multidifusiva. **Ventajas:**
  - Eficiencia: Utiliza menos ancho de banda y energía, con mensajes de menor tamaño.
  - Interoperabilidad: Puede mapearse fácilmente a HTTP, facilitando la integración con sistemas web.

  Desventajas:
  - Conjunto limitado de funciones: Carece de algunas características robustas de HTTP, como mecanismos de seguridad integrada.
  - Fiabilidad: Al estar basado en UDP, existe mayor probabilidad de pérdida de paquetes.

  HTTP/REST, al estar basado en TCP, ofrece fiabilidad y seguridad, siendo ideal para sistemas donde estas cualidades son cruciales, mientras que CoAP es más adecuado para entornos con recursos limitados donde la eficiencia y el bajo consumo son prioritarios, a pesar de los posibles problemas de pérdida de paquetes.

D. Protocolos de Seguridad

Esenciales para garantizar la seguridad de las comunicaciones, la integridad de los datos y la fiabilidad del sistema.

- **TLS (Transport Layer Security):** Protocolo criptográfico ampliamente usado para proteger el tráfico web, correos electrónicos y otras comunicaciones en Internet. Emplea cifrado simétrico y asimétrico para resguardar la confidencialidad de los datos, junto con funciones hash para asegurar la integridad y autenticar la información. **Ventajas:**
  - Amplia adopción en la web y es considerado un estándar para comunicaciones seguras.
  - Proporciona seguridad de extremo a extremo al cifrar los datos entre cliente y servidor.

  Desventajas:
  - Consume bastantes recursos, lo que puede introducir latencia y sobrecarga.
  - Requiere una infraestructura de clave pública (PKI) para la gestión de certificados.
- **Seguridad basada en Blockchain:** Se basa en la tecnología blockchain, aprovechando su naturaleza descentralizada, inmutable y transparente para proteger datos

y transacciones. Distribuye un libro de contabilidad entre todos los participantes, enlazando bloques mediante hashes criptográficos. Los mecanismos de consenso, como la prueba de trabajo (PoW) o la prueba de participación (PoS), validan y aseguran las transacciones contra manipulaciones. **Ventajas:**

- Inmutabilidad. Los datos no pueden modificarse sin consenso, garantizando integridad.
- Descentralización. Elimina la necesidad de confiar en una autoridad central, distribuyendo la confianza en la red.

Desventajas:

- Puede enfrentar problemas de escalabilidad, con velocidades de transacción más lentas que los sistemas centralizados.
- Los mecanismos de consenso como PoW pueden consumir mucha energía.

Mientras TLS es un protocolo centralizado enfocado en proteger la comunicación en Internet mediante cifrado y autenticación, la seguridad basada en blockchain ofrece un enfoque descentralizado, garantizando la integridad y transparencia de los datos a través de mecanismos de consenso y funciones hash.

En los MAS, los protocolos de comunicación son esenciales para lograr una interacción efectiva entre agentes, ya sea en entornos centralizados, descentralizados o híbridos. Protocolos como HTTP/REST y CoAP permiten comunicación directa estructurada, mientras que MQTT y AMQP favorecen la comunicación asíncrona y escalable. En cuanto a seguridad, TLS es ideal para entornos web, y Blockchain destaca por su integridad y confianza en contextos distribuidos.

IV. External Consensus Protocol of Networked Multiagent Systems With Communication Delays Under Event-Triggered Mechanisms

Este artículo aborda el problema del consenso externo en sistemas multiagente en red (Networked Multiagent Systems - NMAS), específicamente en escenarios donde existen retrasos de comunicación y se desea reducir la frecuencia de transmisión de datos. El objetivo es diseñar un protocolo distribuido que permita a los agentes alcanzar un consenso con respecto a una entrada de referencia externa, la cual no es conocida por todos desde el inicio, enfrentando además las restricciones impuestas por los retardos en la red y la necesidad de eficiencia comunicativa.

A. Metodología

La metodología propuesta para hacer frente a estos desafíos combina dos elementos clave:

- **Mecanismo de activación por eventos:** En lugar de actualizar continuamente el controlador y transmitir datos, cada agente monitorea su estado y envía información únicamente cuando se cumple una condición disparadora basada en el error de medición y la diferencia entre su salida actual y la salida predicha. Este mecanismo reduce significativamente la cantidad de actualizaciones y comunicaciones, lo que resulta en un uso más eficiente de los recursos del sistema.
- **Esquema de predicción recursiva:** Dado que los retrasos en la comunicación pueden afectar la precisión del control, se emplea una estrategia de predicción que estima de forma recursiva el estado futuro de los agentes, basándose en información histórica. Con esta predicción, se compensa activamente el efecto del retardo, permitiendo ajustar de manera eficaz las acciones de control aunque las actualizaciones no sean constantes en el tiempo.

El artículo demuestra, desde un punto de vista teórico, que el protocolo propuesto garantiza el consenso externo bajo una topología de red conectada y no dirigida, siempre que los retrasos de comunicación sean constantes. Además, se establece que el método evita el fenómeno de Zeno, es decir, que no se produzca un número infinito de disparos en un tiempo finito, lo que asegura la viabilidad práctica del protocolo. Comparado con los enfoques tradicionales de control continuo, esta estrategia resulta menos conservadora, más eficiente y más adecuada para sistemas distribuidos con restricciones de comunicación.

B. Simulaciones y Resultados

La validez del enfoque es comprobada mediante simulaciones con sistemas físicos representados por tanques de agua interconectados. En las pruebas, se emplearon escenarios con dos y cinco agentes, y se evaluó la capacidad del protocolo para lograr el seguimiento de una señal externa con diferentes configuraciones de topología y niveles de retardo. Los resultados muestran que el protocolo mantiene el seguimiento adecuado y la estabilidad del sistema incluso con retrasos de hasta 0.3 segundos. Además, se evidencia una importante reducción en la cantidad de actualizaciones de control respecto a métodos continuos.

En conclusión, el artículo presenta un protocolo de consenso externo efectivo y eficiente para MAS con retrasos constantes. Su diseño basado en eventos y predicción lo convierte en una herramienta útil para sistemas distribuidos con limitaciones en recursos de comunicación. Entre las posibles líneas de investigación futuras, se plantea extender el protocolo a escenarios con retrasos variables, a sistemas no lineales y a estrategias de activación adaptativas.

V. Conclusión

La coordinación efectiva en MAS distribuidos depende en gran medida de los protocolos de comunicación que rigen la interacción entre los agentes. A lo largo de este trabajo, se han revisado distintos enfoques que abordan este desafío desde perspectivas complementarias. En primer lugar, se muestra que el uso de ontologías basadas en FIPA-ACL y OWL 2 DL proporciona una base semántica común, lo que facilita la interoperabilidad y la integración de agentes provenientes de diferentes dominios y con capacidades diversas. Este enfoque enriquece la comunicación al permitir que los agentes

comprendan y procesen el significado del contenido de forma uniforme.

En segundo lugar, se realizó un análisis comparativo de varios protocolos de comunicación, abarcando desde lenguajes específicos como FIPA-ACL y KQML hasta paradigmas de publicación-suscripción (MQTT y AMQP), modelos de solicitud-respuesta (HTTP/REST y CoAP) y estrategias de seguridad (TLS y soluciones basadas en blockchain). Este análisis permitió identificar las fortalezas y limitaciones de cada protocolo en función de factores como complejidad, eficiencia, consumo de recursos y robustez, ofreciendo una guía valiosa para la selección del protocolo más adecuado según las necesidades y restricciones de cada sistema.

Finalmente, se analizó la propuesta de un protocolo distribuido de consenso externo que integra un mecanismo de activación por eventos junto con una estrategia de predicción recursiva para compensar los retrasos en la comunicación. Este enfoque demuestra que es posible alcanzar un consenso definido por una referencia externa, incluso cuando los agentes no reciben actualizaciones de forma continua, reduciendo la carga de transmisión. Esto resulta especialmente relevante en entornos con limitaciones de ancho de banda, energía o capacidad de procesamiento.

En conjunto, estos trabajos evidencian que la solución óptima en MAS depende del contexto y de los objetivos específicos del sistema. La integración de técnicas semánticas, análisis comparativos y mecanismos adaptativos abre nuevas vías para diseñar protocolos robustos, escalables y eficientes, sirviendo de base para futuras investigaciones y aplicaciones en entornos distribuidos modernos.

REFERENCES

[1] M. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley, 2nd ed., 2009.

[2] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley, 2007.

[3] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[4] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus," *Automatica*, vol. 66, pp. 1–10, 2016.

[5] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[6] K. Gan, C. On, P. Anthony, and A. Hamdan, *A FIPA-ACL Ontology in Enhancing Interoperability Multi-agent Communication*, pp. 151–160. 02 2018.

[7] B. Drasko and K. Rakic, *Analysis of Communication Protocols in Multi-Agent Systems*, pp. 103–116. 01 2024.

[8] Q. Lei, J.-H. Li, G.-P. Liu, and M. Wu, "External consensus protocol of networked multiagent systems with communication delays under event-triggered mechanisms," *IEEE Access*, vol. 7, pp. 25053–25062, 2019.