

Um Protocolo para Comunicação entre Sistemas Multi-Agentes Embarcados

Nilson Mori Lazarin¹, Carlos Eduardo Pantoja¹, Vinicius Souza Jesus²

¹Centro Federal de Educação Tecnológica (CEFET/RJ)
– Rio de Janeiro – RJ – Brazil

²Universidade Federal Fluminense (UFF)
– Rio de Janeiro – RJ – Brazil

{nilson.lazarin,pantoja}@cefet-rj.br, souza.vdj@gmail.com

Resumo. *As opções de comunicação entre SMA baseados em BDI embarcados em plataformas robóticas atualmente são através de comunicação serial para sistemas fechados ou através de integração com redes de para sistemas abertos. Este trabalho apresenta um protocolo de comunicação não orientado à conexão que possibilita comunicação unicast, multicast ou broadcast em uma rede de difusão através de dispositivos radiofrequência (RF), possibilitando a comunicação SMA Embarcado em situação de emergência ou ambientes sem infraestrutura de comunicação prévia.*

Abstract. *The communication options between BDI-based SMA embedded in robotic platforms today are through serial communication for closed systems or integration with open systems networks. This work presents a non-connection-oriented communication protocol that enables unicast, multicast, or broadcast communication in a diffused network through radio frequency devices, enabling Embedded SMA communication in an emergency or environment without prior communication infrastructure.*

1. Introdução

Agentes Inteligentes são entidades autônomas e pro-ativas construídas em software ou hardware capazes de interagirem entre si através da troca de mensagens. Um Sistema Multi-agentes (SMA) é um conjunto de agentes com os objetivos comuns ou conflitantes sob determinada área de um ambiente real ou simulado [Wooldridge 2009]. Já um SMA Embarcado [Souza de Castro et al. 2020] é um conjunto de agentes em um SMA atuando em prol do gerenciamento de um dispositivo físico onde estão inseridos, interfaceando sensores e controlando atuadores. Um SMA — Embarcado ou não — pode ser Fechado, quando seus agentes não podem se mover para outros sistemas ou plataformas, ou Aberto quando é permitido a saída e entrada de agentes no sistema. Esta mesma classificação pode ser aplicada para a comunicação: se agentes em um SMA podem se comunicar com agentes de outros SMA, utilizando alguma infraestrutura de comunicação, é dito que a comunicação é aberta, e quando os agentes só podem se relacionar com agentes do seu mesmo sistema, a comunicação é dita Fechada. Dessa forma, podemos ter SMA totalmente fechados, que não permite a comunicação externa com outros agentes de outros SMA nem a existência de agentes móveis na sua composição, assim como a existência de SMA totalmente abertos, permitindo o inverso [Pantoja 2019].

Existem diversos domínios que exploram a aplicação de agentes e SMA embarcados para possibilitar o tratamento da informação e uma resposta pro-ativa na ponta de dispositivos individuais, em redes de sensores, redes veiculares e redes IoT [Jesus et al. 2018], [Pantelimon et al. 2019], [Venglář et al. 2020]. Contudo, alguns destes domínios possuem dependências de infraestrutura de rede comunicação (i.e., TCP/IP, middleware IoT, redes 3G), o que pode inviabilizar parcialmente a funcionalidade de SMA embarcados tornando-os totalmente fechados, seja momentaneamente ou permanentemente.

Por exemplo, em um cenário onde carros autônomos — embarcados com SMA — estão em uma rodovia distante sem infraestrutura de telecomunicação, onde ocorra um acidente em determinado local desta rodovia, estes veículos não seriam capazes de se comunicar, a não ser que estejam dotados de atuadores próprios que permitam a comunicação a partir de difusão de dados. Ainda sim, mesmo com a possibilidade de envio dos dados por difusão, estes poderiam ser recebidos por todos os veículos. Caso existam nessa rodovia veículos dos tipos civis e de segurança pública (polícia e bombeiro), é interessante que eles possam se comunicar individualmente (*unicast*), em grupos (*multicast*) ou por difusão (*broadcast*).

Atualmente, poucos trabalhos exploram a embarcação de SMA em plataformas robóticas, especialmente quando se trata do modelo Belief-Desire-Intention (BDI) [Bratman 1987], visto que o processo de percepção, processamento de crenças e intenções podem gerar atrasos indesejados se não tratados adequadamente [Stabile et al. 2018]. Alguns frameworks e soluções atuais para este fim permitem a integração de SMA Embarcados totalmente fechados através de interface serial [Lazarin and Pantoja 2015], [Pantoja et al. 2016], ou integração com redes de dados [Pantoja et al. 2018], [Manoel et al. 2020]. Contudo, ou tais soluções são SMA totalmente fechados ou dependentes de infraestrutura de comunicação.

O objetivo deste trabalho é propor um protocolo de comunicação para a criação de SMA embarcados utilizando agentes BDI de forma que seja possível a difusão de informação/comunicação para outros SMA embarcados em distâncias reduzidas. O protocolo poderá ser utilizado em conjunto com outras infraestruturas de comunicação (3G, IoT, etc.) ou quando não existir outros métodos disponíveis para comunicação, permitindo assim que um SMA totalmente Fechado possa transmitir e enviar informações a outros dispositivos.

Para isso, a interface serial Javino [Lazarin and Pantoja 2015], será estendida para permitir comunicação ponta a ponta usando Radio Frequência (RF) através de um protocolo para envio de mensagens de *broadcast*, *multicast* e *unicast*. Também será utilizado o JaCaMo *framework* [Boissier et al. 2013] para a criação do SMA e agentes Argo [Pantoja et al. 2016] para permitir a criação de SMA embarcados com capacidade de interfaceamento de *hardware*. Será apresentado uma prova de conceito com três protótipos de carros para cada um dos modos de comunicação. Este trabalho está organizado da seguinte forma, na Seção 2 os detalhes do protocolo proposto e de sua implementação serão mostrados; a prova de conceito será demonstrada na Seção 3. Por fim, nas Seções 4 e 5, serão apresentados os Trabalhos Relacionados e a Conclusão respectivamente.

2. O Protocolo de Comunicação de SMA Embarcados

Em dispositivos com sistemas embarcados, incluindo SMA, a comunicação é ponto importante pois permite que dados obtidos a partir da leitura sensores possam ser encaminhados para outros dispositivos. Da mesma forma, através de um meio de comunicação, dados também podem ser recebidos de outros dispositivos. Caso contrário, qualquer sistema embarcado, mesmo que dotado de capacidade cognitiva e autônomo, dependeria apenas de suas próprias observações e conclusões para a tomada de decisões e aprendizado. Ao se estabelecer um meio de comunicação, o aprendizado e a tomada de decisão poderá também considerar informações obtidas de um coletivo de sistemas embarcados. Um SMA Embarcado é responsável pelo sensoriamento de um ambiente real através de um conjunto de sensores, pelo processo de análise dos dados capturados e deliberação, e por agir no ambiente usando seus atuadores, além da comunicação com outros dispositivos. Para isso, é necessário que estes dispositivos adotem uma arquitetura de interligação de seus diversos componentes físicos e de fluxo de informação [Matarić 2007].

Neste trabalho, será adotada uma arquitetura física e lógica que permite interligação de componentes de *hardware* para que seja possível habilitar um fluxo de percepções direta de sensores para agentes em um SMA cognitivo [Pantoja et al. 2016]. Esta arquitetura será modificada fisicamente para permitir a comunicação com outros dispositivos com SMA embarcados e logicamente para permitir que mensagens e percepções cheguem a estes SMA. A arquitetura tem quatro camadas lógicas:

- a) **Reasoning:** É a camada responsável pela cognição do dispositivo. Um SMA é adotado para interfacear os sensores e atuadores disponíveis através de uma interface serial recebendo percepções ou mensagens de outros dispositivos, deliberando e respondendo com ações a serem executadas nos atuadores ou mensagens para outros dispositivos.
- b) **Serial:** É a camada que faz interfaceamento entre os dados que trafegam entre o SMA e o hardware adotado. A comunicação se dá através de um protocolo de troca de informações entre a porta serial de uma plataforma microprocessada (i.e., computadores e placas) e microcontroladores. O protocolo de comunicação entre SMA embarcados é implementada nessa camada, onde as mensagens serão captadas e enviadas aos demais dispositivos através de acesso ao hardware de RF disponível.
- c) **Firmware:** É o programa embarcado no microcontrolador onde os comportamentos de acesso aos sensores e atuadores são programados. As percepções são enviadas ao agente que faz o interfaceamento uma vez por ciclo ou quando este a requisita. As ações para controle de atuadores e de mensagens são enviadas de acordo com a necessidade do agente.
- d) **Hardware:** São todos os sensores e atuadores disponíveis no dispositivo e que estão conectados ao microcontrolador escolhido. Nessa camada inclui-se o sensor de RF para a comunicação com os demais dispositivos. São estes sensores e atuadores que produzem as informações que serão transformadas em percepções e recebem as mensagens que serão enviadas ao SMA e outros dispositivos.

Fisicamente, este trabalho embarca o SMA em computadores microprocessados, podendo ser desde um mini computador (i.e., Raspberry) até um computador tradicional, desde que sejam capazes de manter um sistema operacional. Os sensores e atuadores são conectados a uma plataforma microcontrolada (i.e., Arduino, ATMEGA) que utilizará a comunicação serial entre o sistema operacional e o microcontrolador através de um

middleware ou interface serial (i.e., Javino) para a troca de mensagens e informações. Para que o protocolo proposto possa ser utilizado na prática é preciso que o dispositivo com o SMA Embarcado seja construído a partir de uma arquitetura conforme a especificada acima e que adote atuadores RF em sua composição para a comunicação. A arquitetura pode ser vista na Figura 1.

Apesar da arquitetura especificar um determinado microcontrolador na camada do *firmware*, ao adotar o Javino como interface serial, é possível manter a independência entre as tecnologias utilizadas em alto nível das camadas de *firmware* e *hardware*, podendo o microcontrolador ser trocado sem afetar o processamento em alto nível, assim como a linguagem do SMA. Para isso, o projetista deverá disponibilizar para o microcontrolador selecionado, a implementação adequada do protocolo proposto neste trabalho e utilizar alguma linguagem com suporte ao Javino.

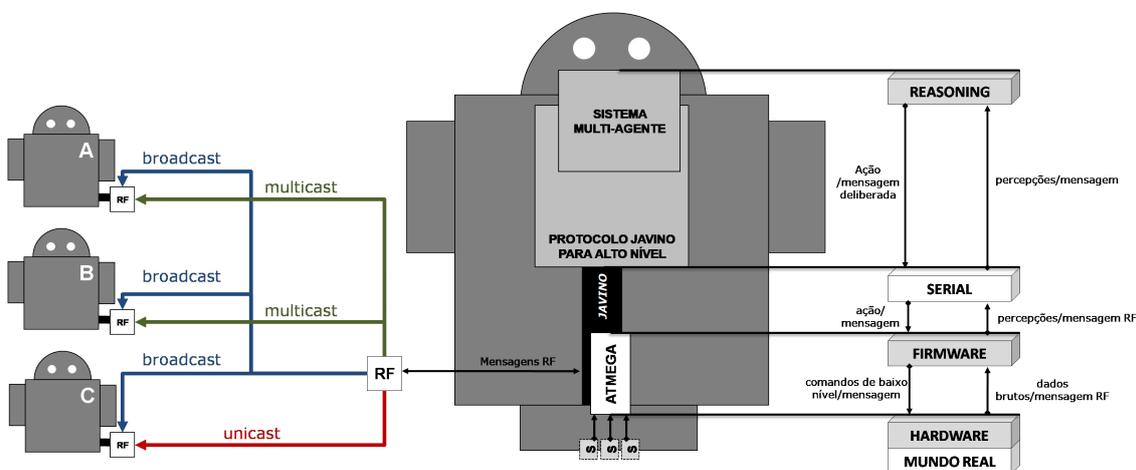


Figura 1. A arquitetura física e lógica para um SMA Embarcado considerando o protocolo proposto.

Quando um dispositivo envia uma mensagem para o outro dispositivo, um agente do SMA que está enviando a mensagem, transfere o conteúdo da mensagem através da interface serial usando o Javino em linguagem de alto nível, que redireciona para o microcontrolador através da porta serial. A mensagem é recebida pelo Javino em baixo nível, que envia a mensagem com as informações de destino. Uma vez que o dispositivo de destino recebe uma mensagem endereçada a ele captada pelo RF, essa mensagem é tratada pelo Javino e enviada a um agente do SMA através da porta serial. O agente ficará a cargo de tratar a mensagem uma vez que esta chegar.

2.1. Comunicação Ponta a Ponta

O protocolo apresentado neste trabalho visa permitir a comunicação entre SMA Embarcados através de transmissor e receptor de radiofrequência (RF) para ambientes sem infraestrutura de comunicação. A decisão de envio de uma mensagem parte da deliberação de um agente pertencente ao SMA de origem, que encaminha essa mensagem para o hardware que controla o RF, responsável pela difusão na rede.

O protocolo possibilita a troca de mensagens *Unicast* e *Multicast* na rede de difusão, dessa forma, um SMA embarcado poderá enviar informações para outro individu-

almente, para um grupo ou para todos os no alcance de transmissão, utilizando um dos três modos de envio:

- **Unicast**: mensagem enviada de um dispositivo para um outro dispositivo específico dentro do alcance de transmissão.
- **Multicast**: mensagem enviada de um dispositivo a qualquer outro inscrito em determinado grupo de dispositivos dentro do alcance de transmissão.
- **Broadcast**: mensagem enviada de um dispositivo a qualquer outro no alcance da transmissão.

O hardware controlador de RF nos dispositivos no alcance da difusão irá analisar a mensagem recebida, conforme definido na Figura 2 e armazenar no *buffer*. Uma vez que for recebida uma transmissão, o cabeçalho da mensagem será analisado. Caso seja uma mensagem válida (formato esperado pelo protocolo), será verificado o endereço de destino. São aceitas mensagens do tipo: *Broadcast*; Mensagens *multicast* destinadas a um dos grupos em que o dispositivo esteja inscrito; Ou mensagem *unicast* destinada a uma de suas identificações na rede. Por fim, somente será armazenada no buffer a mensagem que for considerada completa, através da comparação do valor no campo tamanho da mensagem, informado no cabeçalho, e o tamanho da mensagem recebida.

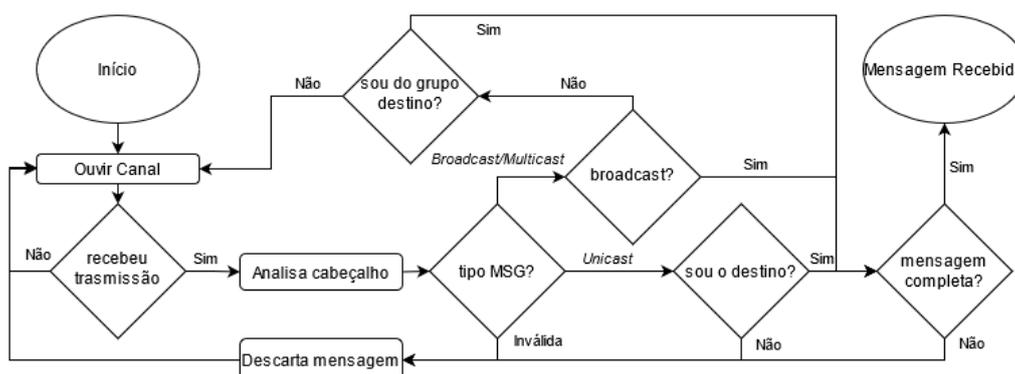


Figura 2. Recebimento de mensagem no meio de difusão.

O protocolo proposto utiliza endereçamento de 24 bits. Os 12 bits mais significativos identificam o grupo e os 12 bits menos significativos representam o membro do grupo. O formato do endereço é apresentado na Figura 3a. O endereço é representado utilizando Alfabeto Base 64 [Josefsson 2006], dessa forma podem ser utilizados caracteres de A à Z, maiúsculo e minúsculo e os caracteres + e /.

É possível identificar até 16.252.928 dispositivos em 3968 grupos de até 4096 membros cada, iniciado em AAAA até 9///. Os endereços especiais //AA até ///+ são utilizados para comunicação *multicast*, um endereço *multicast* inicia com // seguido de dois caracteres que representam o grupo destinatário da comunicação. O endereço especial /// indica uma comunicação *broadcast*. O endereço especial ++++ indica uma comunicação serial para acesso a hardware. A faixa de endereçamento é apresentada na Tabela 1.

O cabeçalho da mensagem do protocolo proposto possui 3 campos. Endereço de destino com 24 bits, representando um endereço de dispositivo em Alfabeto Base 64; Endereço de origem com 24 bits, representando um endereço de dispositivo em Alfabeto

FAIXA DE ENDEREÇOS										
Endereçável	A	A	A	A	até	9	/	/	/	3968 grupos [AA até 9/] 4096 membros por grupo
Reservado	+	A	A	A	até	+	+	+	9	para uso futuro
Especial	+	+	+	+	comunicação serial					acesso ao hardware
Reservado	+	+	+	/	até	/	+	/	/	para uso futuro
Multicast	/	/	A	A	até	/	/	/	+	formato //[Grupo]
Especial	/	/	/	/	comunicação <i>broadcast</i> na rede de difusão					

Tabela 1. Faixa de endereços

Base 64; e Tamanho da mensagem com 12 bits, representado em Alfabeto Base 64, indicando o tamanho do campo Mensagem em bits. A Figura 3b apresenta uma mensagem *broadcast* originada pelo membro FE do grupo CA.

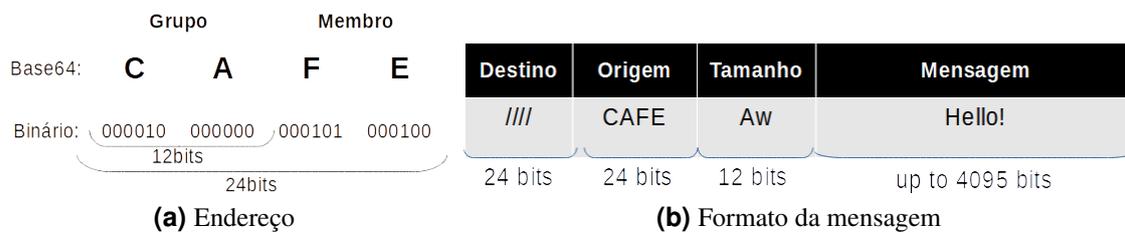


Figura 3. Endereço e formato da mensagem

2.2. Comunicação Agente-Controlador

Para que um agente envie uma mensagem de um SMA embarcado em um dispositivo para outro agente em outro SMA embarcado, é preciso que seja informado três campos obrigatórios: o SMA de destino, a força ilocucionária e a mensagem em si. O destinatário será composto pelo endereço de *hardware* do SMA em base 64 bits. Contudo, é necessário endereçar a mensagem a um agente específico para que a mensagem seja tratada adequadamente visto que o SMA não é capaz de processar mensagens sem a figura de um agente. Dessa forma, é estabelecido que um SMA possua um Agente Difusor responsável por tratar apenas as mensagens vindas de RF. Este agente poderá ter qualquer nome, mas terá anotada em sua crença ou intenção recebida o endereço em base 64 do SMA remetente. Neste primeiro momento, apenas serão consideradas as forças ilocucionárias *tell*, *untell*, *achieve*, e *unachieve* por terem um tamanho de mensagem menor do que as *tellHow* e *AskHow*, e dada a limitação do campo da mensagem do protocolo e do meio de difusão, além de não necessitarem de respostas ao remetente.

Um agente Difusor envia uma mensagem fazendo a chamada a uma ação interna chamada de *difuse* criada especificamente para enviar mensagens através de difusão em RF usando o protocolo proposto. Primeiramente, quando esta ação é chamada, o agente codifica a mensagem separando os campos endereço de *hardware* do remetente, a força ilocucionária e a mensagem por ponto-e-vírgula e encaminha através do Javino que adiciona o endereço do destinatário ao preâmbulo e calcula o tamanho total da mensagem. Em baixo nível, no microcontrolador, a mensagem é difundida com o endereço identificado

no preâmbulo uma vez que esta não está endereçada para ação em atuadores ou captura de percepção ou de mensagens (endereço de destino ++++). Quando o *hardware* do SMA de destino recebe a mensagem, ele a armazena em um *buffer* temporário até que o agente Difusor do destinatário solicite as mensagens recebidas. Como agentes BDI possuem um ciclo de raciocínio bem definido, não é possível o microcontrolador enviar diretamente a mensagem ao SMA. É preciso que haja a sincronização entre o ciclo de raciocínio do agente e o recebimento de mensagens.

Para isso, foi criada uma extensão da arquitetura customizada de agentes ARGO para que seja possível se comunicar com o *hardware* através da porta serial para solicitar percepções e mensagens difundidas. Os agentes ARGO fazem uma solicitação de percepções (*getPercepts*) no passo de captura das percepções do ciclo BDI. Igualmente, agentes Difusores farão a solicitação de mensagens (*getMessages*) no passo onde o agente faz a checagem de recebimento de mensagens (*check mail*). Caso o agente tenha alguma mensagem para ser recebida, ele a decodificará, para criar as mensagens entendíveis para o agente. Perceba que todo agente Difusor também é um agente ARGO, mas o contrário não é verdade, ou seja, os agentes Difusores são uma extensão dos agentes ARGO com uma capacidade adicional de se comunicarem com RF. A Figura 4 mostra o ciclo de raciocínio do agente Difusor.

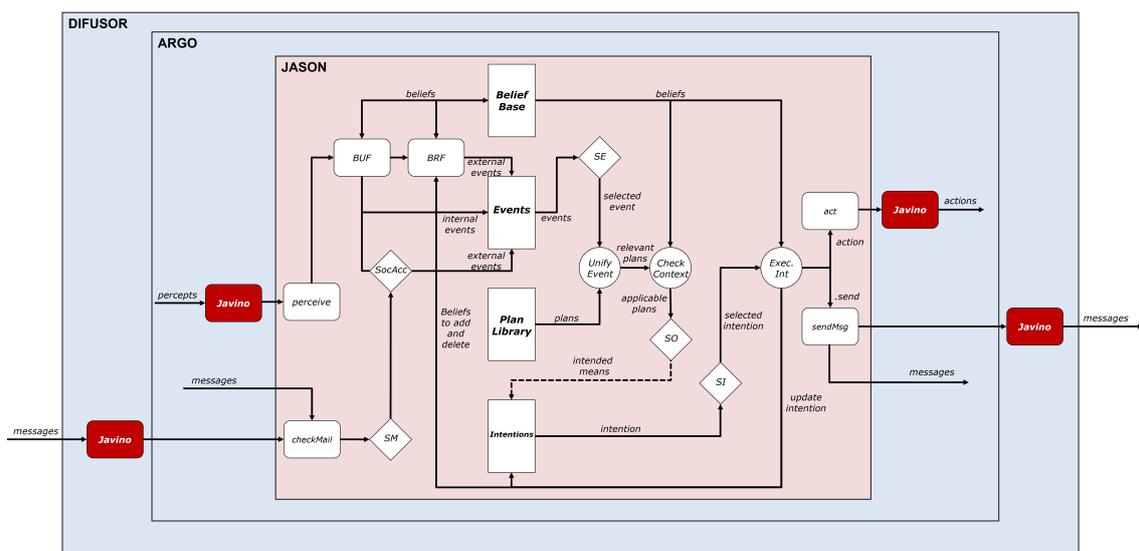


Figura 4. O ciclo de raciocínio de um Agente Difusor.

Para que a abordagem proposta funcione, é necessário que o SMA possua no máximo um agente Difusor para que o endereçamento da mensagem seja centralizado em apenas uma entidade. A existência de mais de um agente Difusor não inviabiliza o funcionamento do SMA, contudo, vai ser gerado uma competição pela aquisição das mensagens, obrigando ao programador e projetista do SMA embarcado a reprodução dos planos nos agentes ou a comunicação caso uma mensagem não seja de responsabilidade de quem a capturou. Além disso, é recomendável que cada SMA adote pelo menos um microcontrolador dedicado para a comunicação RF para atuar separadamente como um suporte caso a infraestrutura de comunicação ou o microcontrolador principal falhem. Novamente, a utilização de um agente Difusor para gerenciar outros atuadores e sensores além do RF não inviabiliza o projeto e não interfere na captura das percepções nem das mensa-

gens, pois estas ocorrem em passos diferentes do ciclo de raciocínio. A recomendação se dá para uma melhor organização de tarefas entre os agentes pertencentes ao SMA. A limitação deste tipo de aplicação centralizada é que caso o agente Difusor seja morto ou vá para outro SMA é necessário ter um mecanismo para preservação de seus conhecimentos [Souza de Jesus. et al. 2021].

3. Prova de Conceito

Para avaliar o protocolo proposto, é apresentada uma prova de conceito considerando um cenário de uma rodovia onde trafegam veículos civis e de segurança pública, como bombeiro e polícia, que são controlados por SMA embarcados sem acesso a nenhuma infraestrutura de comunicação, exceto a comunicação por RF existente em cada veículo. Em caso de algum acidente, o carro que identificar o acidente ou o que sofrer o acidente (caso seus atuadores não tiverem sido afetados) podem enviar uma mensagem (*broadcast*) para os demais veículos na proximidade para que estes fiquem atentos ou simplesmente parem devido ao acidente ocorrido. Além disso, a mensagem pode ser difundida para veículos de segurança pública, que poderão se comunicar com indivíduos do mesmo grupo (*unicast* e *multicast*) ou para demais veículos civis (*unicast* e *broadcast*).

Para isso, foram implementados 4 SMA embarcados usando o Jason e a extensão de agentes ARGO proposta, rodando em 3 *laptops* distintos e 1 em uma Raspberry representando um veículo cada. Além disso, cada um estará vinculado a uma placa microcontrolada com o ATMEGA328 e módulos TX/RX — RF 433Mhz, além de LEDs para sinalização de movimentação. Um dos computadores representará um veículo civil que sofreu um acidente e enviará uma mensagem de *broadcast* solicitando apoio aos demais. Um outro computador e a Raspberry serão veículos de segurança pública em um mesmo grupo que receberão a mensagem de acidente e se comunicarão para alinhar um possível resgate *multicast* e enviarão uma mensagem de *acknowledge* informando que o socorro está a caminho *unicast*. Por fim, o último computador representará um veículo civil que apenas receberá a solicitação de apoio do acidentado porém não fará nada a respeito. A utilização de computadores ou *laptops* apenas adicionam mais poder de processamento ao SMA, não interferindo na comunicação entre o hardware e o software ou com a difusão por RF.

Cada SMA terá apenas um agente ARGO interfaceando o hardware com planos específicos para deliberação dependendo de em qual veículo ele esteja embarcado. No caso do veículo acidentado, o agente enviará uma mensagem de *broadcast* utilizando a nova ação interna criada informando que ele precisa de apoio. No caso do veículo civil que receber essa mensagem, ele apenas informará que a mensagem foi recebida acendendo um LED. Um dos veículos de segurança pública receberá a mensagem de solicitação de apoio e enviará uma mensagem de *multicast* para todos os integrantes do grupo, que informarão estarem de pronto atendimento acendendo um LED. Este mesmo veículo enviará a mensagem de *acknowledge* para o veículo acidentado informando que o apoio já está a caminho. A Tabela 2 exibe a identificação de cada veículo através do protocolo, os planos dos agentes e o comportamento esperado antes e depois da execução.

O execução do sistema retornou o comportamento esperado de todos os SMA embarcados e do protocolo, contudo ainda é preciso aplicar o protocolo em SMA mais robustos e com um nível de complexidade maior, inserindo mais agentes em cada sis-

Veículo	Id	Comportamento Esperado	Planos	Comportamento Obtido
Acidentado	Brok	Solicita apoio	askForHelp(Everyone) wait : help(coming)	LED aceso após mensagem de acknowledge
Bombeiro 1	CBv1	Recebe solicitação de apoio e contata membros do grupo CB	contactTroop: help[Brook]	Mensagem de unicast para o Acidentado e de multicast para o grupo CB enviadas
Bombeiro 2	CBv2	Aguarda contato do Bombeiro 1	wait : accident	LED aceso após mensagem de multicast do Bombeiro 1
Civil	NONE	Nenhum	help(coming)[Brook]	LED aceso após mensagem de broadcast do Acidentado

Tabela 2. Papéis e comportamentos

tema de forma que seja possível medir o custo de processamento mínimo (utilizando o mínimo de componentes da arquitetura possível) e de execução do domínio da aplicação. De toda forma, o resultado esperado foi atingido em um tempo satisfatório considerando se tratar de uma arquitetura alternativa para comunicação dada a indisponibilidade de outros meios, tendo todo os veículos deliberado em menos de 2s. Além disso, é importante embarcar todos os SMA em placas como a Raspberry ou similares para garantir uma análise entre a utilização de computadores e placas microprocessadas. Um critério para a utilização de placas ou computadores pode ser o tamanho do dispositivo em si. Por exemplo, em um carro real é possível utilizar um computador de bordo similar ou mais potente do que um computador com configurações atuais mas, conforme dito anteriormente, a principal diferença se dará em capacidade de processamento para o SMA embarcado, não interferindo na interconexão de *hardware* e *software*.

Alguns pontos relacionados à segurança da transmissão não são abordados no protocolo proposto. A difusão das mensagens trafegam em claro, podendo permitir que mensagens de *unicast* ou *multicast* sejam capturadas por outros veículos não pertencentes a um determinado grupo ou qualquer outro dispositivo que esteja no alcance de transmissão e seja capaz de captar dados de RF. Ou ainda, um agente mal intencionado enviar mensagens forjando sua identidade (se passando por outro dispositivo) durante a transmissão.

Para uma comunicação segura pode-se implementar algum algoritmo de criptografia simétrica no dispositivo responsável pela transmissão RF, dada a facilidade de implementação e menor custo computacional e realizar a troca segura das chaves de transmissão através do algoritmo Diffie-Hellman. Assim, o emissor antes de transmitir a mensagem deverá derivar a chave de transmissão utilizando sua chave privada e a chave pública do destinatário. O receptor, por sua vez, ao receber a mensagem irá derivar a chave de transmissão através de sua chave privada e a chave pública da origem. Dessa forma estará garantido o sigilo, a integridade, a autenticidade e a irretratabilidade da mensagem.

Por fim, existe uma limitação de endereçamento de dispositivos na rede ad-hoc, contudo o objetivo é atuar em áreas menores e não em contextos de cidades inteligentes envolvendo grandes quantidades de dispositivos visto que o protocolo é uma forma de comunicação quando não houver um outro meio de comunicação disponível.

4. Trabalhos Relacionados

Na literatura de SMA embarcados existem diversas aplicações que exploram troca de informações entre SMA distintos. Os agentes *Communicator* [Pantoja et al. 2018] são agentes BDI estendidos capazes de se conectar a uma rede e enviar/receber informações

de outros SMA. Além disso, esses agentes *Communicator* posteriormente foram utilizados para a criação de protocolos bioinspirados para a transferência de agentes entre SMA distintos [Jesus et al. 2018]. Com isso, diferentes SMA não só conseguiam se trocar informações pela rede, mas como também, enviar e receber agentes.

Atualmente, foi percebida um crescente número de aplicações de SMA embarcados em drones. O survey [Pantelimon et al. 2019] reúne diversas aplicações de SMA embarcados em drones buscando identificar nestes sistemas como é feito o gerenciamento do controle de missão e as estratégias de comunicação para implantações desses drones. Os principais resultados foram obtidos através de duas estratégias: centralizada e descentralizada.

Além disso, o trabalho [Venglář et al. 2020] apresenta uma implementação de rede *mesh* para Wi-Fi móvel, considerando que nem todos os dispositivos são habilitados para essa rede Wi-Fi *mesh*. Com isso, foi apresentado um novo método de comunicação para SMA embarcados com resultados que mostram que é possível utilizar esse tipo de rede, ganhando em alcance operacional.

Portanto, todos os trabalhos descritos acima utilizam um tipo de infraestrutura de rede (i.e., TCP/IP, middleware IoT, redes 3G) para a troca de informações. Em caso de falha, os SMA embarcados perdem o canal de comunicação e não possuem nenhuma medida de tratamento. Sendo assim, este trabalho apresenta um protocolo de comunicação secundário que pode ser utilizado, por exemplo, em caso de falha da infraestrutura de rede ou até mesmo para a comunicação de dispositivos em curta distâncias.

5. Conclusão

Este trabalho apresentou um protocolo que permite a comunicação entre dois SMA embarcados através de RF para quando não existe nenhum outro meio de comunicação disponível no momento e local onde estejam os dispositivos. O protocolo prevê três modos de comunicação em nível de *hardware* que auxilia na organização e troca de mensagens em grupos, comunicação individual e envios de mensagens de *broadcast*. Esses modos de comunicação permitem manter um nível mínimo de interação entre os dispositivos e os SMA embarcados para que estes não se tornem totalmente fechados em uma situação onde não tenha disponível uma estrutura de comunicação para IoT, 3G, etc. De toda forma, ainda é uma possibilidade e escolha do projetista, adotar o protocolo mesmo que haja uma estrutura de comunicação disponível, não sendo estas duas conflitantes caso coexistam.

Para permitir a criação do protocolo diversas tecnologias foram utilizadas e estendidas. Primeiramente, por se tratar de comunicação em nível de Hardware, o modo de endereçamento do Javino foi modificado para permitir a criação de grupos e endereçamento individual de dispositivos. Essa extensão utiliza uma base de endereçamento em Base64, diferente da base anterior, contudo não há impacto no funcionamento de soluções que adotaram a versão anterior. Como a modificação foi internamente e transparente ao usuário, foi dedicado um endereço específico (++++) para comunicação entre linguagem de alto nível e hardware.

Da mesma forma, o agente ARGO foi estendido para criar um novo agente chamado Difusor, onde foi adicionado no seu ciclo de raciocínio um passo para captura das

mensagens armazenadas no *buffer* do hardware. A opção de se criar um novo agente se deu para definir bem as responsabilidades dos agentes possíveis de se usar em um SMA embarcado, de forma a não sobrecarregar um ou outro com demais funcionalidades. Um agente Difusor é na prática um agente ARGO com a capacidade de fazer leitura de mensagens. Dessa forma, considerando a aplicabilidade de arquiteturas estendidas de agentes para o JaCaMo, é possível criar agentes de interfaceamento de sensores e atuadores (ARGO), agentes que se comunicam por uma rede IoT (Comunicadores) e agentes de difusão em redes de RF (Difusores).

O protocolo ainda precisa ser testado em cenários mais robustos com uma aplicação mais extensa. Contudo, testes iniciais comprovaram que é possível fazer com que agentes BDI embarcados possam se comunicar como outros agentes embarcados de um SMA distinto de forma efetiva. Como trabalhos futuros, será criado o mecanismo de segurança através de criptografia AES e chaves públicas e privadas, para que as mensagens difundidas no meio não trafeguem em claro. Além disso, novos testes de performance e uma análise mais aprofundada serão apresentadas.

Referências

- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761.
- Bratman, M. (1987). *Intentions, Plans, and Practical Reason*. Harvard University Press.
- Jesus, V., Manoel, F., Pantoja, C. E., and Viterbo, J. (2018). Transporte de agentes cognitivos entre sma distintos inspirado nos princípios de relações ecológicas. In *Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações—XII WESAAC*, pages 179–187.
- Josefsson, S. (2006). The Base16, Base32, and Base64 Data Encodings. RFC 4648.
- Lazarin, N. M. and Pantoja, C. E. (2015). A robotic-agent platform for embedding software agents using raspberry pi and arduino boards. *9th Software Agents, Environments and Applications School*, pages 13–20.
- Manoel, F., Pantoja, C. E., Samyn, L., and de Jesus, V. S. (2020). Physical artifacts for agents in a cyber-physical system: A case study in oil & gas scenario (eeas). In *SEKE*, pages 55–60.
- Matarić, M. (2007). *The Robotics Primer*. Mit Press.
- Pantelimon, G., Tepe, K., Carriveau, R., and Ahmed, S. (2019). Survey of multi-agent communication strategies for information exchange and mission control of drone deployments. *Journal of Intelligent Robotic Systems*, 95:1–10.
- Pantoja, C. E. (2019). *An Architecture To Support the Integration of Embedded Multi-Agent Systems*. PhD thesis, Universidade Federal Fluminense.
- Pantoja, C. E., Jesus, V. S., Manoel, F. C. P. B., and Viterbo, J. (2018). A heterogeneous architecture for integrating multi-agent systems in ami systems. *The Thirtieth International Conference on Software Engineering and Knowledge Engineering (SEKE 2018)*.
- Pantoja, C. E., Stabile, M. F., Lazarin, N. M., and Sichman, J. S. (2016). Argo: An extended jason architecture that facilitates embedded robotic agents programming. In

- International Workshop on Engineering Multi-Agent Systems*, pages 136–155. Springer.
- Souza de Castro, L., Manoel, F., Souza de Jesus, V., Pantoja, C., Borges, A., and Vaz Alves, G. (2020). Integrando sistemas multi-agentes embarcados, simulação urbana e aplicações de iot. In *XIV Workshop Escola de Sistemas de Agentes, seus Ambientes e Aplicações (WESAAC 2020)*.
- Souza de Jesus., V., Pantoja., C., Manoel., F., Alves., G., Viterbo., J., and Bezerra., E. (2021). Bio-inspired protocols for embodied multi-agent systems. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART*, pages 312–320. INSTICC, SciTePress.
- Stabile, M. F., Pantoja, C. E., and Sichman, J. S. (2018). Experimental analysis of the effect of filtering perceptions in bdi agents. *International Journal of Agent-Oriented Software Engineering*, 6(3-4):329–368.
- Venglář, V., Králík, J., Chin, H.-L., and Chen, K.-S. (2020). Mesh wi-fi infrastructure for multi-agent robotic system. In *2020 19th International Conference on Mechatronics - Mechatronika (ME)*, pages 1–4.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.