Estudo da viabilidade do ROS como plataforma para IoT

Vinícius Alves Hax, Nelson Lopes Duarte Filho, Silvia Silva da Costa Botelho, Odorico Machado Mendizabal
Centro de Ciências Computacionais
Universidade Federal do Rio Grande – FURG
Campus Carreiros: Av. Itália km 8 Bairro Carreiros, Rio Grande, Brasil
viniciushax@furg.br, dmtnldf@yahoo.com.br, {silviacb, odoricomendizabal}@furg.br

***Resumo*—O artigo apresenta um estudo a respeito da viabilidade em utilizar o *framework Robot Operating System* como base para um *middleware* de *Internet of Things*. São apresentados os conceitos de IoT e do ROS, e logo após, apresentados critérios de avaliação. Por fim, casos de uso hipotéticos de aplicação são analisados com base nas características existentes do ROS e nas premissas de um ambiente de IoT.**

I. Introdução

Esse artigo descreve um estudo de viabilidade da plataforma *Robot Operating System* (ROS) como base para desenvolvimento de uma infraestrutura voltada para o ambiente de *Internet of Things* (IoT). Devido ao crescente interesse em IoT torna-se importante prover ambientes e ferramentas que facilitem o desenvolvimento de novas soluções. Por se tratar de uma área nova, ainda não estão estabelecidas as ferramentas necessárias para implementação dessas soluções. Nesse contexto, surge a possibilidade de utilizar a plataforma ROS como base para uma solução de infraestrutura de IoT. A sua utilização é interessante pois fornece uma abstração que permite acessar hardwares externos como sensores e atuadores de forma facilitada e a sua implementação foi organizada de forma modular, facilitando a adição de novas funcionalidades. Critérios de avaliação de aplicações IoT são analisados no contexto do ROS para saber se o mesmo pode ser utilizado como base para um *middleware* de IoT. Na seção II é apresentado o conceito de IoT e alguns desafios inerentes desse paradigma. Na seção III é apresentada a plataforma ROS. A seção IV define os critérios utilizados na análise de viabilidade do ROS no contexto de IoT. A seção V analisa o ROS segundo um conjunto de critérios e por fim a seção VI apresenta as conclusões do estudo.

II. Internet of Things

Recentemente surge um novo paradigma tecnológico denominado Internet of Things (IoT), ou Internet das Coisas. O conceito começa a ser explorado cientificamente bem como desperta o interesse de empresas como Verizon, AT&T e Cisco[1] e agências de origem governamental como o Information Society and Media Departament da União Europeia [2]. Também a IoT foi considerada como uma das seis tecnologias civis com maior potencial de impacto nos Estados Unidos[3].

A Internet of Things pode ser definida como "uma infraestrutura de rede global e dinâmica com capacidade de autoconfiguração baseada em padrões e protocolos onde objetos virtuais e reais tem identidades, atributos, personalidade, que usam interfaces inteligentes e são integradas em uma rede de informação" [2]. Esses objetos podem interagir entre si e com o ambiente através de sensores e atuadores diversos. O impacto tecnológico deste novo paradigma computacional pode abranger diferentes áreas de aplicações, tais como setor automotivo, aeroespacial, edifícios inteligentes, sistemas de auxílio à saúde, logística, manufatura, entre outras.

A. Hardware

O *Internet Engineering Task Force* (IETF) [1], instituição responsável por diversos padrões para equipamentos de intercomunicação, criou em 2005 um grupo denominado *IPv6 Over Low Power Wireless Personal Network*, com o objetivo de padronizar a utilização dos protocolos de rede disponíveis em redes de baixa potência sem fio de alcance pessoal. Nos documentos produzidos pelo grupo, [4], caracteriza os equipamentos que compõe essas redes como:

- Tendo pequeno tamanho de pacotes de dados, em torno de 81 bytes.
- Suportando 16 bits ou 64 bits padrão IEEE estendido para endereços na camada de acesso ao meio.
- Baixa largura de banda: Taxas de 250, 40 e 20 kbps.
- Topologia em estrela e em malha.
- Baixa potência. Em muitas ocasiões são dispositivos que utilizam bateria.
- Baixo custo. Geralmente estão associados com sensores e possuem baixa capacidade de processamento e pouca memória.
- Alto número de dispositivos instalados.
- Localização dos dispositivos não é pré-definida, pois são instalados à medida que se tornam necessários.

Além disso os dispositivos integrantes dessas redes são de dois tipos:

- Dispositivos de funcionalidade reduzida, que possuem limitações maiores de processamento e memória e geralmente não roteiam pacotes
- Dispositivos de funcionalidade completa, com maior capacidade do que os anteriores, geralmente atuando como roteadores para os nodos de hardware mais limitado

[1] http://www.ietf.org/

B. Utilização do protocolo IP em IoT

Na medida em que se faz necessário escolher um protocolo de rede para o funcionamento da IoT, uma opção à ser considerada é utilizar o próprio protocolo IP que se encontra atualmente em utilização. Como vantagens podemos citar que o protocolo IP já está amplamente disseminado e conta com diversos softwares de diagnóstico, configuração e resolução de problemas. A especificação aberta do protocolo também é favorável na sua adoção. Alguns dos problemas do IPv4 são resolvidos no IPv6, como o suporte ao grande número de dispositivos e incorporação de mecanismos de autoconfiguração de rede. Essas características o tornam um protocolo propício para IoT.

III. Robot Operating System

A plataforma ROS surgiu baseada na dificuldade de integrar soluções disponíveis na área de robótica. Diferentes tipos de robôs possuem hardware muito variado, dificultando o processo de escrita de software, que tradicionalmente precisa ser escrito especificamente para cada hardware. Além disso os hardwares utilizados possuem propósitos e implementações diversas, incluindo atuadores no formato de rodas até sensores visuais [5]. Segundo [6] a necessidade de profundos conhecimentos específicos necessários para cada pesquisador muitas vezes estão além do que é factível. Com o objetivo de facilitar o desenvolvimento na área de robótica surgiu o ROS.

Um dos critérios no desenvolvimento da plataforma foi utilizar uma comunicação *peer-to-peer*, ou seja, uma comunicação não centralizada, distribuída entre os nodos. Em um cenário em que múltiplos nodos registram informações através de sensores e os transmitem para nodos de processamento, torna-se indesejável uma comunicação centralizada pois esta logo sofreria problemas no tocante à escalabilidade e apresentaria degradação de desempenho. Apesar disso, o ROS possui uma centralização no serviço de localização de recursos. Um dos nós, denominado *master*, é responsável por informar quais os recursos oferecidos, e quem é responsável pelo mesmo. Após essa etapa de descoberta de serviços, a comunicação independe do nó *master*.

A Figura 1 mostra como se dá o processo inicial de comunicação entre nós do ROS. O nó, intitulado "hokuyo", oferece um serviço intitulado "scan". Inicialmente esse nó, anuncia ao nó *master* que está oferecendo o serviço, e onde o serviço pode ser acessado. Um segundo nó, intitulado "viewer", pergunta ao *master* quem está oferecendo o serviço e recebe como resposta informações sobre o *host* e a porta onde o serviço pode ser acessado. Após, "viewer"entra em contato diretamente com "hokuyo"e negocia uma conexão direta que poderá ser feita por TCP conforme mostra a figura ou usando UDP. Dessa maneira o *master* só influencia no momento inicial da comunicação, a partir desse momento ele não exerce influencia nas comunicações já estabelecidas.

A implementação do ROS busca ser compatível com várias linguagens de programação, permitindo que cada

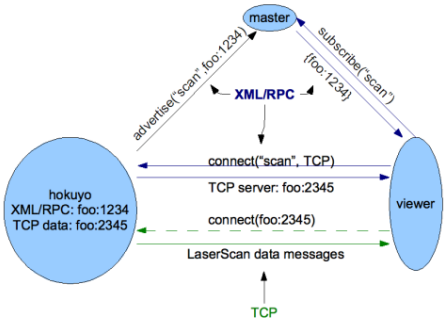

Figura 1. Comunicação no ROS

equipe possa escolher a linguagem mais conveniente. A comunicação inicial da configuração se dá através de XML-RPC, disponível para um grande número de linguagens. Além disso, também existe uma linguagem comum de definição de dados que é simples, porém permite composição através de elementos mais básicos.

Para lidar com a complexidade inerente do ROS, são usados pequenos módulos que se comunicam entre si para realizar tarefas maiores. Somente uma pequena parte da plataforma é implementada de forma centralizada. Além disso, o ROS encoraja o encapsulamento de código, de maneira a reutilizar funcionalidades e algoritmos entre diversos projetos. O próprio ROS reutiliza código de outros projetos, como [7] e [8].

IV. Critérios utilizados

O IETF sugere alguns critérios para projeto de aplicações no contexto de IoT.

- Instalação: De que maneira se dará a instalação dos dispositivos.
- Tamanho da rede: Principalmente no número de dispositivos do grupo.
- Alimentação dos nós: Difere se os nós vão estar ligados na alimentação tradicional ou através de baterias.
- Conectividade: Informa se os elementos da rede estarão conectados de forma permanente, eventual ou de forma periódica.
- Comunicação: Se a comunicação com o restante da rede será através de um ponto de conexão único ou de múltiplos elementos.
- Padrões de comunicação: Ponto para multiponto, multiponto para ponto, ponto à ponto, entre outras possibilidades.
- Nível de segurança: Qual o nível de segurança necessário na aplicação.
- Mobilidade: Se os nós serão móveis ou se permanecerão fixos.
- Qualidade de Serviço: Além das preocupações tradicionais pertinentes à Qualidade de Serviço, é necessário considerar as necessidades específicas dos grupos de dispositivos, de maneira a minimizar a comunicação e a utilização dos recursos computacionais. Além disso os requisitos podem mudar de acordo com o

modelo de entrega de dados.

Em [9] é apresentado o caso de um hospital onde é necessário manter sob controle a temperatura e a umidade de bolsas de sangue dentro de um hospital. Essa variável precisa ser acompanhada desde o momento da coleta até a sua posterior utilização através de sensores nas bolsas de sangue, no veículo de transporte das mesmas e nas salas onde elas são armazenadas. Referente à instalação, ela é feita de forma manual e pré-planejada. Embora possam haver mudanças no processo, elas não ocorrem com frequência e são planejadas previamente. O tamanho é considerado médio, e varia de acordo com o hospital analisado. A alimentação dos nós se daria na maior parte do tempo através de bateria, e a conectividade precisa ser permanente de forma a permitir acompanhar as informações dos sensores durante todo o processo. A comunicação precisaria se dar através de múltiplos elementos para diminuir a ocorrência de falhas. O padrão de comunicação poderia ser ponto à ponto no envio de sinais de controle dos atuadores de temperatura e umidade e múltiplos pontos à ponto para coletar informações de forma agregada. O nível de segurança é alto: as informações precisam obedecer os critérios de disponibilidade, integridade evitando danos aos pacientes. A confidencialidade pode ser um requisito se estiverem sendo transmitidas informações sobre a bolsa tais como doenças do paciente. Por fim a qualidade de serviço precisa ser levada em consideração, pois os sensores vão transmitir informação que vão garantir que os atuadores sejam ativados rapidamente sempre que necessário ou que alarmes sejam ativados para alertar sobre a necessidade de interferência humana.

V. Avaliação da plataforma ROS

A partir dos critérios definidos anteriormente e, utilizando dois cenários hipotéticos, analisamos a viabilidade de utilizar o ROS plataforma base para aplicações de IoT.

A. Cenário 1 - Proteção à saúde

Supondo o cenário de um paciente de hospital que é liberado para ir para casa, porém necessita ficar sob observação, mas mora sozinho. São utilizados sensores para monitorar a temperatura e frequência cardíaca do mesmo, e a informação obtida dos sensores é transmitida para o hospital onde a situação do paciente é avaliada periodicamente, e em função das variações apresentadas um alarme é disparado.

Com relação à instalação, os sensores teriam uma instalação feita de forma manual e planejada de acordo com a casa do paciente. A rede teria um tamanho pequeno se considerado um único paciente. A alimentação dos sensores se daria por baterias, e com exceção da troca das mesmas os sensores estariam permanentemente conectados. O ROS permitiria que o sensor ficasse desconectado para troca de baterias.

Dada a criticidade da aplicação seria recomendável que algum tipo de procedimento diferenciasse situações de falta de energia e falha na comunicação, mas isso poderia ser programado na aplicação. Possivelmente haverá um ponto de conexão centralizador da conexão na casa, mas eventualmente poderia ser configurado uma ligação dupla de comunicação na qual os sensores enviariam as informações por múltiplas rotas de saída. Como o ROS permite múltiplas conexões simultâneas isso não seria um problema.

O ROS não oferece intrinsecamente soluções de segurança, especialmente de forma a garantir a integridade e autenticidade dos dados, pois poderia haver um terceiro elemento escutando as comunicações, ou até mesmo forjando dados dos sensores. Seria necessário embutir aspectos de segurança na API do ROS como por exemplo a capacidade de criptografar os dados enviados ou de verificar a identidade dos elementos envolvidos na comunicação.

A mobilidade não é um problema dentro da mesma rede. A partir do momento em que um nó mude de rede, ele perde a comunicação com os elementos da rede anterior. Caso fosse possível ter um ROS *master* em cada subrede eles poderiam sincronizar as listagens atualizadas de serviços e tópicos existentes entre os *masters*. Essa sincronização precisaria também ser implementada no ROS pois atualmente ele funciona através de um único *master*.

Por fim, atualmente o ROS não trabalha com aspectos de qualidade de serviço, sendo que cada nó é responsável por gerenciar a sua própria comunicação. Devido a isso, não seria possível oferecer QoS com as primitivas existentes atualmente, porém novas primitivas poderiam ser oferecidas levando em consideração estes aspectos. Dentro do contexto da aplicação poderia ser criada a figura de um nó concentrador, inexistente na arquitetura do ROS, responsável por compactar mensagens semelhantes, reduzindo a largura de banda necessária. Para que essa solução pudesse ser escalável seria necessário que os nós concentradores não fossem fixos, mas que elementos da rede pudessem assumir esse padrão de comportamento quando necessário.

B. Cenário 2 - Monitoramento de agricultura

Em uma determinada plantação, uma quantidade grande de nós são instalados manualmente, onde cada um possui sensores diversos, como umidade, temperatura, condição do solo, luz do sol. Ao longo da plantação estão instalados nós com mais capacidade que agregam os dados do demais sensores.

Em relação à instalação, ela seria pré-planejada de acordo com a geografia do terreno. O tamanho da rede é grande devido à necessidade de sensores. A alimentação dos sensores seria feita através de bateria. Com relação à conectividade o ROS ofereceria pleno suporte no cenário, pois os sensores poderiam manter comunicação de dados entre eles. Como a comunicação não é feita de forma centralizada poderia haver múltiplos fluxos de informação simultâneos sem possuir um único gargalo de informação. O ROS permite comunicação entre múltiplos elementos, então os sensores poderiam enviar suas informações para múltiplos agregadores de conteúdo. A segurança não seria

tão essencial nessa aplicação. O fluxo de informação poderia ser enviado dos agregadores para os nós e o mesmo no sentido contrário. A mobilidade não seria levada em consideração pois os nós seriam fixos. A qualidade de serviço não seria relevante nessa aplicação para priorizar mensagens pois todas as informações seriam igualmente importantes, porém também nessa situação seria aconselhável na aplicação que houvesse uma compactação dos dados informados, pois provavelmente haveria grande redundância de informação dentro do grupo de sensores.

VI. Conclusões e trabalhos relacionados

Existem algumas propostas de *middleware* para IoT. Em [10] e [11] são apresentadas propostas da criação de uma camada de software utilizando uma arquitetura orientada a serviços. O artigo [12] utiliza o ROS como *middleware* na criação de alguns cenários de IoT, porém diferente do presente artigo não são analisados quais aspectos do ROS poderiam ser melhorados para esse fim. Outro trabalho é [13] que elenca os principais desafios no desenvolvimento de um software desse tipo.

Com relação à instalação não existem problemas inerentes de utilização do ROS, o mesmo pode ser dito da alimentação dos nós que independe do ROS. A plataforma permite a comunicação diretamente entre os nós, com isso maiores tamanhos de rede são suportados. O gargalo existe no estabelecimento das comunicações que depende do nó *master*. O ROS suporta conectividade limitada, pois quando um dos nós fica temporariamente desconectado, quando a conexão é recuperada, as aplicações que estavam se comunicando continuam a funcionar. O *framework* em si não facilita a comunicação com múltiplos nós, mas permite que elas sejam estabelecidas em nível de aplicação. Também não são oferecidas primitivas que facilitem a segurança ou a qualidade de serviço. A mobilidade é tratada com sucesso desde que a comunicação entre os nós seja possível.

A qualidade de serviço pode ser implementada em nível de aplicação quando for necessário. A segurança pode ser implementada estendendo a API adicionando principalmente funcionalidades para garantir autenticidade, que poderia se dar através de troca de chaves privadas. No quesito segurança outra alternativa seria integrar o IPSec[14] dentro do ROS. Com isso a criptografia pode ser implementada nas primitivas de envio e recebimento de mensagens. Por fim é importante resolver a pendência do *master*, de forma que possam existir múltiplos *masters* que se comuniquem entre si, evitando que existam pontos que podem tornar o sistema lento, especialmente se tratando do alto número de nós previsto no ambiente de IoT. Adicionando-se as características como abstração de hardware, comunicação *peer-to-peer*, interface com diversas linguagens de programação, organização modular do código tornam o ROS como uma alternativa bastante atrativa no desenvolvimento de um *middleware* para IoT.

Referências

[1] R. MacManus, "AT&T & Cisco Talk Up Internet of Things," http://www.readwriteweb.com/archives/verizon_att_cisco_internet_of_things.php, 2010, ultimo acesso em 5 de março de 2012.

[2] M. e. a. Vermesan, O. e Harrison, "The Internet of Things - Strategic Research Roadmap," 2009, cluster of European Research Projects on the Internet of Things, CERP-IoT.

[3] U. N. I. Council, "Disruptive Civil Technologies – Six Technologies with Potential Impacts on US Interests Out to 2025," http://www.dni.gov/nic/confreports_disruptive_tech.html, 2008, conference Report 2008-07. Último acesso em 7 de março de 2012.

[4] I. over Low power WPAN Group, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals RFC 4919," http://datatracker.ietf.org/doc/rfc4919/, 2007, Último acesso em 20 de julho de 2012.

[5] C. Mello, E. Gonçalves, E. Estrada, G. Oliveira, H. Souto, R. Almeida, S. Botelho, T. Santos, and V. Oliveira, "Tatubot–robotic system for inspection of undergrounded cable system," in *Robotic Symposium, 2008. LARS'08. IEEE Latin American*. IEEE, 2008, pp. 170–175.

[6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, 2009.

[7] O. Project, "Open Source Computer Vision Library," Último acesso em 20 de julho de 2012.

[8] R. e. a. Gerkey, B. e Hedges, "The Player Project," Último acesso em 20 de julho de 2012.

[9] I. over Low power WPAN Group, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) RFC 6568," Último acesso em 20 de julho de 2012.

[10] E. Kosmatos, N. Tselikas, and A. Boucouvalas, "Integrating rfids and smart objects into a unified internet of things architecture," *Advances in Internet of Things*, vol. 1, no. 1, pp. 5–12, 2011.

[11] M. Díaz, D. Garrido, and A. Reyna, "One step closer to the internet of things: Smepp."

[12] L. Roalter, M. Kranz, and A. Möller, "A middleware for intelligent environments and the internet of things," *Ubiquitous Intelligence and Computing*, pp. 267–281, 2010.

[13] M. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*. IEEE, 2012, pp. 21–26.

[14] J. Arkko, V. Devarapalli, and F. Dupont, "Using ipsec to protect mobile ipv6 signaling between mobile nodes and home agents," 2004.