

OCTOPUS: UMA PLATAFORMA DE CONTROLE DE PATCH PANELS CONECTADOS A INTERNET

Phablo Márcio de Paiva Souto; Carlos Diego Franco da Rocha; Gabriel Marques; Lucas Hiago Azevedo Dantas;

Michel Santana de Deus.

Instituto Federal do Rio Grande do Norte - Campus Pau dos Ferros, phablo_marcio@hotmail.com

INTRODUÇÃO

Muitas empresas de tecnologia investem milhares de reais por ano para poder atender a demanda de seus clientes, mas acabam muitas das vezes passando por dificuldades técnicas para continuar provendo seus serviços, de maneira eficiente e ininterrupta. Um problema que foi observado em uma determinada empresa, mas que é realidade em muitas, é a falta de soluções específicas para monitorar e controlar equipamentos de *hardware* mais simples, como *patch panels*, equipamento extremamente comum e utilizado em telecomunicações.

A anatel, estatal que regula os serviços de telecomunicações no Brasil, tem leis e regulamentações específicas para empresas provedoras de serviços de telecomunicações e provedores de acesso a internet. Os problemas técnicos que geram interrupção no provimento do serviço contratado pelo cliente geram bastante transtornos e prejuízos às empresas, pela Resolução nº 632, de 7 de março de 2014 regida pela anatel, as empresas que prestam serviços de acesso a internet devem entregar um uptime mínimo de 99,5%, ou seja, durante um mês são toleradas apenas 3h30min sem acesso por parte do cliente, acima disso a empresa pode ser multada e condenada a compensar o cliente de alguma maneira. Outro ponto importante é que períodos acima de 30 minutos sem acesso também podem gerar penalizações a empresa caso o problema seja recorrente.

As empresas buscam soluções para diminuir ao máximo problemas de downtime na rede mas nem sempre as soluções disponíveis no mercado atendem a demanda das empresas ou tem um custo acessível, principalmente para os pequenos e médios provedores. Uma consequência disso é a má qualidade dos serviços oferecidos.

Os *patch panels* são utilizados para organizar os cabos, e possibilitam uma fácil identificação dos pontos de rede no rack. Eles são utilizados para fazer a conexão entre o cabeamento que sai do rack e chegam às tomadas (cabeamento horizontal) ou em outro *patch panel* interligando outro rack (cabeamento vertical). Estes aparelhos são instalados em locais, muitas vezes distantes, nos casos de torres repetidoras sinal, e que podem interferir diretamente no serviço prestado pela empresa provedora caso apresentem algum tipo de falha pois existem outros equipamentos ligados a eles, equipamentos como antenas, roteadores e pontos de acesso.

Os equipamentos presentes no mercado de provedores de internet, a medida que são utilizados, tornam-se suscetíveis a desgaste, seja por danos causados por meios físicos devido a exposição ao calor excessivo, sol, chuva, vento e outros atritos, ou por falta de manutenção e atualização de software. Equipamentos como antenas e roteadores conectados aos *patch panels* quando sofrem com temperaturas elevadas ou estão sobrecarregados, por exemplo, tendem a travar e apresentar problemas, ocasionando a interrupção da conexão a internet das pessoas conectadas a esses equipamentos. Muitas vezes essas interrupções se prolongam por horas e dependem da

locomoção de um funcionário para que o problema possa ser resolvido, gerando prejuízos financeiros às empresas e deixando seus clientes por um bom tempo sem acesso a internet ou com o acesso parcialmente prejudicado, isso gera transtornos para os usuários e consequentemente para as empresas.

Segundo Lee (2003) caso não seja feita manutenção, os equipamentos apresentam falhas que afetam toda a cadeia produtiva, tornando-os incapazes de desempenharem sua função com eficiência. As estratégias de manutenção podem ser divididas em quatro grupos: manutenção corretiva, preventiva, preditiva e proativa (ENDRENYI EL., 2001).

Devido a novas tecnologias de sensores, meios de comunicação e sistemas computacionais embarcados, tornou-se praticável monitorar em tempo real a condição de comportamento dos equipamentos e realizar reparos de forma rápida e precisa, a eletrônica atualmente permite o uso de microcontroladores que podem desempenhar diversas funções e podem ser utilizados para automatizar tarefas. Tendo em vista a problemática apresentada, este trabalho objetiva a proposição e validação de uma arquitetura que possibilite o efetivo gerenciamento remoto de *patch panels* a fim de evitar a interrupção de serviços e agilizar a tomada de decisão da equipe responsável por monitorar a rede e seus dispositivos ativos, diminuindo o tempo de inatividade desses equipamentos e os custos operacionais para realizar essa tarefa. De forma específica a solução deve viabilizar: monitoramento individualizado de portas do *patch panel*; reinicialização dos dispositivos conectados ao Octopus de qualquer local por meio do acesso a internet, seja por computador ou celular; notificações de quando houver alguma falha que alterar o status do equipamento inesperadamente; não necessidade de deslocamento de uma equipe do suporte para fazer o serviço que o Octopus oferece; garantia de menor custo operacional e ganho de tempo em operações dentro da empresa.

METODOLOGIA

A primeira etapa do trabalho se deu através de uma pesquisa exploratória que buscou compreender mais a fundo o problema estudado e a solução a ser desenvolvida, a partir da realização de uma entrevista com o diretor de uma empresa provedora de acesso a internet que presta serviços em 15 cidades, abrangendo Rio Grande do Norte e Ceará. O próximo passo foi constituído por um levantamento de requisitos para o *hardware* e seu sistema embarcado, a princípio, o *hardware* do *patch panel* teria um sistema embarcado que permitiria o acesso remoto ao equipamento através de uma interface web, porém a ideia de construir uma plataforma web onde fosse possível centralizar o acesso de todos os dispositivos foi mais aceita. Após a definição dos requisitos de *hardware* e software iniciou-se o desenvolvimento da plataforma web, denominada Octopus, e o *hardware* do *patch panel*.

Inicialmente foi usado um microcomputador *Phenom X3* de 4 núcleos e 4gb de memória RAM, para o desenvolvimento do sistema web. Foi adotado o *Play Framework* na versão 1.4, implementado na linguagem Java e na IDE do *Eclipse Neon*. A escolha do *Framework* se deu devido à sua facilidade de uso, rapidez no desenvolvimento e desempenho.

Para o *hardware* foi desenvolvido um sistema na linguagem C utilizando o *Arduino IDE*, o software criado também utiliza o protocolo *MQTT*. Segundo MARTINS e ZEM(2016), trata-se de um protocolo de mensagens baseado na arquitetura *publish-subscribe*, voltado para dispositivos restritos e redes inseguras, com baixa largura de banda e alta latência. O componente de *hardware* conecta-se ao *broker*, com auxílio do *MQTT*, e aguarda instruções que devem ser executadas caso estejam direcionadas a ele, caso o software receba um comando advindo da plataforma web, o

sistema faz com que o *hardware* execute a ação e depois envia uma resposta ao sistema web informando se o comando foi executado corretamente. Os comandos contém uma numeração específica que identifica qual equipamento recebeu e/ou executou o comando recebido e garante que outros dispositivos não executarão comandos que não estejam direcionados especificamente para sí.

RESULTADOS E DISCUSSÕES

Para o protótipo do *hardware* foi utilizado 01 *patch panel* de cinco portas da marca *VOLT*, 01 microcontrolador ATmega328p fabricado pela Atmel, 01 placa com o W5100 *ethernet chip*, 05 relés com acionamento de 5V, 05 resistores de 100 ohms e 05 transistores BC548 além de fios variados para fazer as conexões na *protoboard*. Com o uso desses materiais foi criado um protótipo que permite ativar e desativar uma porta POE (*Power On Ethernet*) do *patch panel* através do uso do relé. A escolha do microcontrolador se deu por ser barato, uma ferramenta de prototipagem bastante difundida e de uso simplificado, com ele é possível criar, de maneira mais simples e barata, protótipos eletrônicos ideais para fase de testes e desenvolvimento. A escolha do W5100 foi dada primeiramente por ter um custo relativamente baixo e segundo por ter um conector RJ45 fêmea para conexão cabeada à internet, a placa oferece suporte a conexão em redes TCP/IP.

Para a construção do sistema web, inicialmente foi utilizado um computador Phenom X3 de 2.3ghz com 4Gb de memória RAM e HD de 500GB e executando o sistema operacional Windows 7, foi instalado o *Play Framework* que foi utilizado para desenvolver o sistema. O Play apresenta como arquitetura o padrão *Model, View, Controler* (MVC). Essa arquitetura é organizada em três camadas: modelo (*Model*), visualização (*View*) e o controlador (*Controller*). Onde a camada de modelo representa os dados e a lógica de negócios da aplicação, ou seja, nessa arquitetura é o objeto que interage diretamente com o banco de dados. A camada de visualização é a apresentação na tela, em aplicações web a visão geralmente é feito utilizando *Hypertext Markup Language* (HTML) ou *Xtensible Markup Language* (XML), neste sistema optamos por usar o HTML. E o controller define a maneira como a interface de usuário reage com a camada de dados, ou seja, é o controlador que processa os eventos solicitado pelo usuário. Isso possibilita um desenvolvimento em paralelo já que os três componentes são independentes, além de facilitar a manutenção. Nesse sentido, o MVC foi idealizado para aumentar a flexibilidade e a reutilização (GAMMA et al., 2000, p. 20).

O sistema web possui dois níveis de acesso, o administrador e suporte. No papel de administrador, o usuário é responsável por cadastrar, editar, remover, listar (CRUD) de funcionários, equipamentos, *patch panels*, e torres. A ele também está atribuído o controle sobre os *patch panels*, podendo desativar, ativar ou reiniciar um *patch panel* completo, ou somente uma determinada porta do equipamento. Enquanto no papel de suporte, o usuário poderá somente visualizar os equipamentos, torres, *patch panels* cadastros. A única atribuição que têm em comum com o administrador é o controle sobre os *patch panels*. O sistema web também conta com uma auditoria interna, um histórico de logs que armazena no banco de dados as principais atividades executadas no sistema e o respectivo usuário conectado que executou. O objetivo é averiguar se essas atividades executadas estão de acordo com o planejamento estabelecido previamente pelo administrador de redes.

Adicionalmente, o uso do *Bootstrap 4* como *Framework* para auxiliar o desenvolvimento da interface responsiva do sistema. O *Bootstrap* é utilizado para desenhar telas em html, que serão acessadas via navegador web ou dispositivo mobile. Com ele, pode-se criar sites inteiros que podem ser acessadas facilmente em diferentes dispositivos e diferentes resoluções de tela.

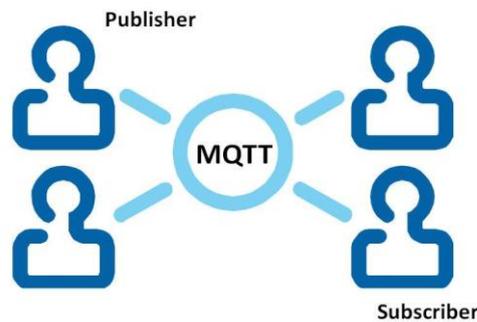


Figura 01 - Ilustração do modelo Publisher-Subscriber utilizado pelo MQTT

O MQTT foi o protocolo de comunicação utilizado para fazer a comunicação entre o sistema web e o software embarcado no *patch panel*. Um cliente MQTT conecta-se a um *broker* MQTT, o *broker* deve ser instalado em uma máquina conectada a internet, e é o responsável pelo recebimento, organização e envio das mensagens dentro dos tópicos, além de garantir a entrega da informação aos assinantes. Dessa forma, pode-se garantir que o comando enviado, através da plataforma web, será recebido pelo sistema embarcado do *hardware* e o mesmo será executado.

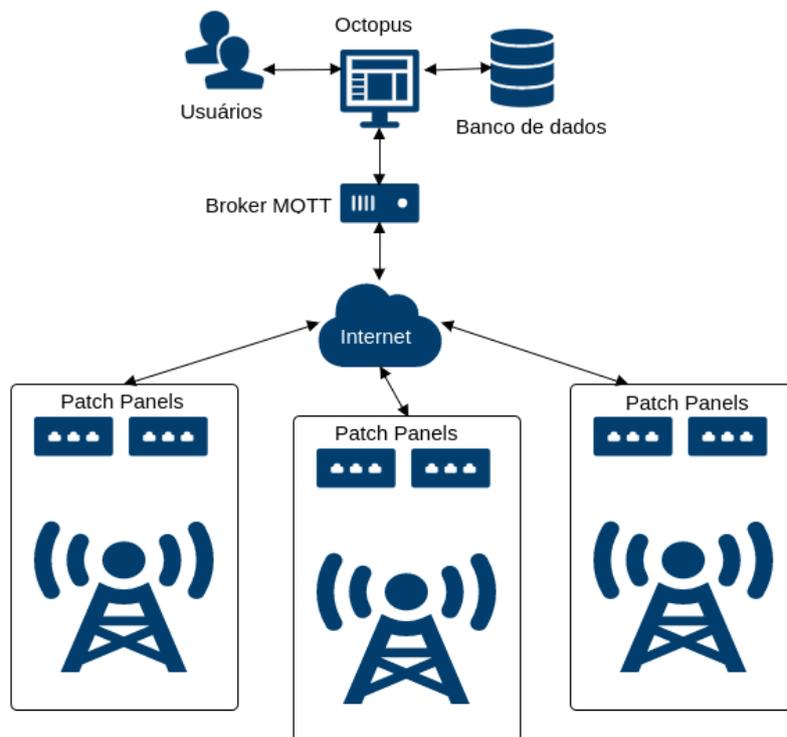


Figura 03 - Ilustração da comunicação entre os componentes

Na figura 03 é possível visualizar a comunicação entre os componentes propostos pela arquitetura do Octopus. O usuário acessa o sistema que por sua vez utiliza um banco de dados para armazenar suas informações, essas informações são mostradas na interface do sistema para que o usuário possa fazer uso, caso o usuário decida ativar ou desativar alguma das portas de um determinado *patch panel*, ele consulta o dispositivo e seleciona qual ação deseja tomar, essa ação gera um comando que é enviado para o *broker* com um número que identifica o equipamento que deverá executar o comando, ao chegar ao *broker*, a mensagem é encaminhada para o dispositivo

que por sua vez executa o comando e envia uma resposta informando se a operação ocorreu da maneira correta, essa mensagem é encaminhada para a plataforma e é apresentada ao usuário em forma de uma notificação na interface gráfica.

Os objetivos elencados inicialmente foram atingidos fazendo uso da arquitetura proposta, a plataforma web e o *hardware* se mostraram capazes de executar todas as ações necessárias identificadas no levantamento de requisitos. Em testes a plataforma e o *hardware* do *patch panel* se comportaram da maneira esperada, enviando e recebendo as mensagens do *broker* e executando as ações esperadas.

CONSIDERAÇÕES FINAIS

Como trabalho futuro, sugere-se o aprimoramento da aplicação Octopus como plataforma web adicionando mais funções, como por exemplo integração com APIs (*Acess Public Interface*) de fabricantes de equipamentos, com o fim de contribuir no trabalho do profissional da área de suporte a provedores, buscar também o desenvolvimento de um *hardware* e um sistema embarcado mais sofisticados, já que atualmente o *hardware* conta com limitações que são as adaptações feitas em um modelo já comercializado, a fim de suprir todas as necessidades dos usuários, entregar um alto padrão de acabamento, confiabilidade e um *hardware* totalmente desenvolvido pela equipe do projeto. Outra proposta é o desenvolvimento de uma plataforma *mobile* para que administradores de redes possam consultar os equipamentos conectados e receberem notificações quando houver um problema na rede, além de permitir a realização de operações através de SMS (*Short Message Service*).

REFERÊNCIAS

ENDRENYI, J. et al. The present status of maintenance strategies and the impact of maintenance on reliability. *IEEE Transactions on power systems*, v. 16, n. 4, p. 638-646, 2001.

GAMMA, Erich et al. *Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos*. Porto Alegre: Bookman, 2000.

LEE, J. *E-manufacturing-fundamental, tools, and transformation*. Robotics and computer-integrated manufacturing, 2003

MARTINS, Ismael Rodrigues; ZEM, José Luís. Estudo dos protocolos de comunicação MQTT e COaP para aplicações machine-to-machine e Internet das coisas. *Revista Tecnológica da Fatec Americana*, v. 3, n. 1, p. 24, 2016.

Resolução nº 632, de 7 de março de 2014, <<http://legislacao.anatel.gov.br/resolucoes/2014/750-resolucao-632>> acessado em 15 de Setembro de 2017.