

Abordagens para Descoberta e Configuração de UCPs por Estações de Trabalho

Andrey Blazejuk¹, Alfredo Gaubert Capella Junior¹, Sérgio Luis Cechin¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{ablazejuk, alfredo.junior, cechin}@inf.ufrgs.br

Abstract. *Automatic discovery and configuration of new devices on a computer network are challenges in industrial environments. In this context, researches on new approaches to make this procedure simpler are extremely important to companies that work to improve the final user experience. In this paper, two approaches to discovery and configuration of CPUs by workstations are proposed. One of the approaches applies to networks that support dynamic configuration of IP addresses and the other to networks where IP addresses configuration is done statically. The evaluation results models the proposed solutions using activity diagrams. The models illustrate automation of tasks that reduce or eliminate user participation in this process.*

Resumo. *A descoberta e a configuração automática de novos dispositivos em uma rede de computadores são desafios em ambientes industriais. Neste contexto, as pesquisas de novas abordagens para tornar este procedimento mais simples são de extrema importância para as empresas que trabalham para melhorar a experiência do usuário final. Neste artigo, duas abordagens para descoberta e configuração de UCPs através de estações de trabalho são propostas. Uma das abordagens se aplica a redes com suporte a configuração dinâmica de endereços IP e a outra a redes onde a configuração de endereços IP é feita estaticamente. A avaliação dos resultados modela as soluções propostas utilizando diagramas de atividade. Os modelos ilustram a automatização de tarefas que reduzem ou eliminam a participação do usuário neste processo.*

1. Introdução

Em ambientes industriais, diversos dispositivos devem estar conectados pela rede para poderem se comunicar através da troca mensagens. Um destes dispositivos pode ser um CLP (Controlador Lógico Programável) [Munir et al. 2010] ou uma UTR (Unidade Terminal Remota) [Jusoh et al. 2014], que contém uma UCP (Unidade Central de Processamento) utilizada para realizar o controle e o monitoramento de processos. Frequentemente, uma nova UCP deve ser inserida na rede ou deve substituir outra com defeito. Além da instalação física do novo equipamento, uma aplicação deve ser carregada no dispositivo para determinar suas configurações e seu comportamento. Este procedimento é realizado pelo usuário e, além de ser trabalhoso, pode levar a erros de configuração de rede.

Uma estação de trabalho é um computador de propósito geral, que executa um software utilizado pelo usuário para programar UCPs através do carregamento de

aplicações. Um exemplo de topologia típica em redes industriais é apresentado na figura 1. Para realizar a transferência da aplicação pela rede, a estação de trabalho e a UCP devem estar configuradas com endereços IP válidos na mesma sub-rede. Assim, o usuário pode selecionar qual é o equipamento que deverá receber a aplicação em uma lista de UCPs encontradas. No projeto SDCD (Sistema Digital de Controle Distribuído), realizado entre o Instituto de Informática da UFRGS e a Altus Sistemas de Automação S.A., são propostas novas abordagens para descobrir e configurar UCPs recém inseridas na rede a partir deste software.

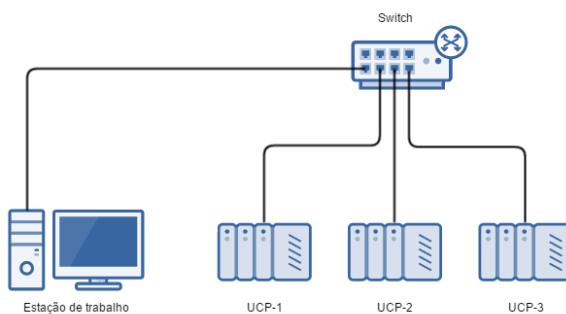


Figura 1. Exemplo de topologia utilizada em redes industriais.

Nos sistemas utilizados atualmente, o software na estação de trabalho envia datagramas UDP para o endereço de *broadcast* da sub-rede. Os dispositivos que possuem um endereço IP configurado dentro da mesma sub-rede da estação de trabalho respondem a estas mensagens, informando seus dados de identificação, e podem ser encontrados. As UCPs suportam apenas a configuração fixa de endereços IP em suas interfaces de rede e a primeira configuração das interfaces do equipamento é feita de forma padrão durante o processo de fabricação. Na instalação de um novo dispositivo, os parâmetros definidos na fábrica dificilmente estarão de acordo com as configurações da rede do usuário, que precisa encontrar o dispositivo para alterá-los. O problema desta abordagem é que uma nova aplicação não pode ser enviada para a UCP, porque ela não está na mesma sub-rede que estação de trabalho, e a configuração das interfaces não pode ser alterada, porque o usuário não pode atualizar a aplicação existente no dispositivo. A solução adotada pelos usuários é alterar manualmente as configurações de rede da estação de trabalho para que ela esteja na mesma sub-rede da UCP.

Neste trabalho, novas abordagens para facilitar a descoberta e a configuração de interfaces de rede de UCPs pelas estações de trabalho são sugeridas. Estas abordagens visam reduzir as limitações existentes nos sistemas atuais, adicionando às UCPs um novo modo de configuração de interfaces de rede, que não exija o *download* de uma nova aplicação, e suporte a configuração dinâmica de endereços IP. Através das soluções propostas, um dispositivo com as configurações de fábrica que for inserido na rede pode ser identificado e reconfigurado de modo que a quantidade de intervenções do usuário seja minimizada tanto na adição de uma nova UCP, causada pela expansão do processo controlado e monitorado pelo sistema, quanto na substituição de um dispositivo defeituoso.

O restante do artigo foi construído da seguinte forma. Na Seção 2, os trabalhos de

pesquisa relacionados a este são analisados. A Seção 3 apresenta as propostas sugeridas para detecção e configuração de novos equipamentos inseridos na rede. Os resultados obtidos com o uso das diferentes abordagens tratadas são demonstrados na Seção 4. Por fim, a Seção 5 finaliza o artigo e sugere trabalhos futuros.

2. Trabalhos relacionados

[Guttman 2001] apresenta as soluções propostas pelo IETF para estabelecer redes IP sem realizar a configuração prévia dos dispositivos ou de serviços de rede. A autoconfiguração no nível de rede permite que os dispositivos configurem suas interfaces com endereços IP únicos, determinem as máscaras de sub-rede que devem utilizar, detectem endereços duplicados e lidem com colisões automaticamente. Um endereço IPv4 *link-local* é configurado para ser único no enlace, mas dispositivos que possuem múltiplas interfaces de rede com suporte a autoconfiguração pode encontrar ambiguidades na transmissão de pacotes.

Em [McAuley et al. 2001], um novo protocolo que distribui as informações de configuração de endereços entre sub-redes, chamado de DCDP (*Dynamical Configuration Distribution Protocol*), é proposto. O DCDP é utilizado em conjunto com um protocolo de configuração de sub-rede, como o DHCP (*Dynamic Host Configuration Protocol*) [Droms 1997], para configurar automaticamente um domínio de rede completo. A distribuição de faixas de endereços pode ser feita do nodo DCDP para o protocolo de configuração de sub-rede, fornecendo a quantidade mínima de endereços possível, ou de um nodo DCDP para outro nodo DCDP, com o nodo pai distribuindo metade de seus endereços disponíveis para o nodo filho.

No trabalho de [Johannessen 2004], diversos protocolos da pilha TCP/IP são brevemente apresentados e suas possíveis aplicações em sistemas de automação industrial são exploradas. Uma solução proposta no artigo para configurar automaticamente dispositivos conectados nestes sistemas faz uso dos protocolos DHCP, DNS (*Domain Name System*) [Mockapetris 1987] e TFTP (*Trivial File Transfer Protocol*) [Sollins 1992]. O protocolo DHCP é utilizado para transmitir as configurações de rede definidas no servidor DHCP ao dispositivo, além dos endereços IP do servidor DNS e do servidor de *boot*. Os dispositivos são acessados por seus nomes, já que o endereço IP correspondente a um nome pode ser obtidos através de consultas ao servidor DNS. O protocolo TFTP é usado para realizar o download do arquivo de *boot* do servidor e, possivelmente, atualizar o firmware do dispositivo para a versão mais recente.

3. Descoberta e configuração de novos dispositivos

A descoberta de UCPs a partir de uma estação de trabalho é necessária para a detecção, configuração e monitoramento destes dispositivos pelo software usado para programá-los. A detecção é realizada quando o usuário do software solicita a realização de uma varredura na rede à procura de todas as UCPs conectadas. Como resultado da varredura, uma lista com todos os equipamentos encontrados é exibida para que o usuário selecione o dispositivo que deseja configurar ou monitorar. A configuração permite a alteração dos parâmetros das interfaces de rede, a definição de módulos redundantes e a criação de tarefas que são executadas periodicamente pelas UCPs. O monitoramento faz uso das informações geradas pelos diagnósticos executados nos dispositivos e exibe estas informações na tela da estação de trabalho.

O procedimento de descoberta utilizado pelos sistemas existentes assume que a interface de rede da UCP procurada possui um endereço IP válido na mesma sub-rede da estação de trabalho. Portanto, as configurações de rede do dispositivo devem ser definidas antes do *download* de uma aplicação. Após a primeira configuração, as alterações posteriores podem ser realizadas com o envio de uma nova aplicação para a UCP sem maiores problemas. Assim, a necessidade de intervenção do usuário na primeira configuração é a maior limitação neste sistema.

As duas abordagens propostas para remover esta limitação são detalhadas a seguir. A Seção 3.1 descreve como a descoberta e a configuração de interfaces podem ser feitas em redes com suporte a configuração dinâmica de endereços IP, mais especificamente através do protocolo DHCP. Esta abordagem não é suficiente para todos os casos, porque uma UCP é identificada unicamente por seu endereço IP em alguns sistemas. Se a configuração dinâmica fosse utilizada em um cenário deste tipo, a substituição de um equipamento com defeito resultaria na atribuição de um endereço IP dinâmico diferente do antigo para o dispositivo substituinte e poderia comprometer o funcionamento do sistema. Para atender a estes sistemas, a proposta para descoberta e configuração em redes que suportam apenas a configuração de endereços IP fixos é apresentada na Seção 3.2. A escolha da abordagem mais adequada depende do tipo de configuração existente na rede do usuário.

3.1. Redes com suporte a configuração dinâmica de endereços IP

O protocolo DHCP é amplamente utilizado em *intranets* para configurar e gerenciar a alocação de recursos de forma dinâmica e automática [Wang and Lee 2002]. Se houver um servidor DHCP responsável por administrar os endereços IP da sub-rede em que a estação de trabalho se encontra, a primeira configuração da UCP pode ser automatizada com uma simples atualização de firmware que inclua um cliente DHCP. O suporte à configuração dinâmica de endereços IP deve ser habilitado por padrão para que, ao inserir um novo dispositivo na rede, o cliente em execução se comunique com o servidor e configure os parâmetros da interface de rede automaticamente. Até que o usuário faça o download de uma aplicação que defina um endereço IP fixo para a UCP, o funcionamento desta abordagem depende da disponibilidade do servidor DHCP, que deve estar ativo para renovar o endereço IP concedido ao dispositivo.

3.2. Redes que exigem a configuração de endereços IP fixos

Em redes que não possuem um servidor DHCP em execução, um endereço IP único dentro da sub-rede deve ser atribuído estaticamente para cada dispositivo conectado. Nesta abordagem, existem duas opções de configuração: a interface de rede da estação de trabalho é reconfigurada para atender à configuração de rede da UCP ou a interface de rede da UCP é configurada de acordo com os parâmetros informados pela estação de trabalho. Em ambos os casos, pacotes devem ser enviados para o endereço MAC de *broadcast* pelo software que executa na estação de trabalho. Os dispositivos conectados que receberem estas mensagens enviam como resposta suas informações de identificação. A partir destas informações, o software pode listar os dispositivos encontrados para que o usuário selecione qual deles deseja configurar.

Uma vez que a UCP tenha sido identificada pelo usuário, a interface de rede que terá suas configurações alteradas deve ser escolhida. Se o usuário optar por mudar a

configuração da estação de trabalho, ele deve informar as novas configurações de rede e o software utilizará estas informações para configurar a interface de rede da estação. Ao alterar as configurações de rede da estação de trabalho, serviços podem se tornar indisponíveis e a comunicação com os demais dispositivos na rede é interrompida. Caso o usuário escolha alterar as configurações da UCP, ele deve informar os novos parâmetros que o dispositivo receberá. Estes parâmetros serão enviados pelo software da estação para a UCP em uma nova mensagem. Ao receber a mensagem, o dispositivo configura sua interface de rede com os novos parâmetros. Dessa forma, a UCP pode se comunicar com qualquer dispositivo na sub-rede sem que tenha recebido uma aplicação da estação de trabalho.

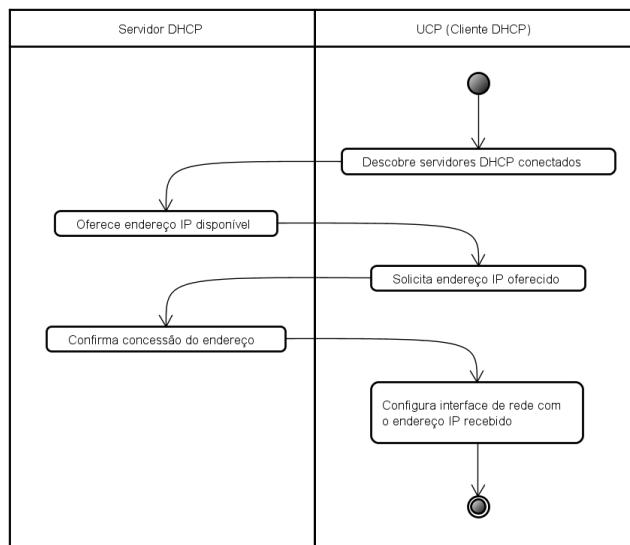


Figura 2. Atividade de configuração de rede com endereço IP dinâmico.

4. Resultados e discussão

Com o objetivo de ilustrar o funcionamento das abordagens propostas, as soluções para configuração de interfaces de rede foram modeladas através de diagramas de atividades [Rumbaugh et al. 2004]. A figura 2 apresenta o funcionamento da configuração de endereços IP dinâmicos, onde existe um servidor DHCP na rede e a UCP executa um cliente DHCP. Esta abordagem possibilita que a alteração das configurações de rede da UCP seja feita automaticamente pelo servidor, que fornece um endereço IP disponível na rede ao cliente. Assim, o usuário não interfere diretamente no processo de configuração inicial e é capaz de acessar o novo dispositivo a partir do software que executa na estação de trabalho.

A segunda abordagem se aplica a redes onde os endereços IP são fixos. Neste caso, o processo de configuração inicial da UCP deve ser disparado e acompanhado pelo usuário no software da estação de trabalho, como pode ser visto na figura 3. O acompanhamento é necessário para selecionar o dispositivo desejado entre os identificados e a

interface de rede que deve ser configurada. Com base nas escolhas feitas pelo usuário, as configurações de rede da estação de trabalho ou da UCP serão alteradas.

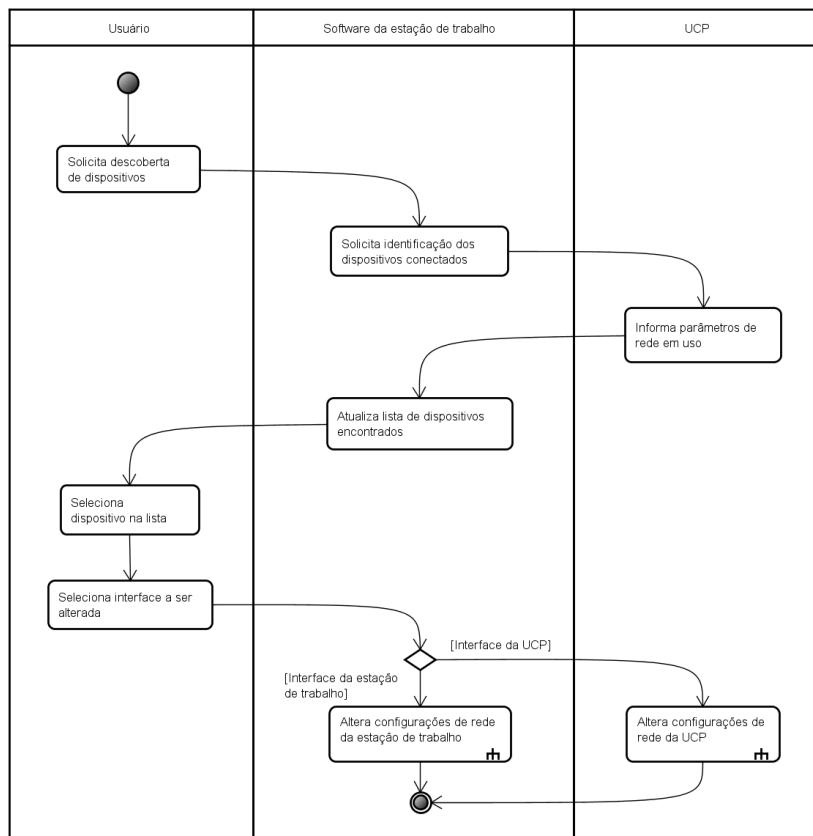


Figura 3. Atividade de configuração de rede com endereço IP fixo.

Na figura 4, a configuração de rede da estação de trabalho utilizando um endereço IP fixo é alterada para possibilitar o download de aplicações para a UCP a partir do software na estação. Assim que os novos parâmetros de rede são informados, esta alteração pode ser realizada pelo software sem que o usuário tenha que acessar as configurações do adaptador de rede manualmente através do sistema operacional da estação de trabalho.

A atividade de configuração dos parâmetros de rede da UCP com endereço IP fixo é ilustrada na figura 5. O uso desta solução implica na participação do usuário para definir explicitamente a nova configuração de rede da UCP. Em seguida, os novos parâmetros de rede são transmitidos da estação de trabalho para a UCP e o dispositivo atualiza a configuração de sua interface de rede automaticamente.

5. Conclusão

A inserção de um novo dispositivo em redes industriais é um procedimento frequentemente necessário e trabalhoso. As abordagens propostas neste trabalho têm como obje-

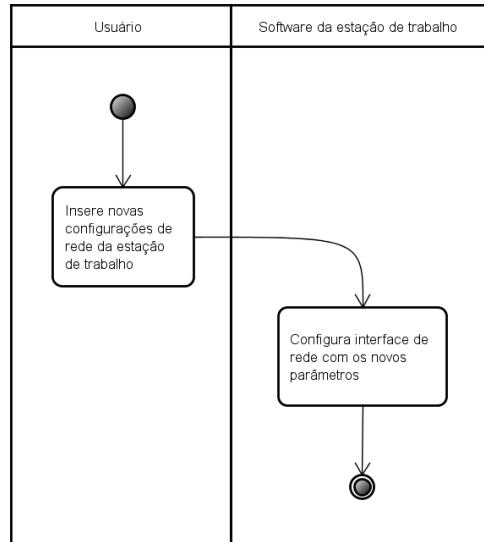


Figura 4. Atividade de alteração das configurações de rede da estação de trabalho.

tivo tornar a primeira configuração destes equipamentos mais fácil. Isso é feito através da automatização por software de tarefas que atualmente são realizadas de forma manual. A escolha da abordagem a ser implementada depende do suporte da rede à distribuição de endereços IP dinâmicos ou não e da preferência de configuração das interfaces pelo usuário.

Os trabalhos futuros desenvolvidos a partir deste podem explorar as limitações impostas por redes com endereços IP configurados estaticamente, propondo meios de identificar um endereço disponível na sub-rede para ser usado na configuração da interface de rede da nova UCP. Assim, o acompanhamento do usuário seria necessário apenas para identificação do dispositivo, enquanto a configuração de UCPs em redes sem alocação dinâmica de endereços IP seria feita de forma completamente automática.

Referências

- Droms, R. (1997). Dynamic host configuration protocol. RFC 2131, RFC Editor. <http://www.rfc-editor.org/rfc/rfc2131.txt>.
- Guttmann, E. (2001). Autoconfiguration for ip networking: enabling local communication. *IEEE Internet Computing*, 5(3):81–86.
- Johannessen, S. (2004). Administering ethernet automation networks. In *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, pages 423–428.
- Jusoh, W. N. S. E. W., Ghani, M. R. A., Hanafiah, M. A. M., and Raman, S. H. (2014). Remote terminal unit (rtu) hardware design and development for distribution automation system. In *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, pages 572–576.

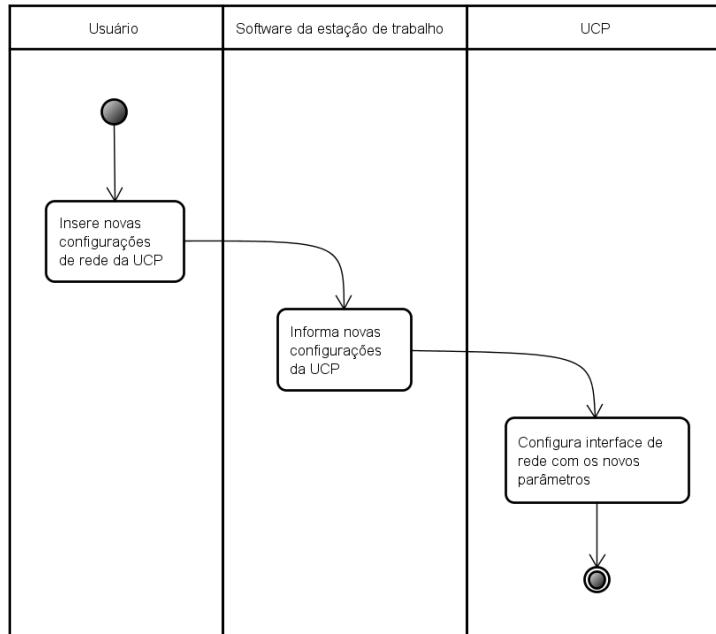


Figura 5. Atividade de alteração das configurações de rede da UCP.

McAuley, A., Misra, A., Wong, L., and Manousakis, K. (2001). Experience-with autoconfiguring a network with ip addresses. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force*. IEEE, volume 1, pages 272–276 vol.1.

Mockapetris, P. (1987). Domain names - implementation and specification. STD 13, RFC Editor. <http://www.rfc-editor.org/rfc/rfc1035.txt>.

Munir, H. A., Saad, N., Junid, S. A. A. S., Zaidi, A. M. A., Yusoff, M. Z., and Jaffar, A. (2010). Real-time communication between personal computer and programmable logic controller for networked control system based on industrial ethernet. In *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*, pages 610–614.

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *Unified Modeling Language Reference Manual, The (2Nd Edition)*, chapter 7, pages 81–84. Pearson Higher Education.

Sollins, K. (1992). The tftp protocol (revision 2). STD 33, RFC Editor. <http://www.rfc-editor.org/rfc/rfc1350.txt>.

Wang, J.-H. and Lee, T.-L. (2002). Enhanced intranet management in a dhcp-enabled environment. In *Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International*, pages 893–898.

Análise de metadados no tráfego do protocolo BitTorrent

Tiago da Silveira Pasa¹, Carlos Vinícius Rasch Alves¹, Eduardo Maroñas Monks¹

¹Curso Superior de Redes de Computadores
Faculdade de Tecnologia SENAC Pelotas (FATEC)

tiagopasa@hotmail.com, cvalves@senacrs.edu.br, emmonks@gmail.com

Abstract. This paper presents techniques to detect BitTorrent protocol and its metadata. Tests comparing DPI techniques and signature-based detection, using open source and commercial tools, were performed. The packages were taken from various torrent sites, with and without encryption enabled. The chosen tools allowed the differentiation between legal and illegal traffic.

Keywords: BitTorrent, deep packet inspection, signatures, IDS, tests.

Resumo. Este artigo apresenta técnicas para detecção do protocolo BitTorrent e seus metadados. Foram realizados testes comparando as técnicas de DPI e detecção baseada em assinaturas, utilizando ferramentas open source e comerciais. As capturas de pacotes foram realizadas de diversos sites torrent, com e sem criptografia ativada. As ferramentas escolhidas permitiram realizar a diferenciação entre tráfego legítimo e ilegal.

Palavras-Chave: BitTorrent, deep packet inspection, assinaturas, IDS, testes.

1. Introdução

Atualmente, o protocolo BitTorrent [Cohen 2008] é amplamente utilizado no mundo pela sua eficiência e facilidade de compartilhamento de arquivos de diversos tipos, tamanhos e para vários fins. Estima-se que 35% do tráfego da Internet é realizado através deste protocolo [Sandvine 2016]. Devido à vasta gama de *downloads* realizados, com conteúdos protegidos por lei sendo distribuídos de forma ilegal [Silva 2015], se faz necessária a pesquisa de técnicas e ferramentas para analisar o conteúdo do tráfego BitTorrent. Considerando que a decodificação deste tipo de fluxo de dados ainda é pouco conhecida e utilizada, devido a sua complexidade, realizar sua identificação é muito importante para prevenir futuros processos judiciais, ocasionados pelo acesso a conteúdos protegidos por direitos autorais. Em virtude do tráfego BitTorrent estar circulando quase totalmente de forma encriptada [Zhe Yang 2012] e devido à ineficácia dos métodos tradicionais de identificação, este trabalho tem por objetivo realizar uma análise das técnicas existentes para detecção e identificação de tráfego BitTorrent.

2. Protocolo BitTorrent

Nesta seção serão apresentadas as principais características do protocolo BitTorrent.

2.1. Funcionamento do protocolo BitTorrent

O protocolo BitTorrent, também conhecido como BTP, é um protocolo *peer-to-peer* (*p2p*) desenvolvido por Bram Cohen, em 2001 [Cohen 2008]. As redes *p2p* se tornaram a principal alternativa na distribuição eficiente de dados na Internet, por terem como característica fundamental o fato de que cada usuário se comporta, simultaneamente, como cliente e servidor. Na Figura 1 pode ser vista a estrutura do arquivo de metadados, gerada a partir da ferramenta dumptorrent [Dumptorrent 2016].

```

root@srv-pasa:~/dumptonitorrent-1.2# ./dumptonitorrent -v [otorrents.com]shrek-2001-720p.torrent
[otorrents.com]shrek-2001-720p.torrent:
Name: Shrek-2001-720p
Size: 629563032 (600M)
Announce: udp://open.demonii.com:1337
Info Hash: 13eF3621736433edcfda6bc7bcec1221526b1ebf
Piece Length: 131072
Creation Date: Sat Mar 10 21:48:02 2012
Created By: uTorrent/3000
Encoding: UTF-8
Files:
    Other/AhaShare.com.txt
    Other/Torrent downloaded from Demonoid.com - Copy.txt 59
    Other/Torrent Downloaded From ExtraTorrent.com.txt 47
    Shrek-2001-720p.BluRay.x264-YIFY.mp4 353
    Shrek-2001-720p.BluRay.x264-YIFY.srt 113650 (111K)
    WW.YIFY-TORRENTS.COM.jpg 130677 (128K)
Announce List: udp://open.demonii.com:1337

```

Figura 1. Estrutura do arquivo de metadados

Para participar de uma rede *p2p* é necessário que seja realizado o acesso a *Sites* públicos ou privados que compartilhem conteúdo através do BitTorrent. Para dar início ao processo de *download* é necessária a obtenção do arquivo de metadados com extensão .torrent, que contém várias informações dos arquivos que serão baixados e dos *trackers* que serão contactados. Na Figura 2 pode ser visto o cliente em contato com o *tracker* e recebendo uma lista com todos os *peers* do *swarm*. Este processo é conhecido como ”anúncio”. Após conectado com os *peers* do *swarm*, o cliente verifica quem tem os pedaços dos arquivos a oferecer. Para cada pedaço recebido, o cliente calcula o *hash SHA1* do pedaço, conferindo se está de acordo com o valor presente no arquivo de metadados. Após o *download*, o cliente passa a ser um *seeder* dentro do *swarm*.

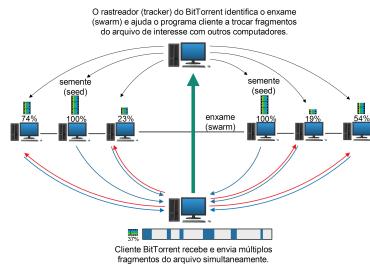


Figura 2. Funcionamento do BitTorrent [HowStuffWorks 2016]

2.2. Criptografia

As técnicas de criptografia *Message Stream Encryption* (MSE) e *Protocol Encryption* (PE) utilizam o algoritmo simétrico de criptografia de fluxo RC4, escolhido pela sua velocidade. O RC4 efetua a troca de chaves combinadas com a *info hash* do arquivo .torrent, visando aumentar a privacidade e confidencialidade. Além disso, elas tentam tornar o tráfego mais difícil de ser identificado por terceiros, incluindo provedores de serviços de Internet (ISPs) e administradores de rede. A encriptação tem por objetivo ofuscar e tornar o tráfego mais difícil de ser detectado, limitando-se a esconder o cabeçalho do protocolo para que não seja possível identificar quem está utilizando o BitTorrent, porém tende a consumir mais recursos do processador, quando ativada.

3. Técnicas de detecção

Nesta seção serão apresentadas duas técnicas de detecção de tráfego, aplicáveis ao padrão de tráfego do protocolo BitTorrent.

3.1. Deep Packet Inspection (DPI)

A técnica de *Deep Packet Inspection* (DPI) é baseada na inspeção do *payload* de pacotes, com o propósito de extração de metadados. Pode operar no modo detecção ou prevenção, para proteger redes ou sistemas, de acordo com o posicionamento que for implantado. Em provedores de acesso à Internet, essa técnica é utilizada para proteger as redes internas, bem como as redes dos clientes [Bujlow and Carela-Espanol 2013]. A DPI é utilizada para implementar certas políticas que cobrem as violações de direitos autorais, conteúdo ilegal e utilização abusiva da largura de banda.

3.2. Detecção baseada em assinaturas

Esta técnica visa a detecção por meio de assinaturas e monitoramento das atividades da rede, procurando por eventos que correspondam a padrões pré-definidos de ataques, tráfego malicioso e outras anormalidades. Os eventos correspondentes à determinada assinatura podem ser visualizados em tempo real, através do console *Linux* ou de *interfaces Web* desenvolvidas para esta finalidade. Embora esta técnica seja mais eficiente para identificar tráfego não encriptado do protocolo BitTorrent, ainda existem partes que podem ser identificadas mesmo utilizando a criptografia MSE/PE [Zhe Yang 2012]. As técnicas baseadas em assinaturas são geralmente implementadas em sistemas conhecidos como IDS (*Intrusion Detection System*) e IPS (*Intrusion Prevention System*) [Moraes 2010]. Combinando as melhores funcionalidades de ambas pode-se, além de detectar, prevenir e realizar o bloqueio de atividades maliciosas.

4. Ferramentas

Nesta seção são analisadas ferramentas que implementam técnicas de detecção de pacotes, por assinatura e por DPI.

4.1. nDPI-ntop

É um *software open source* distribuído sob a licença *GNU LGPLv3*. É um sistema multiplataforma e está disponível para instalação em distribuições *GNU/Linux*, sistemas *Windows* e *Mac OS* [nDPI 2016]. A ferramenta suporta identificação de cerca de 170 protocolos. A última versão lançada do nDPI inclui funcionalidades de decodificar e não apenas detectar o tráfego, possibilitando extrair o *hashid* dos arquivos que estão sendo baixados e identificar o conteúdo.

4.2. Xplico

O Xplico é uma ferramenta *open source* de análise forense de rede. Tem por objetivo extrair o conteúdo dos dados, a partir de capturas de tráfego, e possibilidade de reconstruir as conexão de rede, a partir dos pacotes capturados. É um sistema que está disponível somente para a instalação em distribuições *GNU/Linux*, mais especificamente para *Fedora* e *Ubuntu* [Xplico 2016]. A ferramenta possui suporte a uma série de *plugins* que podem “decodificar” o tráfego de rede de diversos protocolos.

4.3. Suricata

O Suricata [Suricata 2016] é um sistema *open source* com funções de IDS (*Intrusion Detection system*) e IPS (*Intrusion Prevention System*). Implementa uma completa linguagem de assinaturas ligadas a ameaças conhecidas, violações de políticas de segurança e comportamentos maliciosos de *malwares*, *trojans* e outros ataques. Também é uma ferramenta com mecanismos de NSM (*Network Security Monitor*) que realiza o *log* de solicitações HTTP, *logins*, o armazenamento de certificados TLS (*Transport Layer Security*), a extração de arquivos de um fluxo de dados e o armazenamento em disco. Este tráfego pode ser visualizado em tempo real, na forma de gráficos, através da interface *Web Snorby* [Snorby 2016].

4.4. LANGuardian

O LANGuardian [LANGuardian 2016] é uma solução comercial desenvolvida pela Net-Fort. É composto por *software* de detecção de intrusão de rede e possui uma suíte de aplicações de análise de tráfego, tornando-se uma solução única, capaz de detectar anomalias de rede, atividades suspeitas e até mesmo ameaças desconhecidas. Desenvolvida com tecnologia *Deep Packet Inspection*, é capaz de extrair metadados dos pacotes que trafegam na rede. Combinando a técnica de DPI com a de análise baseada em assinaturas, a ferramenta LANGuardian tornou-se uma poderosa suíte de monitoramento.

5. Testes Realizados

Os testes foram aplicados em um ambiente controlado, com o objetivo de avaliar a aplicação das ferramentas e obter dados para análise e comparações.

5.1. Cenário de testes

O cenário de testes foi idealizado sob ambiente físico e virtualizado com VMware Workstation 11 (CPU de 8 núcleos de 3.50GHz e 16GB de memória RAM), contando com cinco máquinas virtuais dedicadas para as ferramentas de análise. Nestas estão instalados os sistemas de monitoramento de nDPI-ntop, Xplico, Suricata, LANGuardian e o *software* hostapd [Hostapd 2016] com funcionalidade de *access point*, por meio de um adaptador USB *wireless* conectado na máquina hospedeira, conforme pode ser visto na Tabela 1.

Tabela 1. Configuração das máquinas virtuais

Softwares	CPU	RAM	HD	Sistema Operacional
nDPI-ntop 1.7	2x3,50 GHz	1GB	60 GB	Debian 7.0
Xplico 1.1.2	1x3,50 GHz	2GB	60 GB	Ubuntu 15.10
Suricata 3.0.1	2x3,50 GHz	4GB	60 GB	Debian 7.0
LANGuardian 14.4.4	2x3,50 GHz	2GB	60 GB	CentOS 7.0
Hostapd 1.0	1x3,50 GHz	1GB	20 GB	Ubuntu 13.10

O hostapd permite a comunicação através de uma *interface wireless* no ambiente de monitoramento de notebooks e smartphones com clientes BitTorrent instalados, conforme pode ser observado na Figura 3.

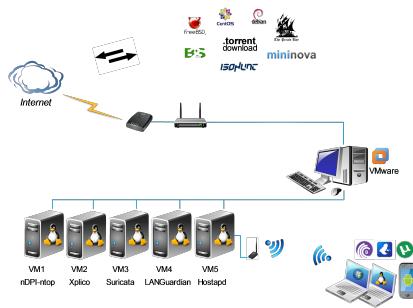


Figura 3. Topologia do cenário de testes

Os clientes torrent selecionados para realização dos testes foram μ Torrent, BitTorrent e Vuze. Os dois primeiros foram escolhidos com base em estatísticas de uso pelos clientes torrent de peers realizando *downloads*, enquanto o Vuze foi escolhido por ser o

primeiro cliente a implantar a criptografia MSE/PE e por suas diversas funcionalidades integradas [Vuze 2016]. Para realização dos testes, os clientes foram instalados em ambiente misto, físico e virtualizado, em sistemas operacionais *Windows 7*, *Windows 10*, *Ubuntu Desktop 12* e *Android 4.1.2*.

5.2. Metodologia de Testes

Inicialmente, os testes foram realizados sem criptografia ativada. Estes consistiram na realização de *downloads* de arquivos torrent a partir de *Sites* públicos, tais como isoHunt [isoHunt 2016], The Pirate Bay [Bay 2016], Debian [Debian 2016] e FreeBSD [FreeBSD 2016], bem como de *Sites* privados como o B2S-Share [B2S 2016]. Todos disponibilizam conteúdos de várias categorias através da arquitetura *p2p*.

No estudo realizado por [Sandvine 2015] já havia sido mostrado que no ano de 2015, cerca de 29% dos usuários de BitTorrent utilizavam criptografia ativada em seus clientes torrent (Figura 4a). Para confirmar que esta opção vem se tornando uma crescente entre os usuários de BitTorrent, foram realizados testes para filtrar *peers* que utilizam as opções de criptografia ativada e não ativada. Os resultados podem ser vistos na Figura 4b. Em comparação com 2015, foi possível perceber que houve um aumento considerável no número de usuários que utilizam a criptografia ativada, o que motivou a realização das simulações de captura de metadados utilizando as ferramentas com esta opção.



Figura 4. Estatísticas de utilização de criptografia no protocolo BitTorrent

6. Resultados

Após os testes, foi realizada a comparação da eficiência das ferramentas nas capturas de metadados do protocolo BitTorrent, em situações distintas, como pode ser observado no gráfico da Figura 5.



Figura 5. Análise dos metadados detectados

Percebe-se que dois dos metadados, o *announce* e o *info hash*, são identificáveis mesmo que submetidos em testes com criptografia ativada. Estes dados capturados podem ser utilizados como “coringas” na diferenciação de tráfego legítimo de ilegal, a partir de buscas realizadas no Google pelo metadado *info hash*. Outra forma seria relacionar o nome do *tracker* que está presente no *announce* ao nome do arquivo torrent, conforme Figura 6.

SENSOR	SIGNATURE	PROTOCOL	SOURCE IP	SPORT	DESTINATION IP
1: Sensor 1	P2P BitTorrent Announce	TCP	192.168.147.171 (CLIENTE3)	57499	130.239.18.159 (btracker.debian.org)
1: Sensor 1	P2P BitTorrent Announce	TCP	192.168.147.171 (CLIENTE3)	57614	130.239.18.159 (btracker.debian.org)
1: Sensor 1	192.168.147.171 (CLIENTE3)		7661229811ef32014879ceeddcd4a48f256c88ba	Google	debian-8.4.0-1386-netinst.iso

Figura 6. Identificação do conteúdo torrent através do *hash* e *announce*

Em praticamente sua totalidade, *hashes* de arquivos torrent disponibilizados em *Sites* públicos são visíveis e identificáveis no buscador do Google [Google 2016], permitindo diferenciar o tipo de conteúdo.

Diferentemente disso, *Sites* de comunidades torrent privadas tais como B2S-Share [B2S 2016] e Manicomio Share [Share 2016], não indexam informações de *info hash* dos seus arquivos torrent em buscadores, dificultando a sua identificação e, por estarem operando em modo privado, somente permitem a comunicação através do seu *tracker* principal, o qual contabiliza a taxa (*ratio*) de *upload* e *download* dos usuários cadastrados. No modo privado não é permitida a descoberta de outros *peers* em um *swarm*. As opções *Local Peer Discover* (LPD), *Distributed Hash Table* (DHT) e *Peer Exchange* (PEX) são desabilitadas por padrão, dificultando ainda mais a detecção dos metadados.

Nota-se que somente as ferramentas LANGuardian e Suricata tiveram a eficácia em detectar os metadados *name* e *files*, possibilitando a identificação do tipo de conteúdo com base nos nomes e arquivos, conforme pode ser visto na Figura 7 e na Figura 8.

1: Sensor 1	192.168.147.171 (CLIENTE3)	b47cd54c0f84d89e6fc819cdcfab3c2b59b2d2b80	Google	FreeBSD-10.3-RELEASE-i386-bootonly.iso
1: Sensor 1	192.168.147.181	a98422c4b383519c98021983f605b16109ed0569	Slime Season 3	18

Figura 7. Metadados *name* e *file* - LANGuardian

```
d1:ad2:id20:...T...@pt< d1:ad2:id20:;1I.....<.
...? 12:implied_portie9: >,$'R12:implied_portie9:
info_hash20:[I]L..._1..Lz. info_hash20:..._0.....
...+ 4:name38:FreeBSD-10. a..14:name14:Slime.Seaso
3-RELEASE-i386-bootonly.is n.34:porti34846e5:token20:
o4:porti34846e4:seeddie5:t ..._0...e.4....U..._e1:q13
oken4:0gj.e1:q13:announce :announce peer1:t4:...:1:v
```

Figura 8. Metadados *name* e *file* - Suricata

As ferramentas nDPI e Xplico não conseguiram detectar os metadados *name* e *files* dos arquivos torrent, mas os campos de *info hash* e *announce* foram detectados, conforme mostra a Figura 9a; porém, estas foram mais eficientes em detectar e apresentar os metadados do campo *announce list*, que se trata dos *trackers* auxiliares, que podem ajudar a identificar a origem do conteúdo torrent. A ferramenta nDPI, em testes com criptografia ativada, teve notável queda de desempenho em detectar o *info hash*, como pode ser visto na Figura 9b.

7. Conclusões

As ferramentas Suricata e LANGuardian foram capazes de identificar e gerar alertas de segurança sobre a atividade do protocolo BitTorrent na rede, bem como a captura de metadados presentes no arquivo torrent. A partir destes dados, foi possível realizar a



Figura 9. Metadados info hash e announce - nDPI

diferenciação de tráfego legítimo de tráfego ilegal. A ferramenta Xplico mostrou-se uma ótima solução de análise forense, detectando os metadados e também identificou a atuação de uma empresa que monitora *trackers* de *downloads* ilegais. Entretanto, o Xplico não é destinado a um monitoramento constante de um ambiente de rede. A ferramenta nDPI, além dos metadados, foi capaz de identificar o protocolo QUIC [IETF 2016].

Durante os testes também observou-se que os clientes µTorrent e BitTtorrent implementam um nível de criptografia inferior a do cliente Vuze, que possui opções de utilizar a rede TOR [TOR 2016] e I2P [I2P 2016] para promover o anonimato na Internet. Durante os testes com os clientes *mobile*, notou-se que nenhum deles tem a opção de utilizar criptografia. Pode-se afirmar que a criptografia implementada pelos clientes não é eficiente, a fim de criptografar de forma integral, a comunicação através do protocolo Bit-Torrent. Conclui-se que atualmente é possível realizar diferenciação de tráfego BitTorrent legítimo de tráfego ilegal, evitando problemas com notificações de direitos autorais.

7.1. Trabalhos futuros

Uma proposta para futuros trabalhos seria um estudo de como realizar o bloqueio do protocolo BitTorrent, com base na classificação e diferenciação de conteúdo legal de ilegal, utilizando ferramentas de IDS/IPS/NSM e verificando a eficiência das ferramentas e bloqueios nesse sentido. Outro trabalho a ser desenvolvido poderia ser o estudo do novo conceito de transferência torrent, o WebTorrent [WebTorrent 2016], que atua de forma híbrida com o protocolo BitTorrent.

Referências

- B2S (2016). Available em: <<http://www.b2s-share.com/>>. Accessed: June 2016.
- Bay, T. P. (2016). Available in: <<https://thepiratebay.se/>>. Accessed: April 2016.
- Bujlow, T. and Carela-Espanol, V. (2013). Comparison of deep packet inspection (dpi) tools for traffic classification.
- Cohen, B. (2008). Bram Cohen the bittorrent protocol specification. http://www.bittorrent.org/beps/bep_0003.html. Accessed: 2016-03-12.
- Debian (2016). Available in: <<https://www.debian.org/>>. Accessed: June 2016.
- Dumptorrent (2016). Available in: <<https://sourceforge.net/projects/dumptorrent/>>. Accessed: April 2016.
- FreeBSD (2016). Available in: <<https://www.freebsd.org/>>. Accessed: June 2016.
- Google (2016). Available em: <<https://www.google.com.br/>>. Accessed: June 2016.
- Hostapd (2016). Available in: <<https://github.com/jenssegers/RTL8188-hostapd>>. Accessed: March 2016.
- HowStuffWorks (2016). Available in: <<http://tecnologia.hsw.uol.com.br/bittorrent.htm>>. Accessed: April 2016.
- I2P (2016). Available em: <<https://geti2p.net/>>. Accessed: June 2016.

- IETF (2016). Available em: <<https://tools.ietf.org/html/draft-tsvwg-quic-protocol-00>>. Accessed: June 2016.
- isoHunt (2016). Available in: <<https://isohunt.to>>. Accessed: April 2016.
- LANGuardian (2016). Available em: <<https://www.netfort.com/languardian>>. Accessed: April 2016.
- Moraes, A. F. (2010). *Segurança em Redes - Fundamentos 1. ed.* Editora Érica Ltda, São Paulo.
- nDPI (2016). Available em: <<https://github.com/ntop/nDPI>>. Accessed: April 2016.
- Sandvine (2015). Available in: <<https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/encrypted-internet-traffic.pdf>>. Accessed: June 2016.
- Sandvine (2016). Available in: <<https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf>>. Accessed: August 2016.
- Share, M. (2016). Available in: <<https://www.manicomio-share.com>>. Accessed: June 2016.
- Silva, R. (2015). Relatório indica que brasil é campeão do mundo em pirataria de séries. <https://tecnoblog.net/172165/brasil-campeao-pirataria-series/>. Accessed: 2016-03-17.
- Snorby (2016). Available em: <<https://github.com/Snorby/snorby>>. Accessed: April 2016.
- Suricata (2016). Available in: <<https://suricata-ids.org>>. Accessed: March 2016.
- TOR (2016). Available em: <<https://www.torproject.org>>. Accessed: June 2016.
- Vuze (2016). Available in: <<http://www.vuze.com>>. Accessed: April 2016.
- WebTorrent (2016). Available em: <<https://webtorrent.io>>. Accessed: June 2016.
- Xplico (2016). Available in: <<http://www.xplico.org>>. Accessed: April 2016.
- Zhe Yang, Lingzhi Li, Q. J. (2012). Zhe Yang, Lingzhi Li, Qijin Ji cocktail method for bittorrent traffic identification in real time. <http://www.ojs.academypublisher.com/index.php/jcp/article/view/jcp07018595>. Accessed: 2016-04-15.