UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

IDENTIFICAÇÃO DE TRÁFEGO DO EMULE USANDO REDES NEURAIS ARTIFICIAIS

RODRIGO LANGE

ORIENTADORA: Profª. Drª. CÉLIA GHEDINI RALHA

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
ÁREA DE CONCENTRAÇÃO INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO

PUBLICAÇÃO: PPGENE.DM - 085/2011

BRASÍLIA/DF: NOVEMBRO - 2011.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

IDENTIFICAÇÃO DE TRÁFEGO DO EMULE USANDO REDES NEURAIS ARTIFICIAIS

RODRIGO LANGE

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE PROFISSIONAL EM INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO.

APROVADA POR:

Profª. CÉLIA GHEDINI RALHA, Doutora, CIC/UnB (Orientadora)

Prof. LUIS PEREIRA CALÔBA, Doutor, COPPE/UFRJ

Prof. HELVIO PEREIRA PEIXOTO, Doutor, DPF

BRASÍLIA/DF, 18 DE NOVEMBRO/2011.

FICHA CATALOGRÁFICA

LANGE, RODRIGO

IDENTIFICAÇÃO DE TRÁFEGO DO EMULE USANDO REDES NEURAIS ARTIFICIAIS [Distrito Federal] 2011.

(xx), 125p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2011).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Informática forense 2. Redes neurais artificiais
3. P2P 4. eMule 5. Classificação de fluxo

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

LANGE, R. (2011). IDENTIFICAÇÃO DE TRÁFEGO DO EMULE USANDO REDES NEURAIS ARTIFICIAIS. Dissertação de Mestrado, Publicação PPGENE.DM – 085/2011, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 125p.

CESSÃO DE DIREITOS

NOME DO AUTOR: RODRIGO LANGE.

TÍTULO DA DISSERTAÇÃO DE MESTRADO: IDENTIFICAÇÃO DE TRÁFEGO DO EMULE USANDO REDES NEURAIS ARTIFICIAIS.

GRAU / ANO: Mestre / 2011

RODRIGO LANGE

Rua Professora Sandália Monzon, 210

CEP 82.640-040 Curitiba - PR - Brasil.

DEDICATÓRIA

Para o amor da minha vida, Ilza,

que faz tudo valer a pena.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por ter permitido que eu iniciasse este programa de mestrado.

À minha esposa Ilza que, com seu apoio incondicional, dedicação e amor, possibilitou que este mestrado fosse realizado da forma mais agradável possível.

Aos meus pais, Valter e Loiri, pelo afeto, dedicação e dignidade com que me ensinaram a viver a vida.

Agradeço à minha orientadora Prof^{a}. Dr^{a}. Célia Ghedini Ralha, que durante todo o mestrado esteve à disposição, com bastante paciência, para orientar-me no que foi necessário.

Aos colegas de Polícia Federal Bruno Werneck e Georger Rommel, pela ajuda imprescindível neste trabalho.

Aos grandes companheiros de alojamento na ENAP, André Peron e Rodrigo Albernaz, pela amizade, incentivo e auxílio em muitas etapas deste mestrado.

Aos amigos do Mestrado em Informática Forense, pelo companheirismo e convivência agradável durante todo esse empreendimento.

Aos companheiros do SETEC/PR que direta ou indiretamente auxiliaram no desenvolvimento e conclusão desse trabalho.

Aos membros da Coordenação do Curso de Mestrado Profissionalizante em Engenharia Elétrica com Ênfase em Informática Forense e Segurança da Informação, que possibilitaram essa parceria entre poder público e academia, permitindo um melhor combate à criminalidade em nossa era digital.

O presente trabalho foi realizado com o apoio do Departamento Polícia Federal – DPF, com recursos do Programa Nacional de Segurança Pública com Cidadania – PRONASCI, do Ministério da Justiça.

RESUMO

IDENTIFICAÇÃO DE TRÁFEGO DO EMULE USANDO REDES NEURAIS ARTIFICIAIS

Autor: RODRIGO LANGE
Orientadora: Profª. Drª. CÉLIA GHEDINI RALHA
Programa de Pós-graduação em Engenharia Elétrica
Brasília, novembro de 2011

O presente trabalho propõe o desenvolvimento de um método de identificação do tráfego de rede gerado pelo aplicativo *peer-to-peer eMule*. Com a identificação do fluxo de rede do *eMule*, podem ser obtidas informações periciais importantes tais como: provas de materialidade, indícios de autoria, comprovação da intenção do agente na conduta criminosa (dolo), delimitação geográfica dos locais para onde foram transferidos arquivos, entre outras informações. A proposta deste trabalho emprega Redes Neurais Artificiais (RNA), com o uso de *Multilayer Perceptron*, para classificar o fluxo de dados que utilizou criptografia e heurística em caso contrário. A RNA foi treinada e testada com fluxos de dados contendo pacotes gerados pelo *eMule*. Parte desse conjunto de treinamento e testes estava criptografado para que a RNA fosse capaz de classificar fluxos de dados independentemente do conteúdo dos pacotes estarem cifrados ou não. Desta forma, este trabalho contribui para a obtenção de informações de relevância pericial, as quais serão utilizadas durante a persecução penal. Como resultado experimental, foram detectados 100% dos pacotes do conjunto de teste não criptografado do *eMule*, sendo que 86,03% do tráfego criptografado foi identificado pela RNA. Os resultados experimentais alcançados demonstram a viabilidade da utilização de RNA para a identificação de tráfego do *eMule*.

ABSTRACT

EMULE TRAFFIC CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

Author: RODRIGO LANGE
Advisor: Prof[a]. Dr[a]. CÉLIA GHEDINI RALHA
Programa de Pós-graduação em Engenharia Elétrica
Brasília, November of 2011

This research presents the development of a method to identify network traffic data generated by the *eMule* peer-to-peer application. Upon this identification, forensic important artifacts may be obtained, such as: materiality evidence, authorship inkling, proof of intention in the course of the criminal behavior, geographical boundaries to where files have been transferred, among other information. The proposed system uses Artificial Neural Networks (ANNs) with Multilayer Perceptron in order to classify the data flow that was encrypted and heuristics in other case. The ANN has been trained and tested with network traffic containing packets generated by *eMule*. Part of the training set was encrypted, with the objective of being able to classify the data, being the content of the packets cyphered or not. The contribution of the presented work is to obtain information of forensic relevance, which will be used throughout litigation. Experimental results demonstrate the viability to use ANNs to identify *eMule* network traffic data, since 100% of not encrypted and 86,03% of encrypted *eMule* traffic were detected.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

ARPA *Advanced Research Projects Agency*, ou Agência de Pesquisas em Projetos Avançados. 47

ARPANet *Advanced Research Projects Agency Network*, ou Rede da Agência de Pesquisas em Projetos Avançados. 47, 48

CP Código Penal Brasileiro - Decreto-Lei nº 2.848, de 07 de dezembro de 1940. 26, 33

CPC Código de Processo Civil Brasileiro - Lei nº 5.869, de 11 de janeiro de 1973. 31

CPP Código de Processo Penal Brasileiro - Decreto-lei nº 3.689, de 3 de outubro de 1941. 30, 31

CTI Coordenação de Tecnologia da Informação. 83

DARPA *Defense Advanced Research Projects Agency*, ou Agência de Pesquisas em Projetos Avançados de Defesa. 35, 47

DFI *Deep Flow Inspection*, ou Inspeção Profunda em Fluxos. 61

DFRWS *Digital Forensic Research Workshop*. 12

DHT *Distributed hash table*, ou Tabelas *hash* distribuídas. 68, 78

DNA *Deoxyribonucleic Acid*, ou Ácido Desoxirribonucleico. 11

DNS *Domain Name System*. 57

DPF Departamento de Polícia Federal do Brasil. 2, 3, 83, 94

DPI *Deep Packet Inspection*, ou Inspeção Profunda em Pacotes. 61, 81

eD2k *eDonkey2000 network*. 7, 24, 52, 53, 67, 68, 76–79, 85, 104

FTP *File Transfer Protocol*. 48, 49, 57, 60, 65, 81, 94, 99, 103

HTTP *Hypertext Transfer Protocol*. 48, 49, 57, 60, 65, 72, 81, 83, 94, 99, 103

HTTPS *Hypertext Transfer Protocol Secure*. 57, 60

IA Inteligência Artificial. 8, 34–36, 54, 59, 63–66, 80, 106

IANA *Internet Assigned Numbers Authority*. 55, 57, 90

IDS *Intrusion Detection Systems*, ou Sistema de Detecção de Intrusão. 59, 60

IP *Internet Protocol*. 3, 4, 27, 50, 61, 64, 65, 69, 72, 83, 84, 90, 99, 101

ISP *Internet Service Provider*, ou Provedor de Acesso à Internet. 4, 5, 7, 27, 28, 53, 55–57, 98, 104, 105, *Veja em Glossário: internet service provider*

KAD *Kademlia network*. 7, 52, 67, 68, 76–79, 85, 104, 105

MLP *Multilayer Perceptron*. 64, 65, 81, 82, 84, 85, 93, 105, 106

NAT *network address translation*. 57, 58

P2P *peer-to-peer*. 1–3, 5–8, 15, 23, 24, 29, 47–67, 69, 76, 77, 80, 81, 83–85, 88, 92, 95, 98, 102, 104, 106

RFC *Request for Comments*, ou Pedidos de Comentários. 48

RIAA *Recording Industry Association of America*. 67

RNA Rede Neural Artificial. 6–8, 34, 36, 37, 39–41, 46, 65, 80, 82, 84, 85, 88–93, 96–98, 104–106

SMTP *Simple Mail Transfer Protocol*. 48, 49, 60, 83

SSH *Secure Shell*. 60

SWGDE *Scientific Working Group on Digital Evidence*. 15

TCP *Transmission Control Protocol*. 7, 48, 55, 58, 61, 63, 68, 69, 75, 76, 78, 83, 88–90, 95, 96, 99

TI Tecnologia da Informação. 14, 20

UDP *User Datagram Protocol*. 7, 55, 58, 61, 75, 78, 83, 88, 90, 95, 96, 99, 105

WWW *World Wide Web*. 48

Capítulo 1

Introdução

O processo de democratização do acesso às tecnologias da informação possibilitou a utilização da Internet por parte significativa da população brasileira. Segundo o Internet World Stats (2011), em março de 2011 o Brasil contava com mais de 75 milhões de pessoas com acesso à rede mundial. Segundo esta referência, em âmbito global, aproximadamente 30% da população mundial tem acesso à Internet.

Infelizmente, a disseminação do uso da Internet trouxe, além de enormes benefícios, graves problemas sociais. Yar (2006) cita como exemplo o crescente número de crimes praticados com a utilização de computadores través da rede mundial.

Dentre esses crimes, conforme ressaltado por Tanenbaum e Wetherall (2010), há a utilização da Internet como meio de troca ilícita de dados entre computadores, principalmente com o emprego de comunicação não-hierárquica do tipo *peer-to-peer* (P2P).

A grande utilização de sistemas P2P pode ser observada no estudo da empresa alemã Ipoque (2009), o qual identificou que mais de 65% dos dados que trafegaram pela Internet na América do Sul durante o período de 2008 a 2009 foi gerado por aplicativos P2P. Em outras regiões do globo esse percentual também foi elevado, como exemplo na Europa onde, de acordo com a região, esse índice variou de 44,77% a 69,96% e, na África, de 42,51% a 65,77%.

Dentre os aplicativos P2P utilizados para a troca de arquivos na Internet, o *eMule* se destaca por ser um dos programas P2P mais utilizado no mundo. Segundo pesquisa realizada em 2008, pelo IBOPE/NetRatings, apresentada em Calazans (2008) (*apud* de Oliveira e da Silva (2009)), esse aplicativo estava presente em 17,2% dos computadores localizados no Brasil. De acordo com Ipoque (2009), o *eMule* foi responsável por aproximadamente 17% do tráfego P2P na América do Sul durante o período de 2008

a 2009, tendo milhões de usuários conectados diariamente compartilhando aproximadamente um bilhão de arquivos.

O compartilhamento de arquivos através de sistemas P2P pode ser legítimo como, por exemplo, no caso de uma empresa que necessita disponibilizar arquivos de forma simples e de baixo custo. Entretanto, também pode ser ilícito, como no caso de distribuição de material envolvendo a exploração sexual de crianças e adolescentes. Conforme Taylor et al. (2010), a prática desse crime não é novidade, mas com a utilização da rede mundial, foi fortemente potencializada. Os arquivos podem ser transferidos de forma fácil e eficiente com a utilização de aplicativos P2P, tornando mais frequente a ocorrência de atividades criminosas que transferem grande quantidade de dados.

De acordo com Yar (2006), a distribuição de material relacionado à pornografia infantojuvenil tem, na Internet, o principal meio de divulgação. Segundo Pinto (2009), a disponibilização e distribuição de material com imagens de conteúdo erótico ou pornográfico, envolvendo crianças ou adolescentes, movimenta bilhões de dólares por ano e cria um mercado de produção de fotos, vídeos, turismo sexual e tráfico de menores.

No Brasil, conforme Nucci (2009), a principal previsão jurídica que criminaliza a exploração sexual de crianças e adolescentes está nos artigos 240 e 241 da Lei nº 8.069/90, Estatuto da Criança e do Adolescente (Brasil, 1990).

De acordo com o inciso I do § 1º do artigo 144 da Constituição da República (Brasil, 1988) [1], quando existir a internacionalidade do delito, a atribuição para apuração é do Departamento de Polícia Federal do Brasil (DPF), sendo atribuição da polícia civil em caso contrário. A transmissão de dados para outros países ocorre comumente quando há a disponibilização e distribuição desse tipo de material por redes P2P na Internet.

A utilização ilegal de aplicativos P2P gera graves problemas sociais, afetando profundamente a sociedade. Entidades estatais como o Ministério Público, Polícias estaduais e federal, Conselhos tutelares dentre outras, além de organizações sociais como a Ordem dos Advogados do Brasil (OAB), SaferNet, Fundo das Nações Unidas para a Infância (UNICEF) e *Internet Watch Foundation*, dentre outras, encontram-se engajadas na busca por formas de reduzir e punir essas condutas criminosas. Para tanto, são realizadas denúncias e investigações que resultam na deflagração de um número cada vez maior de operações policiais e no cumprimento de centenas de mandados de busca e apreensão. Como resultado dessas operações, um número considerável de mídias de armazenamento computacional é encaminhado para a realização de perícia.

[1] I - apurar infrações penais contra a ordem política e social ou em detrimento de bens, serviços e interesses da União ou de suas entidades autárquicas e empresas públicas, assim como outras infrações cuja prática tenha repercussão interestadual ou internacional e exija repressão uniforme, segundo se dispuser em lei;

Em virtude da grande utilização, o *eMule* é um dos principais aplicativos P2P monitorados por forças policiais, sendo alvo de diversas operações para coibir a distribuição de material envolvendo a exploração sexual de crianças e adolescentes.

No âmbito do DPF, dois peritos criminais federais, com o objetivo de monitorar os arquivos compartilhados nas redes utilizadas pelo *eMule*, desenvolveram o aplicativo *EspiaMule* (Dalpian e Benites, 2007). O *EspiaMule* permite o registro dos clientes que estão disponibilizando os arquivos monitorados, separando-os por país e possibilitando a identificação do endereço do *Internet Protocol* (IP) e do identificador do usuário do *eMule* (*user hash*).

Várias operações foram realizadas pela Polícia Federal com a utilização do *EspiaMule*. Informações referentes a algumas dessas operações são apresentadas na Tabela 2.1 (pág. 28). Como exemplo, somente nas operações Carrossel I e II, deflagradas, respectivamente, em 2007 e 2008, foram apreendidos aproximadamente 300 discos rígidos e 3000 discos ópticos (CDs e DVDs), com a prisão ou indiciamento de mais de 500 pessoas, tanto no Brasil quanto no exterior. Outras informações sobre a execução do *EspiaMule* em 2008 e o resultado da Operação Carrossel II são apresentadas na Tabela 2.2 (pág. 29).

Também foram criados aplicativos para o monitoramento de outras redes P2P como, por exemplo, o *Wyoming ToolKit* (WTK), desenvolvido por forças policiais dos Estados Unidos e que monitora a rede *Gnutella* (Wyoming DCI ICAC, 2008). Essa rede é acessada por diversos aplicativos tais como *BearShare*, *giFT*, *iMesh*, *LimeWire*, *Shareaza* dentre outros.

Segundo de Oliveira e da Silva (2009), comparando-se o funcionamento entre o WTK e o *EspiaMule*, enquanto o WTK não encontrou arquivos de pornografia infantojuvenil compartilhados por computadores localizados no Brasil, o *EspiaMule* localizou mais de 1100 computadores compartilhando esse tipo de material através da rede P2P. Esse estudo está consoante com os dados que apontam o *eMule* como sendo o aplicativo P2P mais utilizado no Brasil.

1.1 Motivação

A localização dos criminosos que disponibilizam e distribuem material envolvendo a exploração sexual de crianças e adolescentes através de sistemas P2P inicia-se com a identificação do endereço IP utilizado para a conduta ilícita. Esse endereço IP pode ter sido obtido de diversas formas como, por exemplo, pelo recebimento de denúncia,

por meio de investigação policial ou com o uso de programas de uso pericial como o *EspiaMule*.

De posse do endereço IP, as forças policiais solicitam autorização judicial para quebra do sigilo telemático do computador do suposto criminoso. Essa autorização judicial determina que o Provedor de Acesso à Internet (*Internet Service Provider* - ISP) preste informações cadastrais do terminal que utilizou o endereço IP, possibilitando a identificação do endereço do suspeito.

Entretanto, nem sempre as informações fornecidas pelo ISP são confiáveis. Essa falta de confiabilidade pode possuir várias causas como, por exemplo, ser decorrente da falta de controle adequado por parte do ISP ou pelo grande lapso temporal entre o momento do crime e o do levantamento dos dados cadastrais procedentes da morosidade para realização da denúncia, tramitação vagarosa do inquérito policial ou da ação penal, demora da decisão judicial, entre outros motivos.

Informações incorretas sobre o endereço do suspeito podem levar ao cumprimento de mandados de busca e apreensão em endereços incorretos, com o desperdício de recursos humanos e materiais, além de causar danos morais e materiais ao indivíduo que teve seu domicílio violado injustamente.

Além disso, mesmo que as informações prestadas pelo ISP sejam corretas e o mandado de busca e apreensão tenha sido efetivamente cumprido no endereço do suspeito, ainda assim a comprovação do crime pode ser complexa. Essa complexidade decorre da dificuldade (ou até mesmo inviabilidade) de realização da perícia no material apreendido, como no caso em que o criminoso formatou o disco rígido ou instalou um novo sistema operacional, ou quando é utilizada a criptografia ou esteganografia para proteção dos dados. A falta de comprovação da materialidade do delito pode levar à impunidade do criminoso, representando um grande dano à sociedade, com a perda da credibilidade do poder punitivo do Estado.

Outro aspecto pertinente é a comprovação da transnacionalidade ou não do crime, ou seja, se o envio de material contendo exploração sexual de crianças e adolescentes foi direcionado para outro país ou permaneceu em âmbito nacional. Esse fato é relevante, pois se a transferência limitou-se às fronteiras nacionais, a competência jurisdicional para julgar o processo penal é da Justiça Estadual, e da Justiça Federal, caso tenha ocorrido a transnacionalidade. Essa argumentação normalmente é utilizada pela defesa com o fim de protelar o processamento do suspeito.

Atualmente, a principal forma de obtenção da materialidade do delito de divulgação ou distribuição de arquivos envolvendo pornografia infantojuvenil é através da perícia,

realizada por peritos criminais estaduais ou federais. A demanda da perícia, executada sobre o material apreendido, como discos rígidos, discos ópticos, pendrives dentre outros, tem crescido, acompanhando o aumento do número de apreensões decorrentes de operações policiais de combate à exploração sexual de crianças e adolescentes.

Uma saída para o cumprimento de mandados de busca e apreensão em endereços incorretos, para as dificuldades de realização de perícia e de comprovação da materialidade, autoria, da intenção do agente (dolo) e transnacionalidade do crime poderia ser a utilização de interceptação telemática (gravação do fluxo de dados entre o computador do investigado e a Internet).

Caso seja autorizada judicialmente, a interceptação telemática poderia comprovar a autoria e materialidade em um momento anterior ao da busca e apreensão, diminuindo a carga dos órgãos periciais e diminuindo o tempo necessário para o trâmite processual. Mesmo que o tráfego do *eMule* seja criptografado, é possível identificar a utilização desse aplicativo pelo suspeito, indicando que o endereço fornecido pelo ISP possivelmente está correto.

Tendo em vista que não existem servidores do *eMule* localizados no Brasil, não é possível que seja solicitada autorização judicial para a realização de interceptação telemática desses servidores, pois a jurisdição brasileira somente pode atuar dentro do território nacional. Dessa forma, a interceptação telemática somente pode ser realizada sobre os clientes do *eMule*.

Segundo Gorge (2007), o monitoramento de redes P2P por parte das forças policiais é dificultada pela complexidade inerente a essas redes. Não se dispõe de ferramentas para a identificação e obtenção, dentre os dados interceptados, de informações sobre a utilização de aplicativos P2P como o *eMule*, que possibilitariam a comprovação da materialidade do crime.

De acordo com Taylor et al. (2010), as ferramentas periciais comumente utilizadas como o *Forensic Toolkit* (FTK) (AccessData, 2011), e o *EnCase* (Guidance, 2011), têm como foco a obtenção de evidências a partir do computador e não da interação entre computadores. Essas informações podem ser cruciais para a determinação da materialidade do crime, da autoria e do dolo em compartilhar material envolvendo a exploração sexual de crianças e adolescentes.

Portanto, é necessária a criação de mecanismos para a identificação do tráfego produzido por aplicativos P2P, principalmente o gerado pelo *eMule*, e a obtenção de informações desse fluxo de dados para a correta identificação dos criminosos antes do cumprimento do mandado de busca e apreensão. O emprego desse mecanismo de iden-

tificação, além de aumentar o conteúdo probatório do ilícito praticado, facilitando a comprovação da materialidade do crime, também evitaria danos causados a inocentes, que poderiam ter seu domicílio e intimidade violados injustamente e, por fim, poderia diminuir a carga de trabalho dos órgãos periciais.

Será utilizado um conjunto de heurísticas para a identificação do tráfego não criptografado do aplicativo *eMule*. Para a identificação do fluxo de dados criptografados, este trabalho irá empregar uma Rede Neural Artificial (RNA) pelos seguintes motivos:

- A capacidade de aprendizado através de exemplos e de generalização da RNA, produzindo saídas corretas para entradas que não haviam sido apresentadas anteriormente.
- A classificação realizada apenas com base nas portas utilizadas pela camada de transporte dos aplicativos P2P é ineficiente, em virtude da alocação aleatória de portas, utilizada pela maioria dos aplicativos P2P atuais.
- A classificação baseada apenas no conteúdo (*payload*) dos pacotes de dados, por meio de buscas por assinaturas, é ineficaz em virtude do emprego da criptografia pelos aplicativos P2P mais recentes. Segundo a pesquisa da empresa Ipoque (2009), em 2008/2009, mais de 16% do tráfego referente ao aplicativo *eMule* na Alemanha eram criptografados.
- A classificação pelo comportamento do fluxo de dados é altamente dependente da rede utilizada, pois mudanças como o tamanho do pacote e do tempo de chegada entre eles diminui a taxa de identificação.

A principal contribuição deste trabalho é se concentrar na identificação dos fluxos que apresentem maior relevância pericial. Esse enfoque é diferente do encontrado em outros estudos sobre classificação de fluxos de dados, que têm por prioridade a identificação do maior número de fluxos ou da maior quantidade de dados transferidos, sendo considerado irrelevante a obtenção de informações de cunho penal. Com a correta identificação dos fluxos do *eMule*, torna-se possível a obtenção de informações pertinentes para a comprovação da materialidade, da autoria e do dolo para o cometimento de ilícitos penais envolvendo o *eMule*.

1.2 Objetivos

O principal objetivo deste trabalho é projetar uma RNA para a classificação de tráfego P2P criptografado das redes utilizadas pelo *eMule*, mais especificadamente a rede

eDonkey2000 network (eD2k) e *Kademlia network* (KAD), além de empregar heurísticas para identificar o tráfego sem criptografia. A correta identificação do fluxo de dados referente a essas redes permitirá obter informações para comprovar a autoria e a materialidade de delitos, facilitando a persecução penal e a consequente punição ao criminoso.

Além disso, a identificação de fluxo de dados irá auxiliar na realização da perícia que, em alguns casos, seria impossibilitada ou dificultada pela adoção de medidas para a proteção dos dados presentes no disco rígido do criminoso, como a criptografia ou esteganografia, evitando o desperdício de recursos humanos e computacionais, bem como do tempo necessário a sua realização. Também será reduzida a demanda por perícias, já que os elementos probatórios podem ser obtidos antes do cumprimento do mandado de busca e apreensão.

Por fim, poderá ser evitado o cumprimento de mandados de busca e apreensão em endereços de pessoas inocentes, por falhas das informações prestadas pelos ISPs, pois mesmo que o *eMule* utilize criptografia e impeça a identificação dos dados trafegados pela rede, será possível identificar se tal aplicativo está sendo utilizado pelo suspeito.

Para que se possa atingir o objetivo principal deste trabalho, foram definidos três objetivos específicos:

1. fornecer uma maneira de identificar o fluxo de dados das redes utilizadas pelo aplicativo P2P *eMule*, dentre os dados obtidos em uma interceptação telemática, mesmo que o *eMule* esteja utilizando criptografia para a proteção do *payload*;

2. obter informações que permitam à equipe de investigação ter certeza de prática criminosa, comprovando a materialidade e o destino dos dados transferidos;

3. proporcionar elementos que permitam a comprovação da autoria de delitos.

1.3 Metodologia

A revisão bibliográfica de redes P2P e redes neurais forneceu os elementos necessários para a identificação da metodologia a ser seguida:

1. Serão selecionados informações relevantes do fluxo de dados para o treinamento da RNA como, por exemplo, protocolo de transporte utilizado (*Transmission Control Protocol* (TCP) ou *User Datagram Protocol* (UDP)), *payload*, tamanho do pacote, entre outras.

2. A RNA treinada e o conjunto de heurísticas serão aplicados em arquivos de captura de tráfego real com o objetivo de identificação de fluxos de dados do *eMule*, obtendo, sem necessidade de busca e apreensão, elementos que permitam maior certeza da ocorrência de atos ilícitos, além da obtenção de provas de autoria do delito cometido, facilitando a persecução penal.

A revisão bibliográfica também permitiu o levantamento de informações que possibilitaram a identificação de artefatos periciais do *eMule* (Lange e Ralha, 2011).

1.4 Organização da Dissertação

O restante deste trabalho se divide em duas partes: nos Capítulos 2, 3 e 4 são apresentados os conceitos e técnicas utilizados nesta dissertação, obtidos através de levantamento bibliográfico. Na segunda parte, composta pelos dois últimos capítulos (5 e 6), são apresentados o modelo proposto, os resultados obtidos nos testes realizados e as conclusões decorrentes da análise destes experimentos. De forma mais detalhada, a presente dissertação está organizada da seguinte forma:

- No Capítulo 2 são apresentadas informações referentes às Ciências Forenses e à Informática Forense pertinentes ao presente trabalho, incluindo comentários sobre crimes cibernéticos e perícias em redes de computadores.
- O Capítulo 3 contém conceitos relacionadas à Inteligênica Artificial (IA) e RNAs, tendo por objetivo possibilitar um entendimento do modelo proposto, dos experimentos realizados e as conclusões, apresentados nos capítulos subsequentes.
- O Capítulo 4 apresenta informações relacionadas a redes P2P e dados específicos sobre o aplicativo *eMule*. Também são explanadas técnicas utilizadas para a identificação do fluxo de dados relacionados a aplicativos P2P.
- No Capítulo 5 são expostos, de forma detalhada, o modelo proposto e os experimentos realizados.
- O Capítulo 6 apresenta a análise dos resultados alcançados e as conclusões obtidas na presente dissertação, bem como as sugestões de trabalhos futuros.

Capítulo 2

Ciência Forense e Informática Forense

Neste capítulo será apresentada uma breve revisão sobre as áreas de estudo da Ciência Forense e da Informática Forense. Na Seção 2.1 serão apresentadas informações sobre a Ciência Forense, na Seção 2.2 são apresentadas definições da área de Informática Forense e na Seção 2.3 são mencionadas informações sobre perícias em redes de computadores. Também são apontados elementos sobre crimes cibernéticos na Seção 2.4 e alguns aspectos legais relativos à atuação do perito na Seção 2.5. Por fim, na Seção 2.6, é apresentado o funcionamento da persecução penal no Brasil

2.1 Ciência Forense

O termo ciência indica o método científico utilizado para entender e descrever o universo que nos cerca, através da observação de padrões que possibilitem a criação de leis gerais (Inman e Rudin, 2000).

Em relação ao vocábulo *forense* (em inglês *forensics*), esse termo, segundo o *The American heritage dictionary of the English language* (Houghton Mifflin Company, 2001), significa a utilização de ciência e tecnologia para estabelecer fatos em tribunais cíveis e criminais.

Hankins et al. (2009) sugere que a Ciência Forense é utilizada, principalmente, para preservar, analisar e apresentar evidências em um sistema jurídico como parte de um processo civil ou penal.

Segundo Casey (2009), o método científico pode ser resumido da seguinte forma: chegar a uma conclusão objetiva de uma maneira repetível. A utilização de métodos que possam ser reproduzidos posteriormente é uma garantia para o poder punitivo do Estado e também uma garantia ao acusado, sendo o baluarte contra conclusões incorretas.

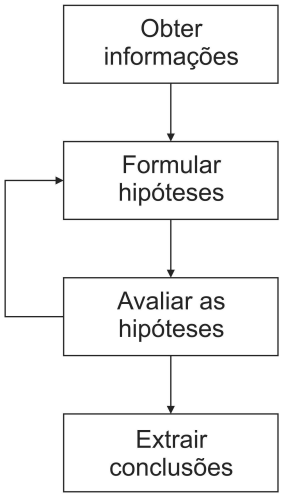


Figura 2.1: Aplicação do método científico na área pericial (adaptado de Casey, 2009)

Casey (2009) apresenta uma das possíveis formas de aplicação do método científico na área pericial. O presente trabalho aplica esse método para obtenção de informações periciais. As etapas, apresentadas sucintamente na Figura 2.1, podem ser descritas da seguinte forma:

1. Obter informações e fazer observações - fase normalmente chamada de exame pericial e envolve a verificação da integridade e autenticidade da evidência, bem como a análise dos dados (recuperação de arquivos apagados, seleção das informações relevantes, obtenção dos metadados entre outros). Essa fase não está limitada às evidências digitais, podendo ser complementada por entrevistas, depoimentos e outros materiais informativos.

2. Formular uma hipótese para explicar as observações - são elaboradas algumas explicações para os dados encontrados nas evidências. Embora as conjecturas sejam influenciadas pelo conhecimento e experiência do perito, ele deve se ater aos fatos, evitando noções preconcebidas.

3. Avaliar a hipótese formulada - várias previsões irão fluir naturalmente de qualquer hipótese (se a hipótese é verdadeira, então pode-se encontrar determinada informação nas evidências) e é função do perito verificar se essas expectativas

são efetivamente encontradas nas evidências, confirmando (ou não) a hipótese. É crucial considerar outras explicações e incluir testes para tentar refutar as outras hipóteses. Se a hipótese inicial não for comprovada, deve-se revisá-la (ou descartá-la) e realizar outros testes.

4. Extrair conclusões e comunicar os resultados - quando uma explicação provável para os eventos relacionados a um crime for comprovada, os peritos devem encaminhar o resultado dos exames para a pessoa ou órgão competentes.

Para Cohen (2010), na maioria das disciplinas forenses, a metodologia científica consiste em quatro elementos básicos: (1) estudo das teorias, métodos e sua base experimental anteriores e atuais; (2) identificação das inconsistências entre as teorias atuais e o resultados de experimentos repetíveis; (3) criar novas teorias e realizar experimentos para testar essas teorias; e (4) publicar os resultados.

Dessa forma, a simples análise das hipóteses por parte do investigador, sem a aplicação do método científico e a validação inerente a esse método, aumenta a chance de erros, pois a tendência é que a análise seja propensa em favor da hipótese criada pelo próprio investigador (Casey, 2009).

Palmer (2001) lembra que os avanços científicos levam algum tempo até que sejam amplamente aceitos nos tribunais, necessitando que os métodos utilizados sejam testados de forma rigorosa através da análise científica. Esse cuidado é necessário, pois as provas podem levar à restrição ou eliminação de liberdades individuais em uma sentença judicial. Palmer cita, como exemplo, que o primeiro caso onde foi utilizado o ácido desoxirribonucleico (DNA) como evidência de um crime ocorreu em 1987, ou seja, somente após dois anos do desenvolvimento da técnica de identificação de perfis de DNA.

2.2 Definições na área de Informática Forense

Nesta seção, serão expostas algumas definições de termos relacionados à Informática Forense. Inicialmente serão exibidas considerações acerca da tradução de termos comumente encontrados na literatura.

Hoelz (2009) afirma que a tradução mais adequada do termo *computer forensics* é perícia em computadores, e não forense computacional, um termo muito utilizado comercialmente no Brasil. O referido autor também propõe que perícia em informática (um termo um pouco mais amplo que perícia em computadores) é análogo ao termo

computer forensics e a área de pesquisa é a Informática Forense, tradução do termo *Digital Forensic Science.*

De acordo com Huebner et al. (2003), no campo de investigação de crimes relacionados à informática existe carência de definições amplamente aceitas na área ou, pelo menos, existem diferentes definições que variam de acordo com a área de interesse. Em decorrência dessa ausência de padronização, existe uma falta de direção clara e de apoio apropriado em seu desenvolvimento. Existem diversos guias de melhores práticas ou recomendações de diversas fontes, mas nenhum padrão internacional amplamente aceito.

Ainda, estes autores consideram irrealista acreditar que tal padrão será criado em um futuro próximo. Entretanto, mesmo que não exista um consenso universal sobre os padrões, são encontradas diversas definições propostas por pesquisadores da área. Perícia em computadores pode ser definida como:

- Farmer e Venema (1999) definiram perícia em computadores como sendo a coleta e análise de dados, livre de distorções ou preconceitos, que permitam a reconstrução de informações ou de acontecimentos que ocorreram em um sistema no passado.
- Segundo Caloyannides (2001), o termo perícia em computadores é definido como sendo o conjunto de técnicas e ferramentas utilizadas para encontrar evidências em um computador.

A perícia em informática pode ser definida como:

- McKemmish (1999), do *Australian Institute of Criminology*, apresentou uma definição de perícia em informática como sendo o processo de identificar, preservar, analisar e apresentar evidências digitais de uma forma juridicamente aceitável.
- Reith et al. (2002) afirmam que o termo perícias em informática originou-se como um sinônimo de perícias em computadores, mas com o passar do tempo sua definição expandiu-se para incluir a perícia de todas as tecnologias digitais.
- Conforme Ieong (2006), perícia em informática é um conjunto de tarefas e processos desempenhados em uma investigação relacionada ao uso dessa tecnologia.

Durante o *First Annual Digital Forensic Research Workshop* (DFRWS), realizado em Nova Iorque, nos Estados Unidos, em 2001, foi apresentada uma definição de informática forense. O presente trabalho irá utilizar essa definição de informática forense, exposta abaixo conforme apresentada por Palmer (2001):

> A utilização de métodos cientificamente estabelecidos e comprovados para a preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação de evidências digitais provenientes de fontes digitais para fins de facilitar ou favorecer a reconstrução dos acontecimentos considerados criminosos, ou auxiliar na prevenção de ações não autorizadas que sejam prejudiciais para as operações previstas.

De acordo com Carrier (2005), as evidências digitais são determinados pela mídia física, pelo sistema operacional, pelo sistema de arquivos e pelos aplicativos do usuário.

Carrier e Spafford (2003) afirmam que o computador deve ser tratado como uma nova cena de crime, e não como uma substância que deve ser identificada. O computador deve ser visto como uma porta que leva a investigação a um novo cômodo. Assim, quando visto dessa forma, os mesmos princípios que são utilizados para tratar uma joalheria que foi roubada podem ser utilizados para tratar um servidor onde informações de cartões de crédito foram copiados – mesmo que a tecnologia necessária para tal análise seja substancialmente diferente.

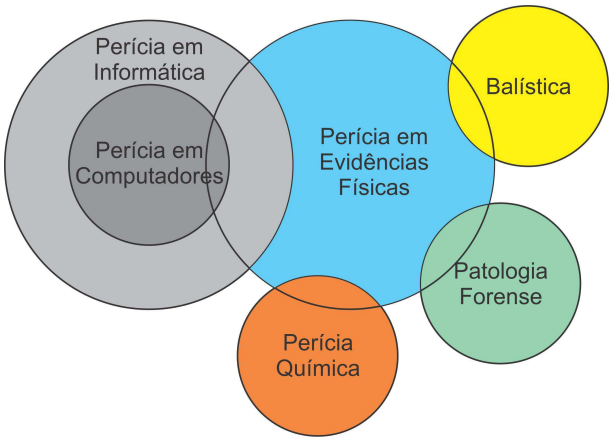


Figura 2.2: A sobreposição entre perícias em informática e em computadores, perícia em evidências físicas e algumas outras áreas forenses (adaptado de Grobler e Louwrens, 2006)

Segundo Grobler e Louwrens (2006), para concluir uma investigação com sucesso, as averiguações devem incluir, além das evidências digitais, também as evidências físicas. A Figura 2.2 apresenta a sobreposição entre perícias digitais, perícia em evidências físicas e algumas outras áreas forenses.

Durante as primeiras investigações em crimes relacionados a computadores, ficou claro que se aplicam nessa área os mesmos princípios básicos que se aplicam a crimes em que o computador não foi utilizado (Huebner et al., 2003).

Os locais de crimes cibernéticos possuem algumas semelhanças com os locais de crimes tradicionais (Palmer, 2001). Um princípio fundamental em locais de crime, de acordo com Huebner et al. (2003), é o *Princípio da Troca de Locard*, criado por Edmond Locard, diretor do primeiro laboratório policial forense, localizado em *Lyon*, na França. Dale e Becker (2007) apresentam esse princípio:

> Onde quer que ele pise, o que ele toca, tudo o que ele deixa, mesmo que inconscientemente, servirá como uma testemunha silenciosa contra ele. Não apenas as suas impressões digitais ou suas pegadas, mas o seu cabelo, as fibras das suas roupas, o vidro que ele porventura quebre, a marca da ferramenta que ele use, a tinta que ele arranhe, o sangue ou sêmen que deixe – tudo isto, e muito mais, testemunhará contra ele. Esta é a prova que não se esquece. Não se confunde com a excitação do momento. Não é ausente como as testemunhas humanas são. Constitui, por si, uma evidência factual. A evidência física não pode estar errada, não pode cometer perjúrio, não pode se ausentar. Somente sua interpretação pode estar errada. Apenas a falha humana de encontrá-la, estudá-la e entendê-la corretamente é que pode diminuir seu valor.

Embora existam semelhanças entre os crimes tradicionais e os crimes cibernéticos, também existem diversas características que diferenciam essas atividades ilícitas. Bryant (2008) apresenta seis dessas características:

1. Diferenças espaciais e temporais - os crimes cibernéticos não estão limitados ao tempo e espaço da mesma forma que os crimes tradicionais. Hoje, um fraudador que utiliza a Internet não precisa estar perto da vítima – o ataque pode ser realizado de qualquer parte do mundo, não sendo limitado por fronteiras geográficas. Além disso, o ataque pode durar menos de um segundo, mas os efeitos podem perdurar por dias, meses e até mesmo anos.

2. Economia de escala - a Tecnologia da Informação (TI) pode disseminar informações de forma ampla, repetitiva e barata. Um exemplo é o envio de programas maliciosos, que têm como objetivo capturar credenciais bancárias, através de milhões de mensagens de correio eletrônico. Assim, pela facilidade de comunicação, mesmo que um percentual muito baixo se torne vítima desse crime, o número de pessoas prejudicadas por tal conduta é alto.

3. Anonimato - os crimes cibernéticos, principalmente aqueles que utilizam a Internet, criam uma ideia de anonimato, onde as identidades podem ser facilmente

falsificadas. Nos crimes tradicionais, onde normalmente é necessária a presença do criminoso, o risco de identificação é muito maior.

4. Mundos virtuais - a natureza virtual do crime pode levar à desinibição por parte do criminoso. Por exemplo, muitas pessoas não tomam cuidado para ocultar que baixaram arquivos com infração de direitos autorais através de redes P2P (Yar, 2006).

5. Atraso legislativo - o desenvolvimento de crimes ocorre de maneira rápida e essa velocidade não é acompanhada pelo poder legislativo. A edição de leis pode levar vários anos para tipificar um determinado crime que tenha sido criado recentemente ou assumido uma nova forma.

6. Dificuldades investigativas - a natureza do crime digital levanta diversos desafios para a investigação. Por exemplo, a utilização de esteganografia e de criptografia podem dificultar e até inviabilizar a análise dos dados protegidos. Além disso, a grande quantidade de elementos técnicos dificulta o entendimento, por parte dos investigadores, dos locais onde as evidências do delito podem ser encontradas.

Carrier e Spafford (2003) apresentam a definição de evidências e locais de crime, fazendo uma correlação entre os crimes tradicionais com os crimes digitais:

- Evidência física – são objetos físicos que permitem afirmar que um crime foi cometido, a existência de um vínculo entre o crime e a vítima ou entre o crime e seu autor. Exemplos de evidências físicas são o computador, um celular e um CD-ROM.

- Evidência digital – são dados digitais que permitem afirmar que um crime foi cometido, a existência de um vínculo entre o crime e a vítima ou entre o crime e seu autor. Exemplos de evidências digitais são os dados encontrados na memória, no disco rígido ou em um telefone celular.

- Local de crime físico – é o ambiente físico onde são encontradas evidências de um crime.

- Local de crime digital – é o ambiente virtual criado por *software* ou *hardware* onde são encontradas evidências digitais de um crime.

De acordo com Whitcomb (2002), em 1998 o *Scientific Working Group on Digital Evidence* (SWGDE) definiu evidência digital como sendo qualquer informação de valor probatório que é armazenada ou transmitida de forma binária. Posteriormente, o termo

"binária" foi substituído por "digital". Evidência digital inclui áudio e vídeo digitais, dados em celulares e aparelhos de fax, dentre outros.

Para Palmer (2001), a análise forense é utilizada, principalmente, em três áreas distintas: (1) área policial, cujo objetivo principal é a persecução penal e é realizada após a ocorrência de um crime; (2) área militar, cuja principal finalidade é a continuidade das operações; e (3) área de indústria e comércio, que tem como objetivo primário a disponibilidade dos serviços. Tanto na área militar quanto na área de indústria e comércio, a persecução penal é, no melhor dos casos, um objetivo secundário, estando o foco na identificação rápida de atividades anômalas em redes e computadores individuais.

As áreas de interesse e pesquisa em Informática Forense, conforme adaptados de Palmer (2001) por Hoelz (2009) são apresentadas na Figura 2.3.

Figura 2.3: Áreas de interesse e pesquisa em Informática Forense (adaptado de Palmer (2001) por Hoelz (2009))

Palmer (2001) categoriza a perícia forense em três tipos:

1. Exames em mídias de armazenamento - exames em dispositivos de armazenamento de dados, tais como mídias ópticas (como CD, DVD e *Blu-ray*) e magnéticas (como discos rígidos, cartões de memória, *pen drives*, fitas, disquetes, dentre outros). Essas mídias podem ser encontrados em diversos dispositivos além de computadores, como em um *Personal Digital Assistant* (PDA), aparelhos de GPS, fax, impressoras, máquinas fotográficas, celulares, consoles de jogos eletrônicos, televisores e veículos (Baryamureeba e Tushabe, 2004);

2. Exames em programas - exames em aplicativos, como programas maliciosos (vírus, *trojans* e *keyloggers*), em sistemas gerenciais que acessam bancos de dados, dentre outros. Esses exames, normalmente de alta complexidade, são realizados, principalmente, com o emprego de engenharia reversa e análise comportamental, de forma dinâmica ou estática (Malin et al., 2008; Burji et al., 2010);

3. Exames em redes de computadores - exames realizados em fluxos de dados interceptados, mensagens de correio eletrônico, sítios da Internet, redes sem fio, análise de arquivos de histórico (arquivos de *log*), detecção de intrusão, dentre outros (Kaushik et al., 2010).

A área em que o presente trabalho tem como foco é a terceira categoria – exames em redes de computadores. Essa categoria de perícia é apresentada em maiores detalhes na Seção 2.3. O objetivo principal é a obtenção de evidências que possam comprovar e materialidade e apresentar indícios de autoria, que é a finalidade do inquérito policial. Maiores particularidades sobre essa peça investigativa são apresentadas na Seção 2.6.

Em relação às evidências, Ruibin et al. (2005) apresenta o conceito de "relevância para o caso". Esse conceito, que tem como objetivo auxiliar na valoração das evidências, é definido como sendo:

> a propriedade de qualquer fragmento de informação utilizada para medir sua capacidade de responder às perguntas investigativas ("quem", "o que", "onde", "quando", "por que" e "como") em uma investigação criminal.

A Figura 2.4 ilustra os níveis de relevância de uma evidência para o caso, conforme apresentado por Ruibin et al. (2005).

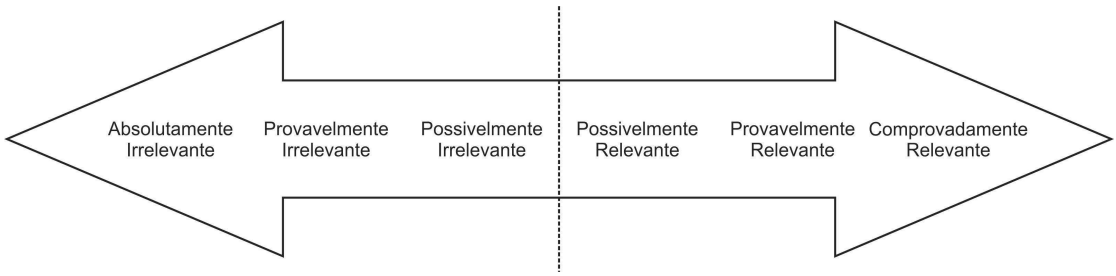


Figura 2.4: Níveis de relevância da evidência para o caso (adaptado de Ruibin et al., 2005)

Ao considerar a relevância de uma evidência para o caso, o perito pode focalizar sua atenção em um determinado subconjunto de arquivos, ignorando, temporariamente, aqueles considerados irrelevantes (Hoelz et al., 2009).

2.3 Perícias em Redes de Computadores

De acordo com Palmer (2001), a perícia em redes de computadores recebe diversas denominações, tais como *network forensics*, *remote forensics* e *cyberforensics*.

Diversos fatores, elencados a seguir, apontam que o monitoramento e análise de dados de redes de computadores (perícia em redes de computadores) será essencial para a atuação das forças policiais.

1. Ausência de limitações impostas por fronteiras geográficas, pois as informações podem atravessar via rede a jurisdição de diversos países. Segundo Huebner et al. (2003), as tecnologias de armazenamento de dados possibilitam que informações estejam em outros países, fora da jurisdição dos tribunais nacionais. Entretanto, de acordo com Wang (2010), esses dados continuam acessíveis aos criminosos através da Internet.

2. Aumento do uso da criptografia. Conforme Huebner et al. (2003), ferramentas que anteriormente possuíam distribuição restrita agora estão disponíveis gratuitamente na Internet para quem quiser utilizá-las como, por exemplo, o aplicativo *TrueCrypt*.

3. Segundo Beebe e Clark (2005), a capacidade de armazenamento de dados está crescendo de forma exponencial. Em um estudo posterior, Beebe (2009) afirma que, em alguns casos, a capacidade de armazenamento ultrapassa a marca dos petabytes.

4. Os dispositivos de armazenamento de dados serão mais difíceis de encontrar. Conforme Huebner et al. (2003), pequenos dispositivos de armazenamento de dados (e fáceis de ocultar ou destruir) de baixo custo, mas com grande capacidade de armazenamento, já são facilmente encontrados no mercado.

5. As perícias em informática são limitadas pela capacidade de processamento do perito. De acordo com Beebe e Clark (2005), com o crescimento do volume de dados, a quantidade de informações necessárias para serem examinadas pelo perito também cresce, diminuindo sua disponibilidade para analisar, de forma meticulosa, os dados encontrados.

6. Devem ser modificados alguns procedimentos de coletas de evidências. Segundo Huebner et al. (2003), alguns dados possuem um ciclo de vida curto e desaparecem, geralmente de forma irremediável. O ciclo de vida pode ser de apenas nanosegundos, se os dados residem nos registradores do computador, um pouco

mais se está na memória principal ou em enlaces de rede, e relativamente mais longo (de segundos a anos) quando reside no disco rígido. Para Farmer e Venema (1999), isso é geralmente chamado de Ordem de Volatilidade (*Order of Volatility*). Assim, conforme Huebner et al. (2003), o fluxo de dados através de conexões de rede, a não ser que seja continuamente monitorado, é irremediavelmente perdido.

Para enfrentar esses desafios, Pollitt (2010) afirma que deverão ser desenvolvidas ferramentas periciais automatizadas, dotadas de capacidades analíticas, possibilitando a identificação de itens importantes sem a necessidade de visualizar todo o conteúdo e que possam interpretar o conteúdo e o contexto.

Uma solução para esses desafios poderia ser a obtenção de elementos que pudessem comprovar a autoria e materialidade do crime em uma etapa anterior ao do cumprimento do mandado de busca e apreensão, sem que seja necessária a análise de todo conteúdo dos discos rígidos apreendidos. Um *framework* genérico para perícias em redes de computadores, conforme Kaushik et al. (2010), é apresentado na Figura 2.5. Esse *framework* é utilizado no presente trabalho.

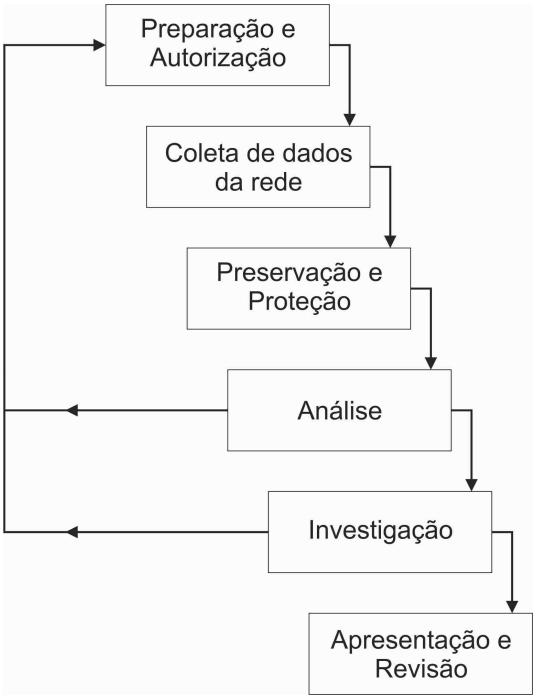


Figura 2.5: *Framework* genérico de perícias em redes de computadores (adaptado de Kaushik et al., 2010)

Uma definição de perícia em redes de computadores ("*network forensics*") é apresentada em Palmer (2001):

> Perícia em redes de computadores é o uso de técnicas cientificamente comprovadas para coletar, unir, identificar, examinar, correlacionar, analisar e

> documentar evidências digitais de múltiplas fontes que processam e transmitem dados digitais ativamente, com o propósito de descobrir fatos relacionados com o objetivo planejado ou a medida de sucesso de atividades não autorizadas que tenham por objetivo perturbar, corromper e comprometer componentes de sistemas, bem como fornecer informações para ajudar na resposta ou recuperação destas atividades.

Existe semelhança entre esta definição de perícias em redes de computadores e a área de Segurança em TI. Entretanto, o foco dessas duas áreas é diferente. O foco da Segurança da TI está na preservação da integridade, confidencialidade e disponibilidade (Ieong, 2006). Esses são os objetivos gerais dos sistemas computacionais, do ponto de vista da segurança da informação (Tanenbaum e Wetherall, 2010). Quanto à Perícia em Informática, segundo Ieong (2006), o foco está no reconhecimento, relevância e confiabilidade da evidência. Essas diferenças entre a Segurança em TI e a área de Perícias em Informática são apresentadas na Figura 2.6.

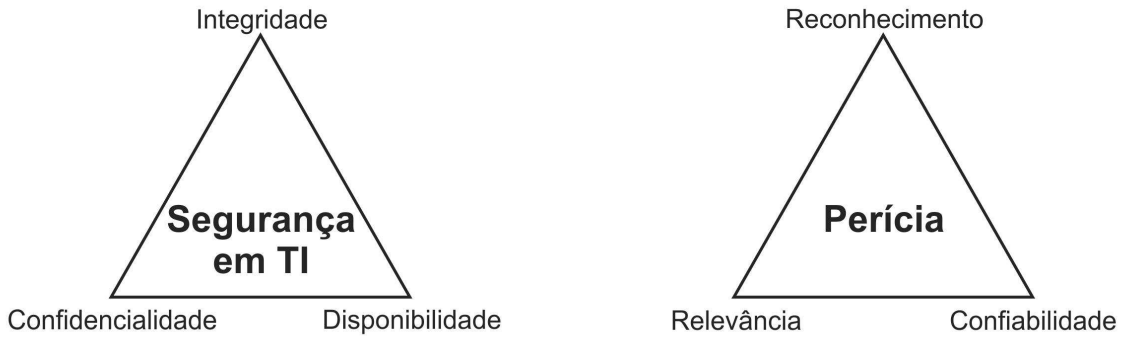


Figura 2.6: Diferenças entre a Segurança em TI e a área de Perícias em Informática (adaptado de Ieong, 2006)

No mesmo sentido, de acordo com Ruibin et al. (2005), existem algumas diferenças entre a área de perícia em informática e a área de Segurança em TI. A atuação da área de segurança é ampla, enquanto a área de perícias atua restrita ao caso que está sendo investigado e tem por objetivo reconstruir o cenário do crime. A atuação na área de segurança é de 24 horas por dia, e a da perícia inicia após a notícia de um crime (*notitia criminis*). Existem mais restrições na área da perícia, principalmente de ordem legal como, por exemplo, o horário de atuação.

Palmer (2001) também apresenta diferenças entre a área de segurança e a de perícia: (1) a área de segurança é preventiva e a de perícias, normalmente, é reativa; (2) a área

de segurança é acionada durante ou imediatamente após os incidentes de segurança, enquanto a perícia é acionada após os incidentes.

O presente trabalho é destinado à realização de perícias em redes de computadores. Maiores detalhes da implementação do presente trabalho serão apresentados nos Capítulos 5, 6 e 7.

2.4 Crimes Cibernéticos

A sociedade, em virtude de sua constante transformação, passou por diversas etapas. A última mudança profunda foi a passagem de um sistema baseado na produção de bens materiais para outro modelo, centrado na produção de informação. A sociedade contemporânea encontra-se na era do conhecimento, onde a informação desempenha um papel fundamental na produção de riqueza e na contribuição para o bem-estar e qualidade de vida das pessoas.

Essa mudança de paradigma causou uma revolução na sociedade, sendo marcada não somente por aspectos positivos, mas também por negativos. No final do Século XX surgiu uma nova modalidade de atividade ilícita que passou a utilizar-se dos recentes avanços tecnológicos, principalmente da Internet. Essa atividade criminosa recebe diversos nomes, como crimes cibernéticos, crimes eletrônicos, crimes virtuais, *cybercrimes* entre outros, em virtude da associação do termo *cyber* com o ambiente virtual disponibilizado pela Internet.

Recentemente, o termo crime cibernético (*cybercrime*) foi adotado pela mídia, comunidade acadêmica, forças policiais e governamentais para discutir e debater os problemas dos crimes relacionados ao emprego da tecnologia, em particular a Internet (Hunton, 2009). Para fins deste trabalho, utilizar-se-á a denominação crimes cibernéticos para esse tipo de delito.

A utilização de tecnologia por parte dos criminosos não é recente. Através da história, os avanços tecnológicos criaram novas oportunidades para atividades ilícitas, proporcionando um leque cada vez maior de possibilidades para mentes criativas (Bryant, 2008).

A conceituação desse tipo de delito não é uma tarefa trivial, em virtude das inúmeras possibilidades em que é utilizada a informática. Conforme Fafinski (2008), não existe uma definição clara e universalmente aceita do que seria crime cibernético. Hunton (2009) complementa que essa ausência de uma definição clara dificulta a identificação e quantificação da verdadeira escala desse tipo de delito. Ainda assim, é possível en-

contrar diversas definição de crime cibernético na literatura. Algumas dessas definições são apresentadas a seguir.

Para Bryant (2008), crime digital ou de alta tecnologia é aquele no qual foi utilizada tecnologia para facilitar a atividade criminosa. Segundo Reith et al. (2002), os crimes cibernéticos não são, necessariamente, novos crimes, pois podem ser crimes clássicos que exploram o poder proporcionado pelo computador e a acessibilidade de informações, principalmente através da Internet .

Huebner et al. (2003) apresentaram uma classificação para facilitar o entendimento dos crimes cibernéticos. Os autores afirmam que essa classificação apenas tem o propósito de auxiliar a compreensão dessa área, já que existem casos em que certas atividades criminosas podem pertencer a mais de uma área, ou serem difíceis de classificar em alguma das áreas descritas:

1. Crimes centrados no computador - a atividade criminosa tem por objetivo sistemas computacionais, redes, mídias de armazenamento ou outros dispositivos computacionais (como, por exemplo, perturbar o funcionamento de um sítio comercial da Internet e modificar seu conteúdo). Essas condutas podem ser vistas como novas ferramentas auxiliando uma nova categoria de crime;

2. Crimes auxiliados por computador - sistemas computacionais são utilizados como ferramentas para o auxílio de atividades criminosas onde a utilização de computadores não é estritamente necessária (como, por exemplo, distribuição de material contendo pornografia infantojuvenil). Essas condutas podem ser vistas como novas formas de se cometer crimes convencionais;

3. Crimes por computador incidentais - atividade criminosa onde a utilização de computadores é eventual (como, por exemplo, contabilidade informatizada utilizada para manter os registros financeiros de tráfico de drogas). Essas condutas podem ser vistas como a utilização de novas ferramentas para substituir ferramentas convencionais (livro de contabilidade substituído por um *software* contábil).

A condenação de um criminoso que pratica crimes cibernéticos é dificultada pela própria natureza deste delito. Para que seja instaurado um processo penal (eventualmente também um processo civil), normalmente é necessária a realização de um processo investigativo prévio. A definição de *investigação* é apresentado por Grobler e Louwrens (2006). Para esses autores, o processo para identificar o que deu errado, com a coleta de evidências suficientes para provar a causa de um evento, pode ser visto como uma investigação.

De acordo com Heiser e Kruse (2002), na esfera penal, o objetivo da investigação é o de processar, com sucesso, o criminoso, comprovando a causa de um crime e sua responsabilidade. A forma de condução da investigação é crucial para que a evidência seja aceita no tribunal.

O número de crimes cibernéticos continua a crescer, criando novas oportunidades para as atividades criminosas enquanto a tecnologia continua a evoluir. As forças policiais devem possuir uma gama abrangente de conhecimentos e capacidade técnica para enfrentar crimes que podem possuir abrangência global, principalmente através da Internet. De acordo com Hunton (2009), para que as investigações sejam conduzidas de forma consistente, existe a necessidade da alta capacitação técnica e do desenvolvimento de ferramentas que irão permitir o combate eficaz à criminalidade.

Entretanto, deve haver grande cuidado no desenvolvimento de novas técnicas e ferramentas. Segundo Marshall (2008), quase toda interação com um dispositivo técnico pode causar mudanças no estado desse dispositivo, eventualmente impactando na integridade da evidência e em sua aceitação no tribunal. Daeid e Houck (2010) afirmam que as ferramentas devem seguir uma metodologia reconhecida, serem testadas e funcionarem de forma precisa, confiável e repetível, além de terem sido revisadas.

Diversos tipos de crimes cibernéticos podem ser cometidos através de redes P2P, como infrações de direitos autorais, disseminação de programas maliciosos e disponibilização e distribuição de arquivos de imagem e de vídeos contende cenas de exploração sexual de crianças e adolescentes (Wolak et al., 2003; Finkelhor et al., 2006; Choo, 2008; Bessadok et al., 2009; Steel, 2009; Taylor et al., 2010; Latapy et al., 2011).

Dentre esses crimes, merece especial atenção a disponibilização e transferência de material contendo cenas de exploração sexual de crianças e adolescentes. Para que se possa identificar a abrangência desse crime, na Seção 2.4.1 serão apresentadas algumas considerações a respeito desse delito.

2.4.1 Compartilhamento de material pornográfico infantojuvenil

O combate ao compartilhamento e distribuição de arquivos de imagem e de vídeo contendo cenas de abuso sexual de crianças e adolescentes foi selecionado como um dos objetivos da Diretriz nº 8 do Terceiro Programa Nacional de Direitos Humanos do Brasil (PNDH-3) (Brasil, 2010). O PNDH-3 estabelece as diretrizes a serem seguidas

pelas políticas públicas em Direitos Humanos do Governo Federal. Dessa forma, o combate a este crime é uma das prioridades das forças policiais.

De acordo com o relatório da Interpol (2009), o crescimento do uso da Internet e o anonimato que a rede pode proporcionar, bem como avanços tecnológicos, como conexões de alta velocidade e câmaras digitais, têm facilitado a atuação de indivíduos com interesses sexuais em crianças para que possam gravar suas atividades, baixar arquivos e compartilhar imagens através da Internet. A exploração sexual infantojuvenil na Internet varia de poses para fotos até a gravação de vídeos de crimes sexuais brutais.

Para Latapy et al. (2011), a disponibilização em larga escala de material contendo pornografia infantojuvenil é um perigo mesmo a usuários que não buscam esse tipo de material (inclusive crianças e adolescentes), que podem ser expostos, involuntariamente, a esse conteúdo extremamente prejudicial. Neste particular, podem levar ao desenvolvimento de um interesse por pornografia infantojuvenil. Além disso, também tem um forte impacto na aceitação, pelo público, da pedofilia e a banalização de tal conteúdo.

Segundo Hughes et al. (2006), 1,6% das buscas e 2,4% das respostas na rede P2P Gnutella são relativas à pornografia ilegal. Na rede eD2k, acessada pelo *eMule*, esse cenário não é diferente. Latapy et al. (2011) apresentaram o resultado de um monitoramento, realizado em 2007 e 2009. Foram registradas centenas de milhões de buscas de arquivos por milhões de diferentes usuários. Dentre essas buscas, aproximadamente 0,25% contêm termos relacionados à pedofilia e 0,2% dos usuários realizaram buscas com pelo menos um desses termos. Isso significa que 1 em cada 400 buscas por arquivos contém termos relacionados à pornografia infantojuvenil — uma busca a cada 22 segundos em 2007 e uma em cada 33 segundos em 2009, por um total de mais de noventa mil diferentes usuários.

Em 2008, após realizarem buscas por arquivos contendo termos relacionados à pornografia infantojuvenil durante 140 dias, Bessadok et al. (2009) identificaram mais de dois milhões e setecentos mil arquivos diferentes, cujos nomes contêm termos relacionados à pornografia infantojuvenil.

Segundo o Wolak et al. (2003), mais de dois terços dos criminosos que cometeram crimes pela Internet contra menores de idades possuíam imagens de conteúdo pornográfico infantojuvenil e, destes, 83% possuíam imagens pornográficas de crianças entre 6 e 12 anos. Além disso, 16% das investigações de posse de material relativo à exploração sexual de crianças e adolescentes levou à identificação de pessoas que diretamente abusavam de crianças.

No Brasil, embora existam diversas normas legais que visam proteger a integridade sexual de crianças e adolescentes, não existe uma tipificação penal específica com o nome jurídico (*nomen iuris*) pedofilia. Dessa forma, como a pedofilia é uma doença caracterizada pela atração sexual por crianças ou adolescentes, nem todo pedófilo é um criminoso, desde que não coloque em prática essa atração sexual.

No entanto, o poder judiciário brasileiro tem entendido que somente a possibilidade de dano à imagem de crianças ou adolescentes já enseja uma punição de tal conduta. Nesse sentido, pode-se citar a deliberação da 5ª Turma do Superior Tribunal de Justiça (STJ), em decisão do Recurso Especial nº 617.221-RJ (Brasil, 2004):

> Para a caracterização do disposto no art. 241 do Estatuto da Criança e do Adolescente, "não se exige dano individual efetivo, bastando o potencial. Significa não se exigir que, em face da publicação, haja dano real à imagem, respeito à dignidade etc. de alguma criança ou adolescente, individualmente lesados. O tipo se contenta com o dano à imagem abstratamente considerada".

É inegável que uma das prioridades, em qualquer Estado Democrático de Direito, é assegurar o adequado desenvolvimento físico e psicológico de crianças e adolescentes (Nucci, 2008).

Em âmbito internacional, a Assembleia Geral das Nações Unidas (ONU) aprovou, em 20 de novembro de 1989, através da Resolução 44/25, a Convenção Internacional sobre os Direitos da Criança. Esse tratado internacional, que visa a proteção de crianças e adolescentes, principalmente em seus artigos 19 [1] e 34 [2], que obriga os estados signatários a protegerem a infância e adolescência de abuso, ameaça ou lesão à sua integridade sexual (Organização das Nações Unidas, 1989). O Brasil ratificou esse tratado em 24 de setembro de 1990.

[1] Artigo 19 - Os Estados Partes tomam todas as medidas legislativas, administrativas, sociais e educativas adequadas à proteção da criança contra todas as formas de violência física ou mental, dano ou sevícia, abandono ou tratamento negligente; maus tratos ou exploração, incluindo a violência sexual, enquanto se encontrar sob a guarda de seus pais ou de um deles, dos representantes legais ou de qualquer outra pessoa a cuja guarda haja sido confiada.

[2] Artigo 34 - Os Estados Partes comprometem-se a proteger a criança contra todas as formas de exploração e de violência sexuais. Para esse efeito, os Estados Partes devem, nomeadamente, tomar todas as medidas adequadas, nos planos nacional, bilateral e multilateral para impedir: a) Que a criança seja incitada ou coagida a dedicar-se a uma atividade sexual ilícita; b) Que a criança seja explorada para fins de prostituição ou de outras práticas sexuais ilícitas; c) Que a criança seja explorada na produção de espetáculos ou de material de natureza pornográfica.

No âmbito nacional, a proteção à criança e ao adolescente está prevista na Constituição da República (Brasil, 1988), em seu artigo 227 [3], e a proteção específica sobre a violência e exploração sexual está no parágrafo 4º desse mesmo artigo [4].

No Brasil, adota-se o critério cronológico na definição de criança e adolescente. A previsão legal está no artigo 2º da Lei nº 8.069, de 13 de julho de 1990 [5] (Estatuto da Criança e Adolescente) (Brasil, 1990). Assim, criança é aquela que possui menos de doze anos e adolescente entre doze a dezoito anos.

A idade mínima para que um indivíduo possa ter desenvolvimento psicológico suficiente para consentir na prática atos sexuais varia conforme o país, cultura e critérios adotados. No Brasil, é crime manter relação sexual com menores de 14 anos, por considerar o desenvolvimento psicológico incompleto para ter plena consciência de seus atos, conforme o artigo 217-A do Código Penal (CP) [6] (Brasil, 1940).

A relação sexual consentida com pessoas entre 14 e 18 anos deixou de ser considerado crime com a edição da Lei nº 12.015, de 07 de agosto de 2009, somente prevendo a possibilidade de crime nos casos de favorecimento da prostituição ou outra forma de exploração sexual, de acordo com o inciso I, parágrafo 2º do artigo 218-B [7] dessa lei.

A maior parte da legislação penal referente à exploração sexual de crianças e adolescentes está na Lei nº 8.069, de 13 de julho de 1990 (Estatuto da Criança e Adolescente) (Brasil, 1990), modificado recentemente pela edição da Lei nº 11.829, de 25 de novembro de 2008. Essas modificações ampliaram as possibilidades de punição, preenchendo determinadas lacunas e através do aumento das penas em algumas condutas criminosas, conferindo modernidade ao Estatuto da Criança e do Adolescente (Nucci, 2009).

Nucci (2008) apresenta os principais tipos penais (crimes) relacionados ao uso da informática, presentes no Estatuto da Criança e do Adolescente. Ressalta-se que apenas

[3] Art. 227. É dever da família, da sociedade e do Estado assegurar à criança, ao adolescente e ao jovem, com absoluta prioridade, o direito à vida, à saúde, à alimentação, à educação, ao lazer, à profissionalização, à cultura, à dignidade, ao respeito, à liberdade e à convivência familiar e comunitária, além de colocá-los a salvo de toda forma de negligência, discriminação, exploração, violência, crueldade e opressão.

[4] § 4º - A lei punirá severamente o abuso, a violência e a exploração sexual da criança e do adolescente.

[5] Art. 2º Considera-se criança, para os efeitos desta Lei, a pessoa até doze anos de idade incompletos, e adolescente aquela entre doze e dezoito anos de idade.

[6] Estupro de vulnerável: Art. 217-A. Ter conjunção carnal ou praticar outro ato libidinoso com menor de 14 (catorze) anos: Pena - reclusão, de 8 (oito) a 15 (quinze) anos.

[7] Art. 218-B. Submeter, induzir ou atrair à prostituição ou outra forma de exploração sexual alguém menor de 18 (dezoito) anos ou que, por enfermidade ou deficiência mental, não tem o necessário discernimento para a prática do ato, facilitá-la, impedir ou dificultar que a abandone: Pena - reclusão, de 4 (quatro) a 10 (dez) anos. § 2º Incorre nas mesmas penas: I - quem pratica conjunção carnal ou outro ato libidinoso com alguém menor de 18 (dezoito) e maior de 14 (catorze) anos na situação descrita no caput deste artigo;

enquadram-se nessa lei os casos onde são encontrados arquivos contendo cenas de sexo explícito ou pornográfica envolvendo criança ou adolescente através das seguintes ações:

- Produzir, reproduzir e registrar (art. 240);
- Vender (art. 241);
- Disponibilizar, transmitir, distribuir, publicar ou divulgar (art. 241-A);
- Possuir ou armazenar registro (art. 241-B).

Ressalta-se que a forma culposa não é punida. Dessa forma, para que o processo penal possa terminar com a condenação do acusado, é necessária a comprovação do dolo, ou seja, a intenção do agente em realizar a conduta proibida pela lei.

2.4.2 Atuação das forças policiais no Brasil

As operações da Polícia Federal e das polícias estaduais, que têm como objetivo combater a exploração sexual de crianças e adolescentes, baseiam-se, principalmente, na identificação do endereço do IP.

Com o endereço IP, as forças policiais solicitam, ao poder judiciário, a quebra de sigilo telemático do suspeito. O direito ao sigilo telemático está previsto no inciso XII do artigo 5º da Constituição Federal [8] (Brasil, 1988). O ISP responsável será oficiado para que informe os dados cadastrais do terminal que utilizou o endereço IP, identificado no momento do crime.

A partir da obtenção dos dados cadastrais do terminal que utilizou o endereço IP, é solicitado ao poder judiciário a expedição de um mandado de busca e apreensão no endereço fornecido pelo ISP. Essa autorização é necessária, tendo em vista que o domicílio é asilo inviolável do indivíduo, conforme o inciso XI do artigo 5º da Constituição Federal [9] (Brasil, 1988).

Até o momento não existe legislação que obrigue os ISPs ao armazenamento dos dados referentes aos usuários que acessaram a Internet através de seus serviços. Também não existe uma padronização dos dados que devem ser disponibilizados às forças policiais nos casos em que a quebra de sigilo é autorizada. Essa ausência de legislação e

[8] XII - é inviolável o sigilo da correspondência e das comunicações telegráficas, de dados e das comunicações telefônicas, salvo, no último caso, por ordem judicial, nas hipóteses e na forma que a lei estabelecer para fins de investigação criminal ou instrução processual penal.

[9] XI - a casa é asilo inviolável do indivíduo, ninguém nela podendo penetrar sem consentimento do morador, salvo em caso de flagrante delito ou desastre, ou para prestar socorro, ou, durante o dia, por determinação judicial.

padronização diminui a confiabilidade das informações prestadas pelos ISPs, levando, em alguns casos, ao fornecimento de endereços incorretos do terminal utilizado pelo criminoso.

Caso deferido, o mandado de busca e apreensão é cumprido, geralmente, no início do dia, com a entrada no endereço (à força, se necessário) e na presença de testemunhas. Normalmente, todos os computadores e mídias de armazenamento são apreendidos e enviados para o setor pericial. É a partir da perícia que erros no fornecimento dos endereços utilizados por criminosos são detectados. Entretanto, os danos à sociedade já foram feitos – a entrada pela polícia no domicílio do suspeito, a observação por parte dos vizinhos, a presença de testemunhas, a retirada de computadores e mídias, dentre outros danos.

A execução de mandados de busca e apreensão em endereços incorretos também cria problemas para as forças policiais. Esses prejuízos advém do desperdício de recursos humanos e materiais, pois, por exemplo, o setor pericial deverá elaborar laudos que serão inconclusivos e inquéritos policiais serão instaurados com suspeitos errados, além da perda de prestígio das organizações policiais frente à sociedade.

Tabela 2.1: Principais operações da Polícia Federal contra o compartilhamento de material contendo pornografia infantojuvenil através de redes P2P até 2010

Operação	**Data**	**Nº de mandados**	**Estados envolvidos**
Anjo da Guarda I	07/07/2005	18	8
Anjo da Guarda II	31/08/2005	3	3
Azahar	22/02/2006	30	11
Carrossel I	20/12/2007	102	15
Arcanjo	06/06/2008	8	1
Carrossel II	03/09/2008	113	18
Turko	18/05/2009	92	21
Laio	15/09/2009	13	4
Ghost I	13/10/2009	1	1
Ghost II	05/11/2009	3	3
Tapete Persa	27/07/2010	81	10
Libras	06/12/2010	4	4
Comic Br	15/12/2010	19	7
	TOTAL	487	-

A Polícia Federal, além de atuações regionais, realizou diversas operações em âmbito nacional contra a disponibilização e distribuição de material relacionado à pornografia infantojuvenil. A Tabela 2.1 apresenta algumas dessas operações onde foram cumpridos 487 mandados de busca e apreensão em todos os estados do Brasil entre 2005 a 2010.

Tabela 2.2: Resultados obtidos pela execução do EspiaMule em março de 2008 conforme Fagundes (2009)

País	**Clientes (*user ID*)**	**Usuários de Internet**[1]	**Clientes / milhões de usuários de Internet**
China	34.756	162.000.000	214,54
Itália	25.437	31.481.928	807,99
França	18.959	32.925.953	575,81
Espanha	15.816	19.765.033	800,20
Alemanha	15.749	50.425.117	312,32
Brasil	13.725	39.140.000	350,66
Estados Unidos	8.210	210.575.287	38,99
Coréia do Sul	5.044	19.040.000	264,92
Polônia	4.111	14.084.600	291,88

[1] Fonte: `www.internetworldstats.com`

A Tabela 2.2 apresenta os resultados obtidos pela execução do aplicativo EspiaMule em março de 2008, quando foram realizadas buscas por mais de cem arquivos contendo cenas de exploração sexual de crianças e adolescentes na rede P2P do *eMule*, de acordo com Fagundes (2009). Esta tabela também ilustra a magnitude do problema, dado o número de computadores que compartilham esse tipo de arquivo.

Para Fagundes (2009), os resultados apresentados na Tabela 2.2 não indicam, necessariamente, que um determinado país tenha um maior número de pessoas compartilhando arquivos relacionados à pornografia infantojuvenil do que outros países, pois a popularidade da rede *eMule* varia conforme o país. A popularidade desse aplicativo é maior nos países asiáticos, América do Sul e países europeus (exceto Inglaterra), enquanto que nos Estados Unidos, Canadá, Inglaterra e outros países, a rede P2P mais popular é o *LimeWire*.

Atribuições da Polícia Federal e da Polícia Civil

A atribuição de atuação da Polícia Federal está prevista nos incisos I a IV do parágrafo 1º do artigo 144 da Constituição Federal (Brasil, 1988):

- *Inciso I - apurar infrações penais contra a ordem política e social ou em detrimento de bens, serviços e interesses da União ou de suas entidades autárquicas e empresas públicas, assim como outras infrações cuja prática tenha repercussão interestadual ou internacional e exija repressão uniforme, segundo se dispuser em lei;*

Os crimes contra a ordem tributária, econômica e contra as relações de consumo foram regulamentadas pela Lei nº 8.137, de 27 de dezembro de 1990. Em relação às infrações penais de repercussão interestadual ou internacional que exigem repressão uniforme, essa matéria foi regulada através da Lei nº 10.446, de 08 de maio de 2002.

- *Inciso II - prevenir e reprimir o tráfico ilícito de entorpecentes e drogas afins, o contrabando e o descaminho, sem prejuízo da ação fazendária e de outros órgãos públicos nas respectivas áreas de competência;*
  A regulamentação sobre o contrabando e descaminho encontra-se no Decreto nº 2.730, de 10 de agosto de 1998. Em relação ao tráfico ilícito de entorpecentes, a norma jurídica que regula a matéria é a Lei nº 11.343, de 23 de agosto de 2006.
- *Inciso III - exercer as funções de polícia marítima, aeroportuária e de fronteiras;*
- *Inciso IV - exercer, com exclusividade, as funções de polícia judiciária da União.*

Em relação à atribuição da polícia civil, de competência estadual, conforme o parágrafo 4º [10] do artigo 144 da Constituição Federal (Brasil, 1988), é residual, ou seja, o que não for de competência da União (competência das Polícias Federal, Ferroviária Federal e Rodoviária Federal ou crime militar) é de competência da polícia civil.

A competência das forças policiais não implica, necessariamente, a mesma competência do judiciário. Assim, é possível que um crime cuja atribuição de investigação seja da Polícia Federal, seja julgado pela Justiça Estadual e não pela Justiça Federal. Um exemplo seria o caso de um crime como o de furto e receptação de cargas, que é de atribuição das polícias estaduais, mas se tiver abrangência em vários estados, passa a ser de atribuição da Polícia Federal. O inquérito, ao ser relatado, deve ser encaminhado ao judiciário estadual para o devido trâmite processual.

2.5 Aspectos Legais

Caso seja cometido um crime que deixe vestígios, a perícia é obrigatória por expressa determinação do ordenamento jurídico brasileiro. Essa exigência legal está prevista no artigo 158 do Código de Processo Penal (CPP) [11] (Brasil, 1941).

[10] § 4º - às polícias civis, dirigidas por delegados de polícia de carreira, incumbem, ressalvada a competência da União, as funções de polícia judiciária e a apuração de infrações penais, exceto as militares.

[11] Art. 158. Quando a infração deixar vestígios, será indispensável o exame de corpo de delito, direto ou indireto, não podendo supri-lo a confissão do acusado.

O responsável pela realização das perícias criminais, de acordo com o artigo 159 do CPP [12] (Brasil, 1941), é o perito criminal oficial. Somente na ausência deste é que podem ser nomeados peritos pelo juiz (peritos não oficiais), conforme o parágrafo 1º desse mesmo artigo [13] (Brasil, 1941).

Deve-se ressaltar que os peritos exercem seu papel como auxiliares da justiça, conforme o artigo 139 do Código de Processo Civil (CPC) [14] (Brasil, 1973).

O principal objetivo dos exames periciais é a elaboração do Laudo pericial. Esse documento técnico tem por objetivo descrever minuciosamente o material examinado e responder quesitos formulados pela equipe de investigação, pelo Ministério Público, pelo acusado ou pelo juiz. Normalmente esses quesitos estão relacionados à materialidade e autoria de um delito. A aceitação dos exames periciais em um processo judicial é condicionada ao emprego de métodos científicos, rigorosamente testados, durante a perícia.

De acordo com o artigo 182 do CPP [15] (Brasil, 1941), o juiz não é obrigado a aceitar o laudo pericial, podendo rejeitá-lo parcialmente ou totalmente. Dessa forma, a confiabilidade das conclusões apresentadas no laudo pericial é um requisito fundamental para que tal documento seja aceito pelo judiciário como elemento de prova apto a produzir efeitos jurídicos.

Forças policiais estão em uma contínua corrida contra os criminosos na aplicação de tecnologias digitais. Para atuar de forma eficiente, essa disputa exige o desenvolvimento de ferramentas que tenham como objetivo buscar evidências digitais pertinentes. Além disso, e talvez seja a parte mais crucial, é a necessidade de que seja desenvolvida uma metodologia em perícias em informática que possa abranger a análise de todos os gêneros de investigações de cenas de crimes digitais (Reith et al., 2002).

Como a perícia em computadores (*computer forensics*) é um campo relativamente novo quando comparado com outras disciplinas forenses, existem esforços para o desenvolvimento de padrões e da estrutura de exames (Baryamureeba e Tushabe, 2004).

[12] Art. 159. O exame de corpo de delito e outras perícias serão realizados por perito oficial, portador de diploma de curso superior.

[13] § 1º Na falta de perito oficial, o exame será realizado por 2 (duas) pessoas idôneas, portadoras de diploma de curso superior preferencialmente na área específica, dentre as que tiverem habilitação técnica relacionada com a natureza do exame.

[14] Art. 139. São auxiliares do juízo, além de outros, cujas atribuições são determinadas pelas normas de organização judiciária, o escrivão, o oficial de justiça, o perito, o depositário, o administrador e o intérprete.

[15] Art. 182. O juiz não ficará adstrito ao laudo, podendo aceitá-lo ou rejeitá-lo, no todo ou em parte.

2.6 Persecução Penal no Brasil

Para que um Estado Democrático de Direito possa punir um cidadão pela prática de um crime, é necessário que se passe por um conjunto de etapas denominado persecução penal. No Brasil, a persecução penal engloba duas fases: a investigação criminal, também chamada de fase pré-processual, e o processo penal, também conhecido como fase processual.

A fase pré-processual, normalmente conduzida pela polícia, tem por objetivo reunir elementos para a comprovação do crime e de sua autoria. Com esses elementos, é possível a formação de juízo do Ministério Público (titular da ação penal pública) para oferecimento da denúncia ao judiciário. Se a denúncia do Ministério Público for aceita por um juiz, instaura-se o processo penal e, a partir deste momento, inaugura-se a fase processual.

A fase processual é o procedimento principal da persecução penal. Ao seu término, uma sentença judicial irá decidir se a pessoa acusada deverá ser condenada ou absolvida.

Na fase pré-processual o inquérito policial é o principal elemento. De acordo com Lopes Júnior (2006), pode-se conceituar o Inquérito Policial, basicamente, como sendo um procedimento administrativo de instrução provisória da autoridade policial, onde se pretende apurar prova de materialidade e indícios de autoria, como preparação do exercício da ação penal (base para oferecimento da denúncia).

O objetivo do inquérito policial é a obtenção de indícios de autoria e prova de materialidade do crime. Sem algum desses dois elementos, não é possível ao Ministério Público oferecer a denúncia ao judiciário.

Por indícios de autoria entende-se o conjunto de elementos que apontam que determinada pessoa foi o autor de um crime. Também é um elemento indispensável para o oferecimento da denúncia por parte do Ministério Público. Na fase pré-processual, não é necessária a obtenção de provas inequívocas de autoria, pois a lei determina apenas a identificação de "indícios", ou seja, elementos suficientes para fundar uma suspeita sobre alguém. Isso muda radicalmente na fase processual, onde são necessários elementos cabais de autoria para a eventual condenação judicial.

A prova de materialidade indica que o crime realmente ocorreu. Assim, por exemplo, o cadáver que apresenta sinais de morte violenta pode ser o resultado de homicídio, mas não indica necessariamente que o crime de homicídio ocorreu (pode ter sido suicídio).

O Brasil adota a Teoria Finalista da Ação para a definição de crime. Por esta teoria, para que um crime tenha ocorrido, são necessários dois elementos: (1) fato típico e (2) antijurídico.

Por fato típico, entende-se a conduta humana (ativa ou omissiva) que faz nascer um resultado previsto na legislação como crime. Se o resultado foi previsto pelo agente, ou seja, o agente tinha o objetivo de atingir tal resultado, diz-se que houve dolo. Caso o resultado não tenha sido previsto pelo agente, sendo um resultado involuntário, ocorreu culpa.

No Brasil, os crimes culposos (quando o agente não tinha por objetivo a obtenção do resultado) somente são punidos quando houver expressa previsão legal. Pode-se citar, como exemplo, o crime de homicídio, que prevê a imposição de pena nos casos em que o agente tinha a intenção de matar (dolo) e nos casos em que o agente, mesmo sem ter tido a intenção de fazê-lo (culpa), mata alguém. Em caso de condenação, a pena imposta irá variar de acordo com a intenção do agente.

A maioria dos delitos não prevê a punição na modalidade culposa. Um exemplo é a conduta de divulgação de material envolvendo a exploração sexual de crianças e adolescentes. Para que um indivíduo seja condenado por esse crime, é necessário que se comprove que o agente tinha intenção de divulgar esse material, pois é possível que tenha divulgado inadvertidamente e não existe a previsão legal para a modalidade culposa desse crime.

A conduta reprovável deve enquadrar-se perfeitamente à prevista no ordenamento jurídico, pois o princípio da legalidade, norteador do Direito Penal Brasileiro, previsto no inciso XXXIX do artigo 5° da Constituição Federal (Brasil, 1988), e também no artigo 1º do Código Penal Brasileiro, determina que: "Não há crime sem lei anterior que o defina. Não há pena sem prévia cominação legal".

Deve-se ressaltar que é possível o uso da analogia no direito penal, entretanto, apenas para o benefício do réu, ou seja, não é possível a criminalização de uma conduta ou a imposição de uma pena com o uso do instituto da analogia. Assim sendo, sem a existência de uma lei anterior que defina um crime como tal, o judiciário, bem como as forças policiais, encontram-se impedidos de atuar.

Quanto à antijuridicidade, toda conduta ilícita é um crime, a não ser que exista alguma situação, prevista no ordenamento jurídico, que exclua a ilicitude. As excludentes de ilicitude estão previstas no artigo 23 do CP [16] (Brasil, 1940).

[16] Art. 23 - Não há crime quando o agente pratica o fato: I - em estado de necessidade; II - em legítima defesa; III - em estrito cumprimento de dever legal ou no exercício regular de direito.

Capítulo 3

Redes Neurais Artificiais

Neste capítulo serão expostas informações acerca de RNAs. Na Seção 3.1 é feita uma breve introdução sobre IA e na Seção 3.2 são apresentados conceitos básicos sobre RNAs. Na Seção 3.3 são apresentadas informações sobre a classificação de estruturas de RNAs e, na Seção 3.4, o funcionamento do sistema de aprendizado dessas redes.

3.1 Inteligência Artificial

Diversos pesquisadores empreenderam esforços na compreensão do funcionamento da inteligência. A busca por respostas à questão de como "um punhado de matéria pode perceber, compreender, prever e manipular um mundo muito maior e mais complicado que ela própria" iniciou-se a milhares de anos (Russel e Norvig, 2004).

No entanto, uma das dificuldades para a compreesão da inteligência está na falta de conhecimento do funcionamento do cérebro humano. A anatomia do cérebro humano consiste em uma estrutura complexa e paralelizada de células nervosas denominadas neurônios (*neurons*). Cada neurônio pode se conectar a milhares de outros neurônios, se comunicando através de ligações chamadas de sinapses.

Embora a neurociência estude o cérebro humano a aproximadamente 150 anos, ainda existem diversas funções que não foram completamente entendidas pelos pesquisadores, os quais não conseguem responder como essa rede de células nervosas inteligadas pode resultar em inteligência.

O advento do computador possibilitou não apenas tentar compreender, mas também aplicar esse conhecimento na construção de dispositivos "inteligentes". Computadores são excelentes para realizar cálculos em alta velocidade, classificar e localizar informações e comunicar-se com outros computadores. Entretanto, em algumas áreas, ainda

não atingiram a mesma eficiência que os seres humanos, tal como reconhecimento de padrões, percepção e controle motor (Haykin, 1998).

Weiss (1999) afirma que os computadores não sabem bem o que devem fazer, pois precisam receber todas as ações que irão executar de forma explícita e antecipada por parte do programador. Caso o computador encontre uma situação que não foi prevista pelo projetista do sistema, geralmente o resultado é insatisfatório (como um *crash*). Isso não causa grandes aborrecimentos em pequenos sistemas de contabilidade, por exemplo, mas existe um grande conjunto de problemas nos quais é necessário que o computador decida por si mesmo, requerendo maior flexibilidade.

A busca pela criação de uma entidade "inteligente" e que possa resolver problemas levou ao surgimento da IA. Segundo Buchanan (2005), esse termo foi criado na *Conferência de Dartmouth*, a primeira conferência sobre o tema, realizada em 1956, no *Dartmouth College*, localizado na cidade de *Hanover*, em *New Hampshire*, nos Estados Unidos. Vários participantes dessa conferência, como John McCarthy, Marvin Minsky, Allen Newell e Herbert Simon foram os expoentes da pesquisa sobre IA nas décadas seguintes à da realização da conferência.

Lungarella et al. (2007) consideram a *Conferência de Dartmouth* como marco inicial do desenvolvimento da IA, pois foi após esse evento que foram criados diversos grupos de trabalho ao redor do mundo para a aplicação de IA tendo por objetivo a emulação e, até mesmo, a superação de alguns aspectos da capacidade humana com a utilização de computadores.

Diversos marcos foram alcançados desde então:

- a vitória do super computador *Deep Blue*, da IBM, sobre o campeão mundial de xadrez, Garry Kasparov, em maio de 1997;
- em 2005, um robô de *Stanford* ganhou o *Defense Advanced Research Projects Agency* (DARPA) *Grand Challenge* ao dirigir, de forma autônoma, por aproximadamente 200 quilômetros em uma estrada improvisada no deserto;
- em fevereiro de 2011, um equipamento da IBM, de nome *Watson*, ganhou de dois campeões (Brad Rutter e Ken Jennings) de *Jeopardy* (um show de perguntas e respostas (*quiz*) criado nos Estados Unidos), ao responder corretamente perguntas feitas em linguagem natural.

3.2 Conceitos Básicos de RNA

A IA tem se valido de várias estratégias para aplicar o funcionamento da inteligência em um sistema computacional. Uma delas é a aplicação do funcionamento biológico básico do cérebro, através da interconexão de vários elementos computacionais simples (Russel e Norvig, 2004). Essa técnica, denominada RNA, propõe um modelo onde são conectados diversas unidades denominadas neurônios artificiais. Cada neurônio estaria "ligado" ou "desligado", sendo a mudança de estado determinada pelo estímulo (representado pelo peso dado a cada entrada).

Ghiassi e Burnley (2010) apresentam os conceitos básicos de RNAs. Os autores afirmam que redes neurais, normalmente, empregam três ou mais camadas. Os dados entram na rede através da camada de entrada, passam por uma ou mais camadas ocultas e saem pela camada de saída. Os nós em cada camada estão ligados aos nós da camada posterior. Os nós das camadas ocultas e da camada de saída recebem as informações dos nós da camada anterior (de entrada ou outra camada oculta) e aplicam uma função de ativação. A função de ativação produz uma saída que é repassada para as camadas subsequentes. Existe um peso que é aplicado em cada conexão e os nós podem apresentar um viés (*bias*), valor somado ao valor ponderado pelo peso. Os valores dos pesos e do *bias* são criados através de um processo de treinamento, geralmente o de retropropagação de erros (*error backpropagation*).

De acordo com Kala et al. (2009), as redes neurais estão sendo aplicadas em diversas áreas, inclusive na área forense, como na identificação de locutor (Mohanty e Bhattacharya, 2008; Shukla e Tiwari, 2007), no reconhecimento facial (Taur e Tao, 2000; Shukla e Tiwari, 2007) e de escrita (Arnold e Miklos, 2010).

Os conceitos, citados nesta seção, serão discutidos de forma mais detalhada nas seções seguintes, iniciando por algumas características dos neurônios biológicos e artificiais nas Seções 3.2.1 e 3.2.2.

3.2.1 Neurônio Biológico

No cérebro humano, embora existam divergências sobre o valor exato, estima-se a existência de aproximadamente 100 bilhões (10^{11}) de neurônios conectados por meio de 100 trilhões (10^{14}) de sinapses.

Cada neurônio é composto, basicamente, de três partes: (1) o corpo celular, onde está localizado o núcleo da célula; (2) os dendritos, que fazem a conexão com outros

neurônio; e (3) o axônio, que é uma longa fibra e faz a transmissão dos impulsos nervosos do corpo celular até os dendritos.

A comunicação entre os neurônios ocorre através da transmissão elétrica entre as sinapses. As sinapses são unidades funcionais e estruturais fundamentais no processo de transmissão de informações. Cada conexão pode impor, em um determinado momento, excitação ou inibição, mas nunca ambas simultaneamente (Haykin, 1998).

Yegnanarayana (2004) apresenta algumas características que fazem com que a rede neural biológica seja superior ao mais avançado computador:

- Robustez e tolerância à falhas – o desempenho não parece ser afetado pela morte de células nervosas, pois as informações são distribuídas pela rede neural. Além disso, existe um grande número de neurônios, gerando redundância no armazenamento e processamento de informações.
- Plasticidade – ocorre um ajuste de forma automática como resposta a mudanças ambientais, sem a necessidade de instruções pré-programadas. Essa flexibilidade decorre da criação de novas sinapses e da modificação das sinapses existentes.
- Habilidade de lidar com grande variedade de situações – o cérebro humano consegue lidar com diversas situações, mesmo com dados inconsistentes. Cada neurônio processa as informações que possui em âmbito local, e repassa essas informações para os neurônios que estão ligados a ele. Dessa forma, não é necessária a existência de uma unidade central de controle.
- Alto paralelismo – as operações são desempenhadas de forma paralela, com uma grande quantidade de neurônios trabalhando simultaneamente em cada tarefa.

Essas vantagens compensam as desvantagens de tempo de processamento das redes neurais biológicas. Os computadores apresentam uma velocidade de processamento muito maior, sendo que as instruções são executadas na ordem de nanosegundos (10^{-9} segundos), enquanto que o tempo de execução das instruções no cérebro está na ordem de milissegundos (10^{-3} segundos).

3.2.2 Neurônio Artificial

É possível replicar algumas das características do sistema neural biológico na construção de RNAs com o emprego de neurônios artificiais. Pode-se demonstrar que tais redes apresentam processamento distribuído e paralelismo, além de armazenar informações

de forma distribuída nos pesos de conexão, com o objetivo de melhorar a tolerância a falhas (Yegnanarayana, 2004).

Atribui-se a criação de um modelo que descreve um neurônio artificial ao trabalho de McCulloch e Pitts (1943). Neste trabalho, Warren McCulloch (neurofisiologista e psiquiatra) e Walter Pitts (lógico) apresentaram uma simulação do funcionamento de células nervosas vivas em um neurônio artificial, por meio de um modelo de resistores variáveis e amplificadores. Uma representação do funcionamento de um neurônio artificial individual é apresentado na Figura 3.1.

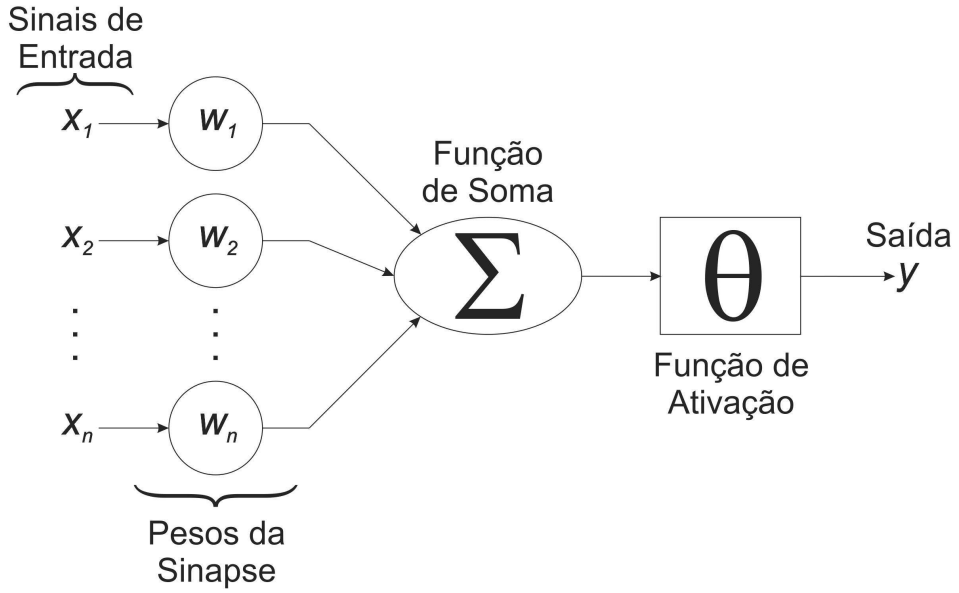


Figura 3.1: Modelo matemático individual de um neurônio artificial (adaptado de McCulloch e Pitts, 1943)

Para obter a saída, o modelo de McCulloch e Pitts (1943) utilizava uma função de ativação de limiar (θ). Essa função, apresentada graficamente na Figura 3.4 (a), tem como função restringir a amplitude da saída de um neurônio (Haykin, 1998) e pode ser expressa matematicamente conforme apresentado na Equação 3.1:

$$g(in_i) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \tag{3.1}$$

Um dos resultados obtidos pelo trabalho de McCulloch e Pitts (1943) foi a criação de representações de funções booleanas básicas. A Figura 3.2 demonstra o emprego de um neurônio artificial de modo a atuar como portas lógicas, conforme apresentado por Russel e Norvig (2004). O limiar é representado por W_0.

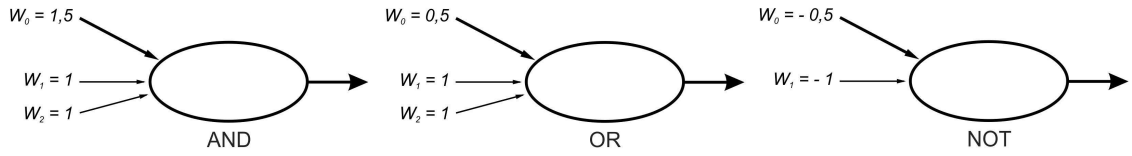


Figura 3.2: Portas lógicas E (*AND*), OU (*OR*) e NÃO (*NOT*) com a utilização de neurônios artificiais (adaptado de Russel e Norvig, 2004)

No trabalho de Warren McCulloch e Walter Pitts, os pesos sinápticos dos neurônios artificiais eram fixos. Posteriormente, foram desenvolvidas outras abordagens que previam a flexibilidade dos pesos sinápticos, ampliando as possibilidades de utilização da RNA. Uma dessas abordagens é o modelo apresentado por Russel e Norvig (2004), exibido na Figura 3.3.

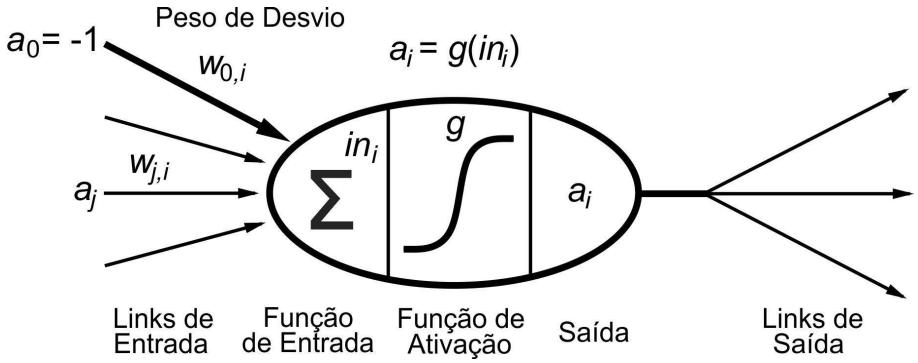


Figura 3.3: Representação do modelo de um neurônio artificial ligado em rede proposto por Russel e Norvig (2004)

Em 1949, Hebb propôs a utilização de um método de treinamento para o ajuste dos pesos das conexões (modificações sinápticas). Hebb faz uma analogia com o cérebro, onde sua conectividade é modificada de forma contínua, de acordo com o aprendizado de tarefas funcionais. Em seu *postulado de aprendizagem*, afirma que uma sinapse é aumentada pela repetida ativação de um neurônio por outro neurônio.

Quinze anos após o trabalho de McCulloch e Pitts (1943), Rosenblatt (1958) propôs a utilização de aprendizagem supervisionada como modelo de aprendizagem.

O trabalho de Minsky e Papert (1969) demonstrou, de forma matemática, que existem limites do que a rede de camada única consegue calcular. Os autores também afirmaram que não existem elementos para acreditar que redes de múltiplas camadas consigam superar esses limites.

Segundo Haykin (1998), depois do trabalho de Minsky e Papert (1969), houve um hiato de pesquisa na área, diminuindo o número de pesquisadores que continuaram trabalhando na área de redes neurais.

Nos trabalhos de Rumelhart et al. (1985) e Rumelhart e McClelland (1986) foi relatado o emprego do algoritmo de retropropagação (*backpropagation*) para aprendizagem por máquina. Esse algoritmo, que tornou-se o mais utilizado para o treinamento de redes neurais de múltiplas camadas, despertou novamente o interesse por essa área. Diversas pesquisas, desde então, demonstraram as diversas possibilidades de utilização de redes neurais.

Nas redes neurais, os links de entrada correspondem aos estímulos criados por outros neurônios, e cada entrada possui um peso associado, denominado peso sináptico e que vai determinar a intensidade e o sinal da conexão.

Caso o estímulo, gerado por neurônios vizinhos (sinapses), chegasse a um patamar pré-determinado, o neurônio artificial mudaria seu estado para "ligado". A força de conexão entre cada neurônio, chamado de peso sináptico, é utilizada para armazenar o conhecimento gerado. Inicialmente, cada unidade i efetua o somatório, das suas entradas ponderadas pelos pesos, conforme apresentado na Equação 3.2:

$$in_i = \sum_{j=0}^{n} W_{j,i} a_j \tag{3.2}$$

Em seguida, é aplicada uma função de ativação g sobre esse valor como na Equação 3.3:

$$a_i = g(in_i) = g\left(\sum_{j=0}^{n} W_{j,i} a_j\right) \tag{3.3}$$

A função de ativação, de acordo com Russel e Norvig (2004), atende a dois propósitos: (1) a unidade deve estar ativa (próxima de +1) quando as entradas "corretas" forem recebidas, e que esteja inativa (próxima de 0) quando as entradas forem "incorretas", (2) além disso, a ativação necessita ser não-linear para que a RNA inteira não entre em colapso. Na Figura 3.4, são apresentados, como exemplos, três escolhas para a função de ativação g: a *função de limiar*, a *função linear por partes* e a *função sigmóide* ($\frac{1}{1+e^{-x}}$).

De acordo com Haykin (1998), a função de limiar é normalmente, na área de engenharia, conhecida como função de *Heaviside*. A função sigmóide, com gráfico em forma de "S", é a forma mais comum de função de ativação, tendo uma mistura entre comportamento linear e não-linear. Um exemplo de função sigmóide é a função logística, que pode ser expressa conforme a Equação 3.4, onde a é o parâmetro de inclinação da função sigmóide.:

$$g = \frac{1}{1 + e^{(-ax)}} \tag{3.4}$$

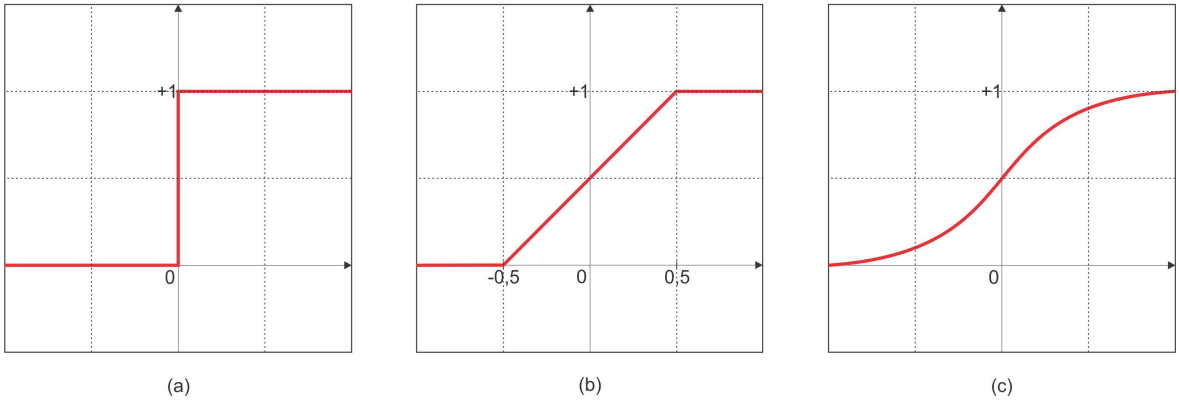


Figura 3.4: Três funções de ativação: (a) função de ativação de limiar, onde o valor de saída é 0 se a entrada for negativa e 1 caso a entrada seja positiva (b) função linear por partes e (c) função de ativação sigmóide (adaptado de Haykin, 1998)

Em função da aplicação utilizada, as funções de ativação podem variar de 0 a +1 ou de −1 a +1.

3.3 Estruturas de Rede

De acordo com Haykin (1998), a maneira em que estão organizados os neurônios em uma RNA está intimamente ligada ao algoritmo de aprendizagem utilizado para o treinamento da rede.

Russel e Norvig (2004) classificam as estruturas de redes neurais em duas categorias: (1) redes de alimentação direta (*feedforward*) e (2) redes cíclicas ou recorrentes (*recurrent*). As redes de alimentação direta não tem estado interno além dos pesos associados às entradas. Dois exemplos dessas redes são apresentados nas Figuras 3.6 e 3.8. Já as redes recorrentes utilizam suas saídas para alimentar novamente as entradas, apresentando um efeito de memória de curto prazo, e, eventualmente, um comportamento instável e até mesmo caótico. Um exemplo dessa rede é apresentado na Figura 3.5. Por este motivo, embora possuam aplicações específicas, as redes recorrentes são difíceis de compreender. Dessa forma, serão apresentados apenas dados relacionados a redes de alimentação direta deste ponto em diante.

As redes de alimentação direta representam uma função de suas entradas, e os pesos na rede atuam como parâmetros dessa função. Com a modificação dos pesos (parâmetros), muda-se a função que a rede representa, e é esse o processo utilizado para o aprendizado. Na Seção 3.3.1 serão apresentadas informações sobre a rede de única

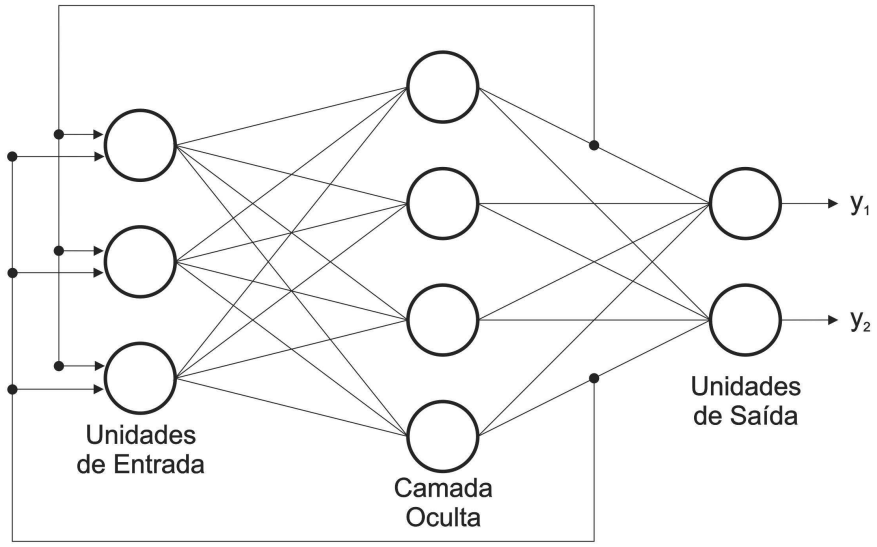


Figura 3.5: Estrutura de RNA de alimentação recorrente

camada e na Seção 3.3.2 serão expostos dados sobre as redes de alimentação direta de várias camadas.

3.3.1 Rede neural de alimentação direta de única camada

A rede neural de alimentação direta de única camada (rede de perceptron) é a forma mais simples de redes neurais em camadas. Nas redes de camada única, cada entrada se comunica diretamente com as saídas, inexistindo camadas intermediárias. Um exemplo desse tipo de estrutura de rede, com quatro neurônios de entrada e dois neurônios de saída, é apresentado na Figura 3.6.

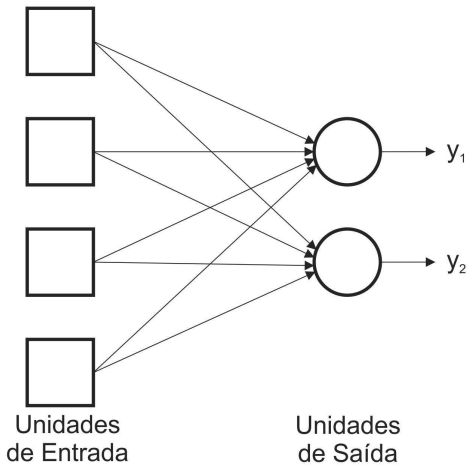


Figura 3.6: Estrutura de RNA de alimentação direta de uma única camada (adaptada de Russel e Norvig, 2004)

Com a utilização de uma função de ativação de limiar, o perceptron consegue representar, além de funções booleanas elementares, como o E, OU e NÃO (Figura 3.2), algumas funções mais complexas. Entretanto, muitas funções não podem ser representadas com o perceptron de limiar. Nessas funções, não é possível uma separação entre os grupos com apenas uma reta, ou seja, o perceptron somente consegue representar uma função linearmente separável (Russel e Norvig, 2004).

Na Figura 3.7 são apresentados três exemplos para a utilização de redes perceptron. Nos dois primeiros casos (função AND e OR) é possível o emprego desse tipo de rede, pois são funções linearmente separáveis. Já no terceiro exemplo (XOR), por não ser possível separar os dois grupos com apenas uma reta, não existe maneira de que um perceptron de limiar possa aprender essa função.

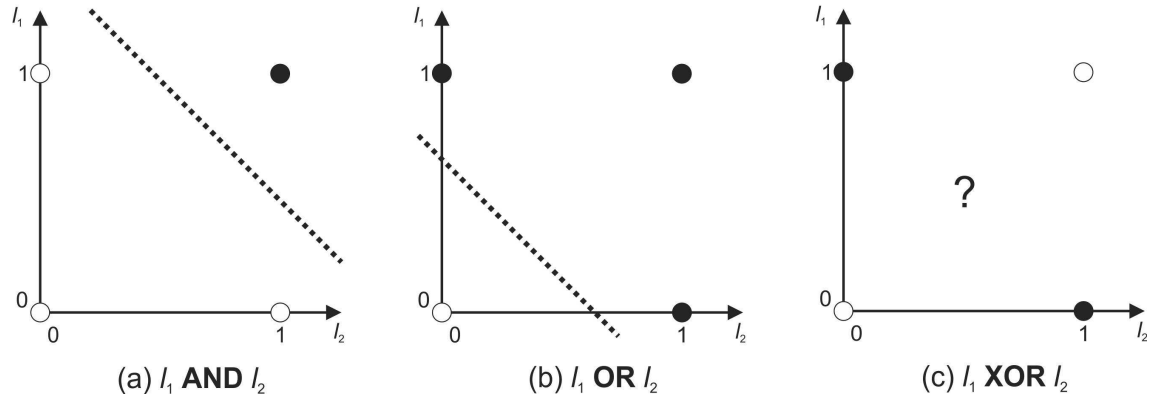


Figura 3.7: Separação linear em perceptron de limiar, conforme apresentado por Russel e Norvig (2004)

De acordo com Russel e Norvig (2004), apesar de limitações no aprendizado de algumas funções, os perceptrons de limiar podem aprender qualquer conjunto de treinamento linearmente separável com o emprego de um algoritmo simples.

3.3.2 Rede neural de alimentação direta de várias camadas

Essa estrutura de rede se distingue por possuir uma ou mais camadas ocultas (além das camadas de entrada e saída), cujas unidades são chamadas de unidades ocultas ou neurônios ocultos. Essa camada oculta possui a habilidade de extrair estatísticas de ordem elevada, sendo útil quando a rede possui muitas entradas (Haykin, 1998). O caso mais simples envolve apenas uma camada oculta, conforme apresentado na Figura 3.8.

Com a adição de camadas ocultas, ocorre um aumento do espaço e hipóteses que a rede neural pode representar. Esse aumento possibilita, por exemplo, a obtenção de uma

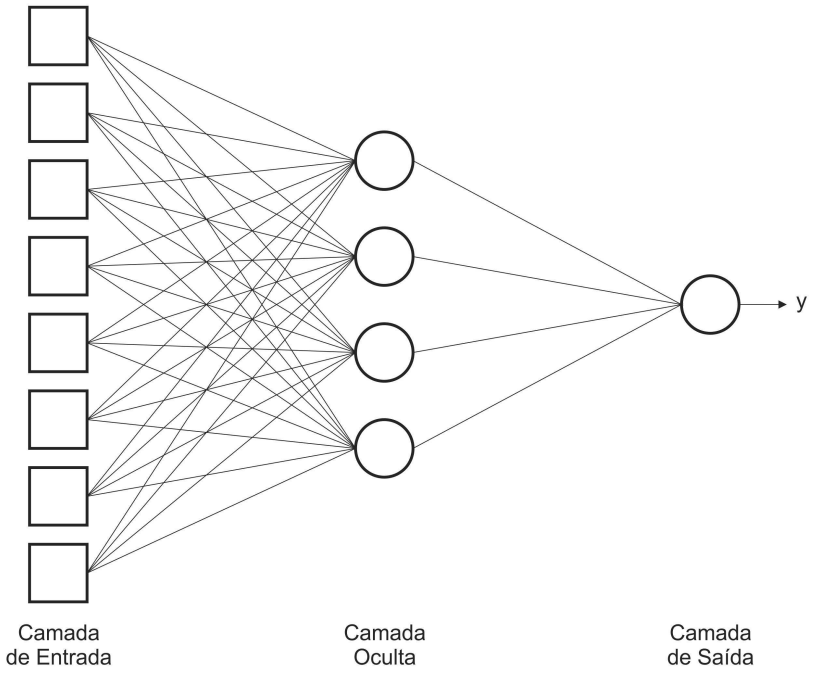


Figura 3.8: Rede neural de alimentação direta de várias camadas, com oito entradas, uma camada oculta e uma saída (adaptada de Haykin, 1998)

função de cume, através da combinação de duas funções de limiar opostas e limitando o resultado. Com a adição de mais unidades ocultas, é possível criar mais funções de cume (Russel e Norvig, 2004). Esse exemplo é apresentado na Figura 3.9.

Quanto cada camada está conectada a todos os elementos da camada adjacente, essa rede é chamada de totalmente conectada. Caso falte alguma conexão, diz-se que a rede é parcialmente conectada (Haykin, 1998).

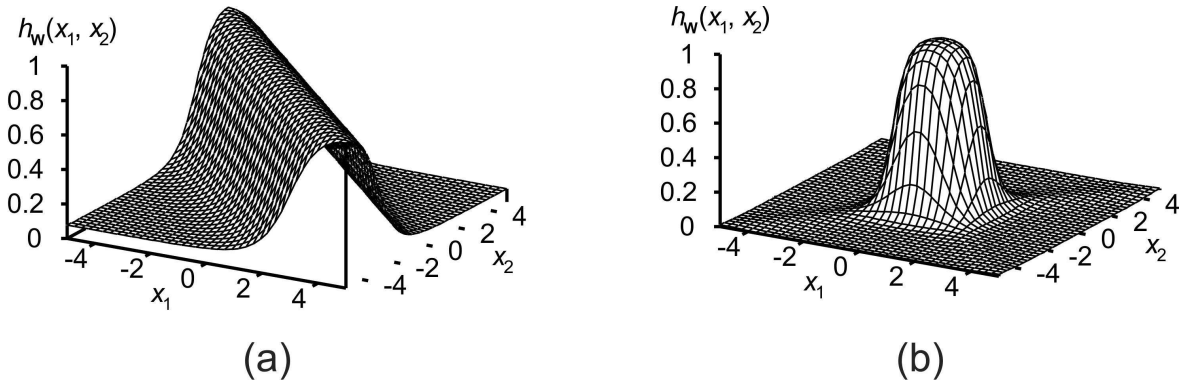


Figura 3.9: (a) Combinação de duas funções de limiar para produzir um cume e (b) combinação de dois cumes para produzir uma colina, conforme Russel e Norvig (2004)

3.4 Aprendizado

Para que uma rede neural possa aplicar o conhecimento, inicialmente ela deve obter as informações necessárias e armazenar esses dados à medida que for aprendendo.

A classificação é um processo de atribuição de um objeto a um conjunto de classes, baseado nos atributos desse objeto. Ghiassi e Burnley (2010) afirmam que o processo de classificação deve definir, inicialmente, um conjunto de treinamento para o aprendizado da rede neural. Em seguida, deve ser aplicado em um conjunto de testes para a identificação da taxa de acertos desse processo de classificação.

Russel e Norvig (2004) apresentam três formas de aprendizado:

1. aprendizagem supervisionada – são apresentados exemplos e a respectiva saída esperada;
2. aprendizagem não-supervisionada – envolve a aprendizagem de padrões na entrada, quando não são mencionadas as saídas previstas;
3. aprendizagem por reforço – é a categoria mais geral, na qual o agente deve aprender a partir do reforço (também chamado de recompensa), onde pode receber alguma indicação de que seu comportamento é ou não desejável

A aprendizagem supervisionada é uma forma popular de aprendizado. Os pesos das conexões são modificados através de um conjunto de amostras de treinamento rotuladas, também chamadas de exemplos da tarefa. Cada exemplo consiste na entrada e a respectiva saída correta. Caso a saída coincida com a saída desejada, não são modificados os pesos. Caso contrário, os pesos são alterados, minimizando a diferença entre a resposta efetivamente obtida e a resposta considerada correta, de acordo com um critério estatístico pré-determinado. O treinamento continua até que não hajam modificações significativas dos pesos ou até que se chegue a um limite de acerto pré-estabelecido.

De acordo com Russel e Norvig (2004), uma forma "clássica" de busca na otimização nos pesos é através da soma dos erros quadráticos (*sum of squared errors*). Um exemplo de aplicação da soma dos erros quadráticos para um único elemento de treinamento, sendo x a entrada, y a saída desejada e $h_w(x)$ a saída do perceptron no exemplo, pode ser escrito conforme a Equação 3.5:

$$E = \frac{1}{2}Err^2 \equiv \frac{1}{2}(y - h_w(x))^2 \tag{3.5}$$

Dessa forma, os pesos são ajustados um a um, após cada exemplo, para reduzir o erro. Cada ciclo em que é realizado o treinamento com o emprego de todos exemplos é denominado *época*.

De acordo com Russel e Norvig (2004), embora os erros nas camadas de saída de redes de única camada sejam claros, nas redes neurais que apresentam camadas ocultas o erro é de difícil identificação, pois os dados de treinamento não informam o valor correto para a saída nas camadas ocultas. Entretanto, é possível fazer a propagação de retorno ou retropropagação (*back-propagate*) do erro da camada de saída para as camadas intermediárias. Conforme esses autores, como podem existir várias saídas, e Err_i é o i-ésimo componente do vetor de erro $y - h_w$. Também é apropriada a criação de um erro modificado $\Delta_i = Err_i \times g'(in_i)$ para que a regra de atualização de pesos fique da forma da Equação 3.6, onde α é a taxa de aprendizagem:

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i \tag{3.6}$$

Russel e Norvig (2004) também apresentam uma forma resumida do processo de propagação de retorno (*backpropagation*):

- Calcular o Δ para o(s) elemento(s) de saída, empregando o erro encontrado.
- Iniciando na camada de saída, propagar o Δ para a camada anterior e modificar o peso entre as camadas, até ser alcançada a camada oculta ligada à camada de entrada.

O modelo proposto da RNA que empregará conceitos apresentados neste capítulo será apresentada no Capítulo 5.

Capítulo 4

Redes *Peer-to-Peer*

Neste capítulo serão apresentadas informações a respeito de redes P2P, como a evolução história do paradigma P2P (Seção 4.1), seu funcionamento básico e algumas formas de classificação desse tipo de aplicativo (Seção 4.2). Também serão apontadas, na Seção 4.3, formas de identificação de tráfego P2P e na Seção 4.4 informações sobre o *eMule*.

4.1 Histórico

O conceito por trás das redes P2P não é novo. De fato, a Internet, quando criada, possuía funcionamento semelhante ao empregado pelos aplicativos P2P atuais. A criação da Internet começou durante a guerra fria entre os Estados Unidos e a União das Repúblicas Socialistas Soviéticas (URSS). O lançamento do foguete *Sputnik* pelos soviéticos, em 1957, teve como consequência a fundação da *Advanced Research Projects Agency* (ARPA), pelo governo norte-americano, no ano seguinte. Essa agência foi criada para ampliar a capacidade de defesa dos Estados Unidos, melhorando a tecnologia empregada pelo país contra avanços de adversários (DARPA, 2011).

A ARPA precisava de uma forma de conexão segura e eficiente entre as bases militares norte-americanas, através de diferentes redes de computadores. Na década de 60, foi iniciado o desenvolvimento da *Advanced Research Projects Agency Network* (ARPANet). Era necessário garantir a continuidade da transmissão de informações, mesmo na ocorrência de ataques nucleares que destruíssem as principais rotas de comunicação, possibilitando uma coordenação de atividades militares (Leiner et al., 2009). Essa agência teve seu nome modificado para DARPA em março de 1996.

Segundo Oram (2001), o modelo utilizado pela ARPANet não era o cliente/servidor, mas sim o de igualdade entre os computadores, mais próximo do modelo P2P.

Na década de 70, novas aplicações surgiram, como o Telnet e o *File Transfer Protocol* (FTP), que expandiram a capacidade de comunicação. Embora essas aplicações funcionassem com base no modelo cliente/servidor, cada computador poderia agir em um momento como cliente e logo após como servidor.

Nessa etapa do desenvolvimento tecnológico, foram definidos padrões que levaram à criação do *hardware* e *software* necessários à comunicação entre os computadores, e que se mantém até hoje, com algumas alterações (Yar, 2006). Esses padrões foram documentados em uma série numerada de normas denominada *Request for Comments* (RFC). Como exemplo, pode-se citar o RFC 793, que define o TCP, o RFC 2616, relacionado ao *Hypertext Transfer Protocol* (HTTP), o RFC 2821, relativo ao *Simple Mail Transfer Protocol* (SMTP) e o RFC 959, referente ao FTP. Frequentemente são criados novos padrões, tendo sido discutidos e documentados através de uma RFC.

Surgiram outras redes na década de 70 e 80, paralelas à ARPANet, como a NSFNET (da *American National Science Foundation*) e JANET (*Joint Academic Network*), do Reino Unido. A gestão de autoridades militares norte-americanas perdurou até 1990, quando então o controle da ARPANet foi repassado para a área civil, sob a coordenação da *National Science Foundation* (NSF).

Com a evolução tecnológica, houve maior especialização dos serviços prestados pelos servidores. Essa mudança ocorreu em virtude da necessidade de maior robustez, segurança, disponibilidade e melhores enlaces de comunicação. Os computadores foram estruturados de forma que algumas poucas máquinas de grande capacidade (servidores) fornecessem serviços a um grande número de outras máquinas (clientes) – o chamado modelo cliente/servidor.

Diversas aplicações utilizadas na Internet atualmente são baseadas no modelo cliente/servidor como, por exemplo, o correio eletrônico e o *World Wide Web* (WWW). Neste modelo, cada um dos elementos que se comunicam exerce apenas um papel: ou como cliente ou como servidor. O computador servidor fornece o serviço e o computador cliente utiliza esse serviço.

O problema do modelo cliente/servidor é que os recursos computacionais disponíveis nos clientes são desperdiçados. Na década de 80, os computadores clientes possuíam razoável poder computacional e houve um crescimento de estudos sobre uma forma de melhorar o aproveitamento dos recursos eventualmente ociosos dos clientes.

Diversos aplicativos foram desenvolvidos como frutos desses estudos, culminando, em junho de 1999, com o Napster. Esse aplicativo possibilitou uma grande popularização de redes P2P. No ápice de sua utilização, o Napster possuía mais de 50 milhões de usuários e funcionava em uma estrutura P2P que utilizava servidores para gerenciar os nós conectados e os arquivos compartilhados. O Napster foi fechado, em julho de 2001, por meio de uma sentença judicial por infração de direitos autorais.

A arquitetura de rede P2P difere radicalmente da arquitetura de rede cliente/servidor, na qual os clientes acessam os recursos disponibilizados pelos servidores. No P2P, o acesso é feito entre nós (ou pontos), que atuam, simultaneamente, como servidores e como clientes, inexistindo uma hierarquia, uma divisão fixa entre os papéis a serem desempenhados por esses computadores.

O interesse em redes P2P está relacionado com a quantidade de bytes transportados por esse tipo de aplicação. Segundo o estudo da empresa alemã Ipoque (2009), realizado em 2008/2009, 45% a 70% do tráfego da Internet era referente a aplicativos P2P, suplantando o tráfego HTTP, FTP e SMTP. Esse grande volume de tráfego deve-se à característica dos aplicativos P2P que proporcionam uma excelente base para a criação de uma grande rede de compartilhamento de dados (Crowcroft et al., 2004).

A aplicação predominante na qual é utilizada a arquitetura P2P (mas não a única), é o compartilhamento de arquivos pela Internet, possibilitando aos usuários a busca, obtenção e a transmissão de dados. Em algumas aplicações, o usuário realiza consultas em servidores centrais que armazenam a relação de arquivos que estão sendo disponibilizados pelos usuários conectados a ele. Já em outras aplicações, é possível a consulta direta aos arquivos que são disponibilizados por determinado usuário.

Recentemente, diversos aplicativos P2P de troca de arquivos receberam atenção por terem alcançado grande popularidade. Dentre esses aplicativos, pode-se citar o *Limewire* [1], o *KaZaa* [2], o *eMule* [3], o *BitTorrent* [4] e o *Vuze* [5].

A comunicação entre os nós de um sistema P2P ocorre através da troca de mensagens que são definidas por um protocolo e transmitidas pela Internet. Segundo Buford et al. (2009), as principais características desse tipo de aplicativo são: (1) os protocolos são construídos na camada de aplicação do modelo OSI; (2) na maioria das implementações, cada nó possui um identificador (ID); (3) muitas mensagens, definidas em protocolos

[1] `http://www.limewire.com`
[2] `http://www.kazaa.com`
[3] `http://www.emule-project.net`
[4] `http://www.bittorrent.com`
[5] `http://www.vuze.com`

P2P diferentes, são similares; e (4) o protocolo suporta algum tipo de roteamento de mensagens.

Para seu funcionamento, os aplicativos P2P criam uma rede virtual (*overlay*) entre os recursos distribuídos disponibilizados pelos nós pertencentes à rede P2P. Após um nó ter localizado o recurso que deseja obter, a comunicação é feita diretamente entre os nós ou através de nós intermediários. Segundo Rocha et al. (2004), a Internet é um exemplo de rede *overlay*, pois utiliza o IP como solução de conexão entre redes com tecnologias diversas, tais como ATM, *Frame Relay*, PSTN entre outras.

4.2 Classificações de Redes P2P

As redes P2P podem ser classificadas de diversas formas. Neste ponto, serão apresentadas três classificações: a primeira, que utiliza como critério a funcionalidade do aplicativo P2P (Seção 4.2.1), a segunda, o grau de centralização (Seção 4.2.2), e a terceira, a estrutura de rede (Seção 4.2.3).

4.2.1 Classificação com base na funcionalidade

O sucesso do Napster, no início da década de 90, chamou a atenção para o paradigma P2P, que passou a ser aplicado em diversas áreas. Atualmente, esses aplicativos podem ser agrupados, tendo como foco sua funcionalidade, da seguinte forma:

- Distribuição de conteúdo - essa é a aplicação mais conhecida das redes P2P. Além dos aplicativos de compartilhamento de arquivos como o *eMule*, *Shareaza*, *Gnutella*, *Kazaa* e *Limewire*, também podem ser incluídos os de distribuição de fluxo de mídia (*streaming*), como o *TVUPlayer*, *Goalbit*, *Abroadcasting*, *Zattoo*, *PPLive*, *Octoshape*, *Joost*, *CoolStreaming*, *Cybersky-TV*, *LiveStation* e *Didiom*.

- Computação distribuída - as funcionalidades de compartilhamento de recursos, inerentes às redes P2P, são aplicadas para a distribuição de carga de processamento. Como exemplos, temos o *SETI@home* (busca por emissões de rádio extraterrestres), *Docking@Home* (modelagem de proteínas), *Climateprediction.net* (previsão meteorológica), *MilkyWay@Home* (tenta deduzir a estrutura da Via Láctea) e *distributed.net* (tenta quebrar a cifra criptográfica RC5).

- Comunicação - a utilização do modelo P2P para possibilitar a comunicação entre terminais é frequente. Como exemplos, pode-se citar o *Skype*, *ICQ*, *Yahoo! Messenger*, *Microsoft Messenger* (MSN) e *Tencent QQ*.

4.2.2 Classificação com base no grau de centralização

A classificação com base no grau de centralização, conforme Androutsellis-Theotokis e Spinellis (2004) é exposta na Figura 4.1, a qual apresenta o modelo cliente/servidor (a) e classificações de arquiteturas P2P com relação ao grau de centralização (b, c e d).

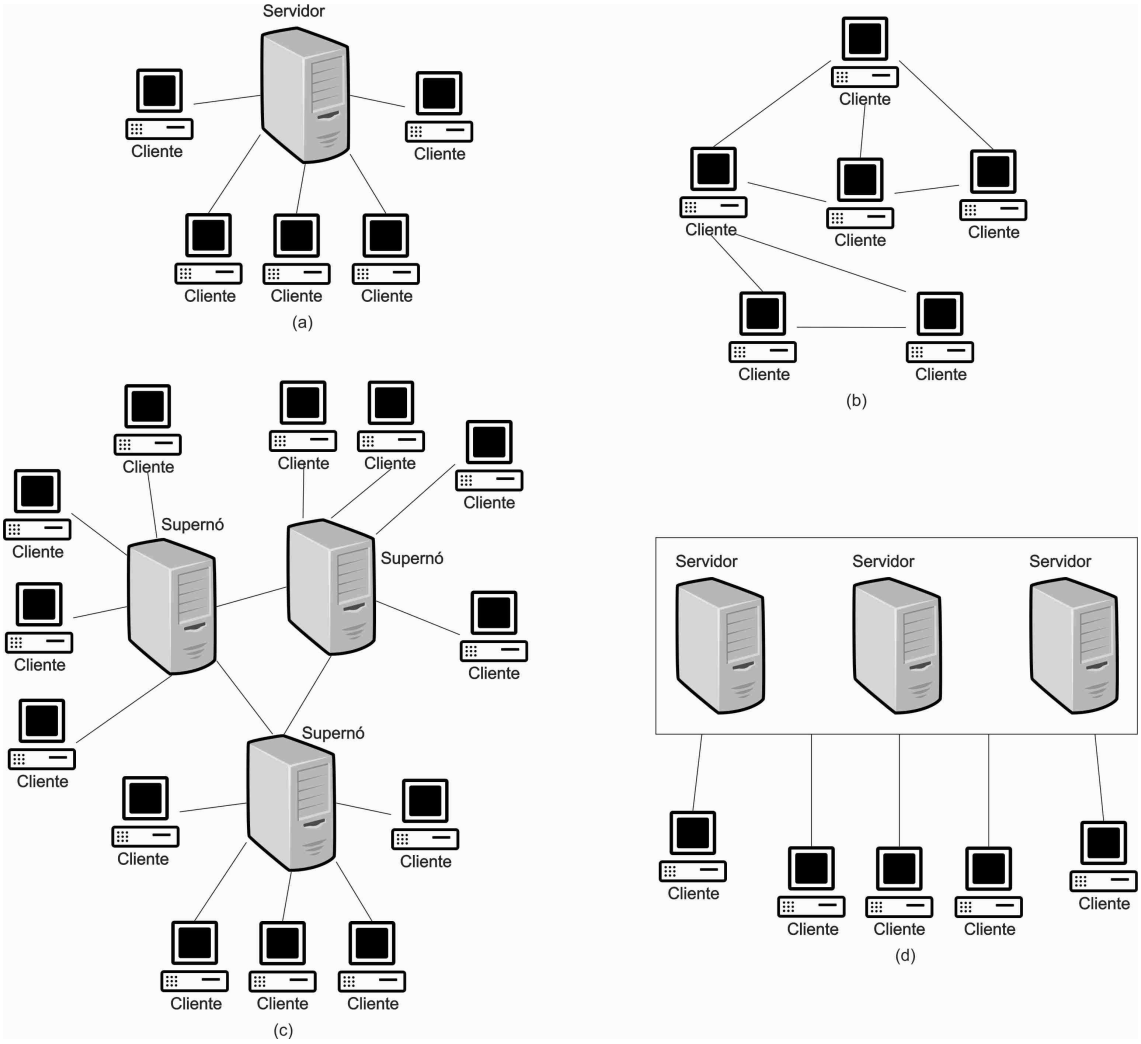


Figura 4.1: Classificação de redes P2P pelo grau de centralização da rede segundo Androutsellis-Theotokis e Spinellis (2004): (a) Modelo Cliente/Servidor (b) Puramente descentralizado (c) Parcialmente centralizado e (d) Híbrida descentralizada

Como principais características dos diferentes modelos arquiteturais P2P, Androutsellis-Theotokis e Spinellis (2004) citam:

- Arquitetura Puramente Descentralizada (*Purely Decentralized Architectures*) (b) – todos os nós da rede possuem a mesma hierarquia, funcionando simultaneamente como cliente e servidor, inexistindo uma coordenação central dessas atividades. Os nós desse tipo de rede são chamados frequentemente de "*SERVENTS*"

(*SERVer + cliENTS*). A rede KAD, acessada pelo *eMule*, é um exemplo desse tipo de arquitetura.

- Arquitetura Parcialmente Centralizada (*Partially Centralized Architectures*) (c) – a base é a mesma encontrada na arquitetura puramente descentralizada, mas nesta arquitetura alguns nós assumem um papel mais importante, agindo como índices de arquivos disponibilizados por nós locais. Esses nós são chamados de supernós (*supernodes*). A forma de escolher se um nó será normal ou um supernó varia de rede para rede. É importante ressaltar que para que a rede possa ser tolerante a falhas, quando um desses nós deixa de estar acessível, a rede deve substituí-lo de forma automática. Um exemplo é a rede *Gnutella*.

- Arquitetura Híbrida Descentralizada (*Hybrid Decentralized Architectures*) (d) – nesses sistemas, existe um servidor central que facilita a interação entre os nós armazenando metadados dos arquivos que são compartilhados. Esse servidor central responde quais nós disponibilizam os arquivos buscados. Nesta arquitetura, os servidores são pontos únicos de falha, podendo paralisar a rede quando inativos, tornando a rede vulnerável a ataques, censura ou falha técnica. O servidor armazena uma tabela com as informações dos usuários que estão conectados a este servidor e a relação de arquivos que são compartilhados por cada usuário. Quando recebe uma solicitação de busca por arquivos, o servidor responde com a relação de usuários que estão disponibilizando esse arquivo. Suas vantagens são a simplicidade de implementação e a velocidade e eficiência para a localização de arquivos. Um exemplo dessa arquitetura é a rede eD2k, utilizada pelo *eMule*.

4.2.3 Classificação com base na estrutura de rede

A classificação de sistemas P2P, tendo como critério a estrutura de rede, leva em conta a forma de criação da rede *overlay*, se são criadas de forma não-determinística (também chamada *ad hoc*) ou se seguem alguma regra específica para a entrada de nós ou de conteúdo na rede. Conforme Androutsellis-Theotokis e Spinellis (2004), as redes P2P, são classificadas, de acordo com a estrutura, em:

- Redes Desestruturadas (*Unstructured Networks*) - nestas redes, a localização dos arquivos (conteúdo) não está relacionada com a estrutura da rede *overlay*. Em uma rede desestruturada, os mecanismos de busca de recursos variam do método de força bruta, como a inundação (*flooding*) da rede até que o arquivo seja localizado, ou utilizando estratégias que otimizam recursos, tal como *random walk*

ou tabelas de roteamento. O mecanismo de busca utilizado afeta fortemente a rede, principalmente em relação à disponibilidade e escalabilidade do sistema. Esse tipo de rede é mais apropriado para sistemas onde a frequência da entrada e saída de nós é bastante alta.

- Redes Estruturadas (*Structured Networks*) - essa arquitetura foi criada como uma forma de solucionar os problemas de escalabilidade encontradas nas redes desestruturadas. Em redes estruturadas, os arquivos (ou ponteiros para os mesmos) são colocados em posições específicas, existindo um mapeamento entre o conteúdo (por exemplo, um arquivo) e a sua localização (por exemplo, a identificação de um nó), para que buscas possam ser eficientemente roteadas até o nó que possui o conteúdo desejado. Um exemplo é a rede eD2k, acessada pelo *eMule*.

- Redes Fracamente Estruturadas (*Loosely Structured Networks*) - é uma solução intermediária, que aproxima as redes estruturadas das desestruturadas. Nessas redes, a localização exata de um conteúdo não é completamente especificada, apontando apenas para uma localização aproximada. Assim, em alguns casos, mesmo que o arquivo esteja sendo disponibilizado por algum nó, é possível que não seja encontrado pelo processo de busca.

4.3 Identificação do Tráfego P2P

Diversos motivos levaram à pesquisa sobre identificação do tráfego gerado por aplicativos P2P. De acordo com Sen e Wang (2004); Karagiannis et al. (2005); Moore e Zuev (2005); Sun et al. (2010), podem ser citados:

- Engenharia de tráfego – gerentes de grandes redes precisavam de um meio de monitorar o comportamento da rede e de seus elementos para fins de engenharia de tráfego como, por exemplo, com a identificação de tráfego anômalo ou de problemas de roteamento;

- Manutenção da Qualidade de Serviço (*Quality of Service* - QoS) e de custos – gerentes de redes e ISPs necessitavam de uma maneira de diminuir a prioridade de pacotes de dados de aplicativos P2P, mantendo a QoS em um nível adequado para os usuários da rede, sem a necessidade de maiores investimentos em infraestrutura;

- Segurança e detecção de intrusão – diversos programas maliciosos, como vírus e *botnets*, utilizam-se de redes P2P para o controle de computadores infectados;

- Persecução penal – entidades privadas e governamentais precisavam de formas de denunciar e punir condutas criminosas que empregavam redes P2P para a prática de delitos.

Tendo em vista o desenvolvimento de sistemas de identificação criados para detectar o fluxo gerado por aplicativos P2P, esses programas sofisticaram os protocolos, adotando diversas técnicas para impedir a identificação do respectivo fluxo pelas ferramentas existentes.

Para Feng (2010), a técnica de classificação de tráfego envolve duas etapas principais: (1) a análise do fluxo é executada para extração de certas informações que caracterizam esse tipo de transferência de dados, como o cabeçalho ou conteúdo do pacote (*payload*) ou informações do fluxo; (2) o tráfego desconhecido pode ser classificado comparando-se as informações obtidas na etapa anterior.

Feng (2010) apresenta uma taxonomia dos métodos de classificação, que podem ser divididos em: (1) classificação baseada em portas (Seção 4.3.1); (2) classificação baseada em assinaturas (Seção 4.3.2); e (3) classificação baseada em características do fluxo (Seção 4.3.3). Além disso, também é possível a combinação de algumas dessas técnicas com o objetivo de melhorar a taxa de reconhecimento de aplicações P2P.

Xu et al. (2009) propõem que, além dos três métodos apresentados por Feng (2010), também seja incluída a utilização de IA para a identificação de tráfego P2P (Seção 4.3.4), já que a descoberta de conhecimento (*knowledge discovery*) e mineração de dados podem ser aplicadas no reconhecimento de padrões.

Quanto ao momento de classificação dos pacotes de rede, existem duas possibilidades (Callado et al., 2009; John et al., 2010):

1. Classificação *online* – os pacotes são analisados à medida que passam por um determinado ponto na rede, possibilitando a tomada de alguma medida nesse momento como, por exemplo, a diminuição da prioridade do tráfego ou a limitação do uso de largura de banda (*throughput*);

2. Classificação *offline* – os pacotes e fluxos de dados são armazenados e classificados posteriormente, não sendo possível, neste caso, a adoção de alguma medida restritiva sobre os dados.

Em relação à forma de monitoramento do fluxo de dados, de acordo com Callado et al. (2009); John et al. (2010), existem duas classificações:

1. *Monitoramento passivo*, onde o tráfego que passa pela rede somente é redirecionado, registrado ou analisado, sem nenhum tipo de alteração;

2. *Monitoramento ativo*, no qual ocorre uma modificação do fluxo através da inserção de pacotes na rede, supressão de pacotes ou sua modificação.

As forças policiais, quando realizam interceptações telemáticas, utilizam a classificação *offline* e o monitoramento passivo. Dessa forma, uma cópia fiel do fluxo de dados é enviado, por parte das operadoras de telecomunicação ou pelos ISPs, para ser analisado pelas forças policiais. Para que essa prova seja aceita judicialmente, não pode ocorrer inserção, supressão ou modificação de pacotes de dados.

4.3.1 Classificação baseada em portas

Segundo Feng (2010), a primeira forma de identificação de fluxos de dados relacionados a aplicativos P2P emprega o número da porta utilizada na camada de transporte pelos protocolos TCP ou UDP. A porta é representada, nesses protocolos, como um número de 16 bits, que pode variar de 0 a 65535. As portas podem ser divididas em três grupos:

1. de 0 a 1023 são as portas bem conhecidas (*well-known ports*) e são atribuídas pela *Internet Assigned Numbers Authority* (IANA) [6];

2. de 1024 até 49151 são conhecidas como portas registradas (*registered ports*) e também são inscritas na IANA;

3. a partir de 49152 até 65535 são as chamadas portas dinâmicas, privadas ou efêmeras (*dynamic*, *private* ou *ephemeral ports*).

A utilização do número da porta dos protocolos de transporte TCP e UDP como método para a classificação de fluxo apresenta diversas vantagens:

- É o método mais fácil de ser implementado – basta verificar se os números das portas utilizadas na camada de transporte (protocolos TCP ou UDP) correspondem aos das portas utilizadas pelos aplicativos P2P.

- Não é intrusivo, ou seja, não é necessário o acesso ao conteúdo (*payload*) do pacote de rede, mantendo a privacidade dos usuários.

[6] A relação completa das portas registradas pode ser obtida em `http://www.iana.org/assignments/port-numbers`

- Basta identificar um pacote do fluxo para que todo o fluxo possa ser considerado P2P, já que as portas são as mesmas durante toda a conexão.
- Funciona mesmo que o aplicativo P2P utilize criptografia dos dados.
- Pode ser aplicado em canais com alto tráfego de dados por ser bastante rápido.
- Pode ser facilmente aplicado em roteadores.

A Tabela 4.1 apresenta o protocolo utilizado e números de portas conhecidas de algumas aplicações P2P.

Tabela 4.1: Números de portas conhecidas de algumas aplicações P2P, conforme proposto por Chen et al. (2006) e Liang e Kumar (2005)

Aplicação P2P	**Número da Porta**	**Protocolo**
Limewire	6346/6347	TCP/UDP
Morpheus	6346/6347	TCP/UDP
BearShare	6346	TCP/UDP
eDonkey	4662	TCP/UDP
eDonkey2000	4661-4665	TCP/UDP
EMule	4662	TCP
	4672	UDP
BitTorrent	6881-6889	TCP/UDP
WinMx	6699	TCP
	6257	UDP
FastIdentify	1214	TCP/UDP
Gnutella	6346-6347	TCP/UDP
MP2P	41170	TCP/UDP
DirectConnect	411-412	TCP/UDP
Kazaa	1214	TCP/UDP

Durante nove dias, Saroiu et al. (2002) monitorou o *link* da Universidade de Washington e, com base nas portas utilizadas, identificou que a maior parte do tráfego era originada dos aplicativos P2P *Kazaa* e *Gnutella*. Também analisou o impacto que um sistema de cache P2P teria no cenário analisado, com uma diminuição de mais de 40% do tráfego total.

Foi realizada uma análise do comportamento do fluxo de dados, em Sen e Wang (2002, 2004), de três redes P2P (*Gnutella*, *FastTrack* e *DirectConnect*) e diversos aplicativos (*Bearshare*, *Limewire*, *Kazaa*, *Grokster* e *Morpheus*). O critério para a identificação do fluxo de dados P2P foi o conjunto de portas utilizadas por tais aplicativos, sendo o estudo beneficiado por esta facilidade. Os dados analisados (800 milhões de registros) foram obtidos de roteadores em diversos pontos do *backbone* de um ISP, através da utilização do serviço *NetFlow*, da Cisco.

Tutschku (2004) e Plissonneau et al. (2006) também realizaram estudos de identificação de fluxo P2P, mas desta vez gerado pelo *eMule*, através do monitoramento das portas utilizadas pelo aplicativo, obtendo bons resultados.

Diversos estudos (Gerber et al., 2003; Karagiannis et al., 2004; Sen et al., 2004; Hurley et al., 2011) apontam que o monitoramento por portas não é mais eficaz, pois a maioria dos aplicativos P2P passou a utilizar, para comunicação, portas aleatórias ou então portas registradas na IANA, como 21 (FTP), 80 (HTTP), 53 (*Domain Name System* (DNS)) e 443 (*Hypertext Transfer Protocol Secure* (HTTPS)).

Karagiannis et al. (2004) afirmam que, em 2002, apenas duas redes P2P (*eDonkey* e *FastTrack*) utilizavam portas aleatórias. Nas outras redes, apesar dos aplicativos suportarem a escolha de portas dinâmicas, esse recurso não era ativado por padrão. Esse cenário modificou-se a partir de 2003, onde a maioria dos aplicativos passou a utilizar portas aleatórias de forma automática, sem a necessidade de intervenção por parte do usuário.

No monitoramento realizado em um *link* da *France Telecom*, em 2003, Azzouna e Guillemin (2004), apenas com base no número da porta, identificaram que mais de 56% do tráfego total analisado correspondia ao gerado por aplicativos P2P, principalmente o *eDonkey*, que respondia, sozinho, por 50% do tráfego total. Não foi identificado 26% do tráfego total mas, pela utilização de portas dinâmicas por parte dos aplicativos P2P mais recentes, provavelmente são referentes a aplicativos P2P.

Gerber et al. (2003) apresentaram os resultados do monitoramento no *backbone* de um grande ISP, realizado em 2002 e 2003, que indica que, em junho de 2002, apenas três portas (relacionadas a aplicativos P2P) eram responsáveis por 60% do tráfego. Esse percentual desceu para 20% em fevereiro de 2003, sendo necessário agrupar 1000 portas para perfazer 60% do tráfego, indicando a distribuição das portas utilizadas pelos programas P2P.

Com o aumento do tráfego de dados gerado por aplicativos P2P, as empresas responsáveis pelo fornecimento do acesso à Internet (ISPs) e gerentes de rede passaram a bloquear ou limitar o fluxo de dados desses aplicativos. Foram motivados principalmente pelo aumento dos custos e problemas legais inerentes à distribuição de dados com infração a direitos autorais ou de conteúdo proibido.

Como reação às limitações impostas, que identificavam os fluxos P2P através da porta utilizada, esses aplicativos passaram a utilizar portas aleatórias, sendo empregadas até mesmo portas registradas na IANA. Além disso, com a disseminação do uso de *network address translation* (NAT), em algumas conexões o número da porta é alterado pelo

equipamento responsável pelo NAT, impedindo a correta identificação de um fluxo P2P e resultando em classificações incorretas (falsos negativos e falsos positivos).

Dessa forma, a identificação por portas utilizadas na camada de transporte pelos protocolos TCP e UDP tornou-se ineficiente, tendo sido necessário o desenvolvimento de novas metodologias para a identificação do fluxo de rede desses aplicativos.

4.3.2 Classificação baseada em assinaturas

Em resposta ao emprego de portas aleatórias por parte dos aplicativos P2P, passou-se a utilizar a classificação baseada em assinaturas para identificação do fluxo de dados gerado por esses programas. Esse monitoramento faz uma análise dos dados encontrados no pacote de comunicação (*payload*), buscando por sequências de caracteres (assinaturas) criadas por aplicativos P2P conhecidos. Entretanto, para que esse reconhecimento seja eficiente, é necessário identificar as assinaturas geradas por vários aplicativos P2P diferentes.

A Tabela 4.2 apresenta assinaturas de aplicativos P2P adaptadas de Karagiannis et al. (2003) e Feng (2010). O termo "`0x`" indica valores em hexadecimal e valores entre aspas indicam uma *string*.

Tabela 4.2: Assinaturas de alguns aplicativos P2P, adaptadas de Karagiannis et al. (2003) e Feng (2010)

Aplicação P2P	**Assinatura**
BitTorrent	`0x13`"Bit"
	"GET /announce?info_hash"
	"GET /torrents/"
	"GET TrackPack"
	`0x13`"BitTorrent"
	`0x00000005`
	`0x0000000D`
	`0x00004009`
eDonkey	`0xE319010000`
	`0xE3`
	`0xC5`
Gnutella	"GUNT"
	"GIV"
Kazaa	"X-Kazaa"
	`0x270000002980`
	`0x280000002900`
	`0x29000000`
	`0xC028`

Karagiannis et al. (2003) empregam uma série de heurísticas para identificar tráfego P2P pois, dependendo do protocolo e da métrica utilizada, de 30% a 70% do tráfego P2P não foi detectado pela classificação baseada em portas.

No mesmo sentido, Bleul et al. (2006a) criaram um método de identificação baseado na checagem de assinaturas e de informações geradas pela camada de aplicação. Esse trabalho foi inspirado no trabalho de Sen et al. (2004), onde são utilizadas técnicas de sistemas de detecção de intrusão (*Intrusion Detection Systems* (IDS)) para identificar conteúdo específico em pacotes de rede.

Em um trabalho posterior, Bleul et al. (2006b) apresentaram o resultado de um monitoramento no qual foram empregadas assinaturas para a identificação do fluxo P2P com o aplicativo *Netfilter*. Segundo os autores, a taxa de identificação do *eMule* ficou por volta de 92%.

Em Bolla et al. (2008) são utilizadas assinaturas para apontar a existência de dois grupos distintos de fluxos de dados relacionados a aplicativos P2P: sinalização e transferência de dados. Nesse trabalho, são apresentadas as diferenças em quatro métricas utilizadas para diferenciar esses grupos de comunicação: tempo de chegada, duração, volume e tamanhos médio dos pacotes.

Wang et al. (2010b) demonstra a utilização de IA para a identificação automática de assinaturas em tráfego desconhecido. Essas assinaturas podem ser empregadas, posteriormente, para a identificação dos aplicativos que geraram esse fluxo.

Nos trabalhos de Keralapura et al. (2009, 2010) foi descrita uma aplicação de dois estágios para classificação de tráfego P2P. A primeira etapa identifica o fluxo P2P através de uma métrica temporal entre os fluxos. O segundo estágio extrai, de forma automática, as assinaturas necessárias para identificar o fluxo P2P.

Zhang et al. (2010) asseguram que o método de identificação por assinaturas pode alcançar altas taxas de identificação e não é influenciado pela utilização de portas aleatórias. Entretanto, depende de duas suposições: (1) deve existir um conhecimento prévio dos protocolos utilizados; e (2) deve ser fácil analisar o conteúdo dos pacotes de rede.

Alguns problemas com a utilização de classificação por assinaturas são apresentadas por Haffner et al. (2005). Dentre eles, pode-se citar:

- a existência de grande diversidade de aplicativos, além de variações dentro de um mesmo aplicativo entre suas versões;

- pouca ou nenhuma documentação, pois em muitos casos tratam-se de protocolos proprietários e fechados, dificultando a obtenção de informações sobre as assinaturas;
- necessidade de análise profunda do protocolo para identificar trechos de potenciais assinaturas;
- alto poder computacional necessário;
- baixa escalabilidade.

Em virtude desses problemas, nesse estudo foi desenvolvido uma metodologia para identificação automática de assinaturas através da aplicação de técnicas de inteligência artificial. Foram aplicados três algoritmos de aprendizado (*Naïve Bayes*, *AdaBoost* e *Maximum Entropy*) em aplicações comuns, como FTP, SMTP, HTTP e HTTPS (mas nenhum P2P). Foi obtida uma taxa de acerto de mais de 99%, com exceção do *Secure Shell* (SSH), que ficou em 86%.

Em Sen et al. (2004) é apresentada uma aplicação que utiliza IDS e assinaturas para detectar e filtrar o tráfego P2P, obtendo um grau de identificação superior quando comparado com a classificação baseada em portas. Entretanto, os autores também afirmam que, futuramente, a classificação com base em assinaturas terá o mesmo fim da que é baseada em portas, pois os aplicativos irão utilizar criptografia para evitar a detecção.

Embora algumas abordagens tenham obtido alto grau de acerto na identificação de aplicações P2P, a tendência é que a taxa de acerto obtida com o emprego dessa metodologia diminua com o tempo. Karagiannis et al. (2004) afirmam que a classificação com base em assinaturas tende a ser cada vez mais ineficiente, pois um número crescente de aplicativos P2P vêm utilizando criptografia para a transmissão de pacotes e transferência de arquivos, inviabilizando a classificação com base na análise no *payload* dos pacotes de rede.

4.3.3 Classificação baseada em características do fluxo

A classificação baseada em características de fluxo observa o comportamento específico de uma determinada aplicação ou grupo de aplicações na rede. Como exemplos de comportamento pode-se citar o tempo entre a chegada de pacotes, a porta utilizada, a duração do fluxo e a relação entre dados enviados e recebidos. Ressalta-se que a porta da camada de transporte pode ser utilizada, mas não como único método de identificação de fluxo de dados.

Para Zhang et al. (2010), pacotes com a mesma origem (endereço IP e porta), destino (endereço IP e porta) e protocolo da camada de transporte (TCP ou UDP), transferidos durante um determinado período de tempo, são considerados como pertencentes ao mesmo fluxo.

De acordo com Dai et al. (2010), as maiores tecnologias atuais para identificação de fluxo P2P são a *Deep Packet Inspection* (DPI) e *Deep Flow Inspection* (DFI). O DPI realiza uma análise no *payload* do pacote, buscando por características específicas. Este método possui uma alta taxa de acerto, mas possui algumas limitações. Em primeiro lugar, não consegue identificar novas aplicações das quais ainda não havia obtido as informações necessárias. Em segundo lugar, caso seja empregada criptografia ou o protocolo seja proprietário, é muito difícil ou até mesmo impossível identificar corretamente o pacote, e criptografia parece ser uma solução a ser cada vez mais utilizada por aplicativos P2P. Por fim, o uso de memória é muito alto, diminuindo a escalabilidade dessa solução. Esse autor apresenta uma metodologia que emprega as vantagens do DFI e tenta minimizar as desvantagens do DPI. A Tabela 4.3 apresenta um quadro comparativo entre o DPI e o DFI.

Tabela 4.3: Comparação entre o DPI e o DFI para a identificação de dados P2P, conforme Dai et al. (2010)

Diferença	**DPI**	**DFI**
Objeto de análise	Pacote	Fluxo
Taxa de acerto	Bom	Ruim
Suporte a dados cifrados	Não	Sim
Complexidade	Ruim	Boa

No trabalho de Crotti et al. (2007), somente o tamanho e o tempo de chegada entre os primeiros n pacotes são utilizados para criar uma identificação estatística (uma impressão digital) de um protocolo da camada de aplicação. Essa identificação estatística, criada através de técnicas de agrupamento, é utilizada para medir a similaridade de um certo fluxo ao protocolo correspondente.

Cheng et al. (2008) apresenta um método de identificação de fluxo baseado na distribuição de *hosts* remotos e na distribuição de portas remotas. Com esse método, o autor apresenta índice de acerto de aproximadamente 95% no reconhecimento de tráfego gerado pelo aplicativo BitTorrent.

Hu et al. (2009) apresentam uma estratégia de classificação que utiliza diversas variáveis referentes ao fluxo de dados como por exemplo, o número de pacotes, quantidade de bytes, tamanho do primeiro e segundo pacote de dados, duração do fluxo, média do

tamanho dos pacotes, variação do tempo de chegada dos pacotes. Para o aplicativo BitTorrent, foi atingida uma taxa de sucesso na classificação de aproximadamente 95%.

O aplicativo denominado BLINC, apresentado no trabalho de Karagiannis et al. (2005), utiliza a observação e identificação de padrões encontrados na camada de transporte. Esses padrões são analisados em três níveis crescentes de detalhes: social, funcional e de aplicação. Essa abordagem não utiliza o acesso ao conteúdo do pacote, número de portas ou outra informação adicional além das encontradas pelos aplicativos de análise de comportamento de fluxo. Os autores criaram uma associação entre um *host* da Internet com as aplicações que são executadas nele, necessitando de vários fluxos de dados para uma correta identificação. Dessa forma, esse método não consegue classificar um fluxo de dados isolado.

Segundo Feng (2010), a maioria das técnicas são demoradas pois necessitam de recursos computacionais intensos, pois devem ser aplicadas a todos os fluxos que passarem por determinado ponto, além de não conseguirem identificar programas desconhecidos, funcionando somente para os que já haviam sido classificados anteriormente. Além disso, conforme Iacovazzi e Baiocchi (2010), grande parte das técnicas de identificação por características do fluxo utiliza o tamanho do pacote como base. Essa característica pode ser facilmente alterada pelos aplicativos P2P através do preenchimento (*padding*) e da fragmentação dos pacotes, aliados à criptografia.

4.3.4 Classificação com a utilização de Inteligência Artificial

Como visto, as classificações do fluxo de programas P2P baseadas em portas, assinaturas e características do fluxo de dados possuem méritos e problemas. A Tabela 4.4 sintetiza algumas das características dessas técnicas de classificação, conforme Feng (2010).

Tabela 4.4: Comparação entre métodos de classificação de fluxo de dados, de acordo com Feng (2010)

Classificação	**Características**					
	Exatidão	Escalabilidade	Robustez	Desempenho	Capacidade de classificação	Suporta criptografia
Baseada em portas	Boa	Ruim	Ruim	Bom	Sim	Sim
Baseada em assinaturas	Boa	Ruim	Boa	Ruim	Sim	Não
Baseada em características do fluxo	Ruim	Boa	Ruim	Bom	Não	Sim

De acordo com Wang et al. (2010a), as técnicas de IA para a identificação do fluxo de dados podem ser divididas em duas classes:

1. Agrupamento (*clustering*) ou aprendizagem não supervisionada (*non supervised learning*), tendo como exemplos:
    - K-médias (*K-Means*) (Bernaille et al., 2006; Erman et al., 2007c)
    - *Expectation-maximization* (EM) (McGregor et al., 2004)
    - AutoClass (Erman et al., 2006)
2. Aprendizagem supervisionada (*supervised learning*), tendo como exemplos:
    - Árvores de decisão (Zhang et al., 2010)
    - Redes Bayesianas (Auld et al., 2007; Park et al., 2008)
    - *Naïve Bayes* (Moore e Zuev, 2005; Bonfiglio et al., 2007)
    - *Support vector machine* (SVM) (Li et al., 2007; Este et al., 2009)
    - *Multi-layer Perceptron* (MLP) (Park et al., 2008; Chen et al., 2009; Braga, 2007)

Uma técnica de identificação de aplicações, bastante útil para monitoramentos *online*, baseada na observação do tamanho dos cinco primeiros pacotes de dados em um fluxo TCP, é proposta em Bernaille et al. (2006). Emprega clusterização não supervisionada, utilizando o algoritmo *K-Means*, para criar as classes das aplicações com base no tamanho dos pacotes.

Erman et al. (2006) apresentaram uma abordagem que emprega um algoritmo de classificação não supervisionado denominado AutoClass para a classificação do fluxo de rede, com uma taxa geral de acerto de mais de 90%. Foi obtida uma taxa de 80% de identificação do tráfego do aplicativo *Limewire*, o único aplicativo P2P analisado. Erman et al. (2007c) obtiveram uma precisão na classificação de 95% em relação aos fluxos e 80% em relação à quantidade de bytes transferidos com o emprego do algoritmo de clusterização *K-Means*. Em relação aos aplicativos P2P analisados, a taxa de identificação foi de 81,32%.

McGregor et al. (2004) utilizaram o algoritmo EM para a classificação de fluxos tendo como base o tempo de chegada entre pacotes dividido pelo tamanho dos pacotes e que indicam o tipo de aplicação que gerou esses dados. Com esse método, diversos protocolos foram classificados em vários agrupamentos, mas nenhum aplicativo P2P.

Zhang et al. (2010) empregaram árvores de decisão, com o algoritmo C4.5, para chegar a uma taxa de identificação de 96.7% do tráfego P2P. Entretanto, a comparação foi realizada apenas para agrupar diferentes classes de aplicativos P2P, como comunicação instantânea (*Skype*), *streaming* (*PPLive* e *PPStream*) e compartilhamento de arquivos (*eMule*, *BitTorrent* e *Thunder*). Não foram realizadas comparações entre fluxos gerados por aplicativos P2P e fluxos gerados por outros programas.

Auld et al. (2007) utilizaram redes Bayesianas para a classificação de fluxos, empregando 249 características baseadas no cabeçalho dos pacotes de rede (sem o número da porta ou o endereço IP). Chegaram a uma precisão de 97,2% na identificação do tráfego dos aplicativos *Kazaa*, *BitTorrent* e *Gnutella*.

Park et al. (2008) empregaram quatro algoritmos diferentes: o *Multilayer Perceptron* (MLP), J48, REPTree e Redes Bayesianas para a classificação do tráfego de rede, obtendo uma precisão de 80% a 90%. As características escolhidas para a classificação foram: endereços IP de origem e destino, portas de origem e destino, total de bytes transferidos, duração da conexão, tamanho do pacote e tempo de chegada entre pacotes. Foram analisados os aplicativos P2P *BitTorrent*, *Fileguri*, *Soribada* e *Gample*.

É utilizada IA para classificar fluxo de rede em Moore e Zuev (2005). Os autores empregaram *Naïve Bayes* para categorizar o tráfego, classificando-o por aplicação. A taxa de acertos geral variou de 65%, na classificação simples, a 95%, quando combinado com outras técnicas para redução de variáveis. Foi obtida a taxa de 55,18% para os aplicativos P2P analisados (*Kazaa*, *BitTorrent* e *Gnutella*).

Bonfiglio et al. (2007) utilizaram uma técnica composta por dois passos para a identificação do tráfego gerado pelo aplicativo Skype. Inicialmente, identificaram a aleatoriedade do conteúdo dos pacotes gerados pela criptografia empregada pelo *Skype*. Em seguida, empregaram *Naïve Bayes* utilizando o tempo de chegada entre pacotes e o tamanho de cada pacote. Foram identificados aproximadamente 100% do tráfego do *Skype*, independentemente da utilização de criptografia.

Li et al. (2007) empregaram SVM para a classificação do fluxo gerado por sete classes de aplicações, cada uma com alguns aplicativos representativos da classe. A classe P2P apresenta os programas *MSN*, *Soulseek*, *Skype*, *BitTorrent*, *eDonkey*, *Qq* e *100bao*. Os autores selecionaram dezenove características do fluxo para que fosse possível a classificação *online*. O sucesso geral aproximado de detecção foi de 96,9% a 99,4%. Em relação aos aplicativos P2P, a taxa de identificação foi de 95,18%. Os autores não apresentaram informações sobre a utilização de criptografia ou qual a configuração de cada aplicativo P2P.

Este et al. (2009) apresentaram o resultado do emprego de SVM para a classificação de tráfego em redes IP. Neste trabalho são apresentados comentários acerca das questões que devem ser respondidas para que possa ser utilizado SVM , de forma abrangente, na classificação de fluxos de rede. A característica utilizada foi o tamanho do *payload* do pacote de rede. Os fluxos analisados foram capturados entre 2002 e 2007. Foram analisados os fluxos dos aplicativos *BitTorrent* e *MSN*, tendo sido obtida uma taxa de identificação de 91,2% a 96,8%. A identificação foi realizada com base na porta utilizada pelas aplicações.

Chen et al. (2009) propuseram a utilização de MLP para a classificação do tráfego P2P. Foi implementado um protótipo que realizou um treinamento *offline* e, posteriormente, foi aplicado durante uma classificação *online*, obtendo 96,5% de precisão na classificação do fluxo de dados P2P. Os aplicativos analisados foram o *BitTorrent* e o *QQLive*.

Braga (2007) apresentou uma RNA MLP para a identificação do tráfego dos aplicativos *BitTorrent* e *eMule*. O tráfego gerado por esses aplicativos P2P foi comparado apenas com os protocolos HTTP e FTP, tendo sido obtida uma taxa de identificação de 85%.

Além desses trabalhos, também foram apresentados artigos nos quais são comparados o desempenho de diversos algoritmos de IA para classificação de fluxo de dados. Um exemplo é o trabalho de Williams et al. (2006), onde é apresentada uma avaliação dos algoritmos *C4.5 Decision Tree*, *Naïve Bayes*, *Nearest Neighbour*, *Naïve Bayes Tree*, *MLP*, *Sequential Minimal Optimization* e *Redes Bayesianas*. Os autores também apresentam uma análise de diversas características utilizadas durante o processo de classificação, concluindo que existe grande potencial para a identificação de fluxos de rede com emprego de IA.

Segundo a classificação apresentada por Brownlee e Claffy (2002), o fluxo de dados gerado por aplicativos P2P pode ser dividido pelo seu tamanho e por sua duração. Os fluxos de grande volume de dados (ou seja, se o tamanho for maior que x) são chamados de "elefantes" (*elephants*), e os de baixo volume (menores que x) são os "camundongos" (*mice*). Os fluxos de baixa duração são as "libélulas" e os de alta duração são as "tartarugas". Nesse estudo, foi identificado que embora (45%) dos fluxos sejam de curtíssima duração (menores que dois segundos) e 98% dos fluxos sejam menores que 15 minutos, os fluxos de alta duração que duram horas e até mesmo dias respondem por 50% a 60% do total do tráfego de dados.

Erman et al. (2007a) avaliou os dados capturados por seis meses na Universidade de Calgary (apresentado em Erman et al. (2007b)) e estimou que se x for considerado como sendo 228 KB, 1% dos fluxos respondem por 73% do tráfego e para x igual a 3,7 MB, 0,1% dos fluxos correspondem a 46% do tráfego. Dessa forma, mesmo com

uma precisão de 99,9%, ainda é possível uma classificação incorreta de 46% do total de dados transferidos.

Sen e Wang (2004) classificaram o tráfego gerado por aplicativos P2P em dois grandes grupos: sinalização e transferência de dados. Estão incluídos na sinalização os pacotes utilizados para o estabelecimento das conexões, buscas por arquivos e respostas a essas buscas. Agrupados na transferência de dados estão os pacotes efetivamente relacionados ao envio e recebimento de arquivos.

O foco da maioria dos sistemas de classificação é na detecção da maior quantidade de bytes possível – focando-se nos fluxos "elefantes". Entretanto, para fins periciais, os fluxos de sinalização ("camundongos" ou "libélulas") são extremamente importantes, pois são nesses fluxos que estão os pacotes responsáveis pelas divulgação dos arquivos compartilhados, pelas buscas realizadas, pelos pedidos de arquivos e pelas conversas realizadas. É nesse tipo de fluxo, normalmente, que se pode encontrar provas da intenção (dolo) do agente de buscar, receber e compartilhar arquivos contendo cenas de exploração sexual de crianças e adolescentes.

Uma crítica de Salgarelli et al. (2007) é que já foram publicados diversos trabalhos sobre a identificação de fluxos de Internet, mas não existe uma base objetiva e científica para avaliar os resultados, seja por falta de informações claras sobre as técnicas ou simplesmente pela diferença no conjunto de dados que foram utilizados para o treinamento e os testes das técnicas de IA.

A contribuição do presente trabalho é que, diferentemente dos trabalhos apresentados anteriormente, onde o foco é a classificação do maior número de bytes ou de fluxos, o objetivo desta dissertação é a obtenção de informações de relevância forense, com a finalidade de comprovação de autoria e materialidade para uso durante a persecução penal. Essa diferença é significativa, pois mesmo que uma determinada técnica identifique corretamente 99,9% dos bytes transferidos, informações que poderiam comprovar o dolo ou a autoria poderiam ser perdidos, pois os pacotes que transportam esse tipo de informação são pequenos, influenciando pouco a taxa de acertos dessa técnica.

Isso não quer dizer que os fluxos grandes ("elefantes") não sejam relevantes, pois nesses fluxos estão presentes os pacotes referentes às transferências de arquivo propriamente ditas. Dessa forma, nesses fluxos pode-se encontrar a materialidade dos crimes previstos nos artigos 240 e 241 da Lei 8.069/90 (Estatuto da Criança e do Adolescente) (Brasil, 1990).

4.4 eMule

O *eMule* é um aplicativo de compartilhamento de arquivos P2P baseado no programa eDonkey2000. Foi lançado em 13 de maio de 2002, por Hendrik Breitkreuz (também conhecido como Merkur), insatisfeito com o cliente eDonkey2000. Para desenvolver o *eMule*, foi realizado um processo de engenharia reversa no protocolo utilizado na rede do eDonkey2000, que era proprietário e fechado.

O eDonkey2000 foi criado pela empresa MetaMachine em 2000, logo após o lançamento do programa Napster, apresentando algumas melhorias em relação a este aplicativo: (1) possibilitava que o cliente baixasse partes do mesmo arquivo simultaneamente de vários nós; (2) os servidores formavam uma rede de busca, redirecionando buscas de usuários conectados em outros servidores e com isso aumentando as fontes de um arquivo; e (3) identificação de arquivos com base no código *hash* ao invés do nome do arquivo.

O programa eDonkey2000 teve suporte até 2005 e, em 2006, a empresa MetaMachine concordou em pagar à *Recording Industry Association of America* (RIAA) 30 milhões de dólares por infrações de direitos autorais realizados com o eDonkey 2000 e parou de dar suporte à rede eD2k.

Diversas funcionalidade foram adicionadas pelo *eMule* em relação ao eDonkey2000, podendo-se afirmar que se trata de um novo sistema (Caviglione e Davoli, 2008). Dentre elas pode-se citar: sistema de créditos, compressão dos dados enviados pela rede e utilização da rede KAD. Além disso, o *eMule* permite que diversos arquivos sejam baixados simultaneamente e cada arquivo pode ser baixado de vários clientes ao mesmo tempo (Sheng et al., 2010).

Hoje, o *eMule* é um dos aplicativos P2P mais utilizados em todo o mundo, sendo o aplicativo mais baixado do *SourceForge*, com mais de 560 milhões de *downloads* até agosto/2011, segundo a lista do *SourceForge Top Downloads* [7].

Desde julho de 2002, o *eMule* é um *software* livre, distribuído sob a *GNU General Public License*, tendo seu código-fonte disponibilizado no sítio oficial do aplicativo [8] em linguagem Microsoft Visual C++.

Em virtude da popularidade e da disponibilização do código-fonte, o *eMule* foi utilizado como base para diversos outros clientes, chamados de *mods* (modificações) do cliente *eMule* original. São exemplos dessas modificações o *aMule*, *jMule*, *xMule*, *MLDonkey*

[7] Disponível em: `http://sourceforge.net/top/topalltime.php?type=downloads`
[8] `http://www.emule-project.net/`

e o *DreaMule*. A versão atual do *eMule* (*0.50a*), lançada em 07 de abril de 2010, pode acessar as redes (*overlay*) KAD e eD2k.

Embora utilize servidores como o Napster, qualquer pessoa pode criar um servidor do *eMule*, caracterizando a rede eD2k, quanto à estrutura de rede, como estruturada, e quanto ao grau de centralização, como híbrida descentralizada.

A partir da versão 0.40, o *eMule* passou a acessar a rede KAD (com algumas modificações no projeto original), uma implementação de uma *Distributed hash table* (DHT) que não possui servidores ou outra forma de centralização. Nessa rede, todos os nós são clientes e servidores simultaneamente (Caviglione e Davoli, 2008).

Dessa forma, a rede KAD, por sua vez, pode ser classificada como uma rede desestruturada (quanto à estrutura de rede) e em puramente descentralizada (quanto ao grau de centralização).

4.4.1 Funcionamento do eMule

Nesta seção, serão apresentados os conceitos básicos relacionados ao funcionamento do *eMule*. Essas informações foram obtidas a partir da documentação oficial do *eMule* e da análise do código-fonte desse aplicativo [9].

Identificação dos usuários do eMule (*User ID*)

Quando o eMule é executado pela primeira vez, é criado um identificador do usuário denominado *User ID* ou *User Hash*. Esse identificador, que possui 16 bytes de tamanho, é gerado aleatoriamente, sendo substituídos o 6º byte pelo valor decimal 14 (`0x0E` [10]) e o 15º byte pelo valor decimal 111 (`0x6F`).

Identificação da conexão do cliente e do servidor do eMule (*Client ID*)

Quando é estabelecida uma conexão TCP entre o cliente e o servidor do *eMule*, é atribuído um identificador para esta conexão, denominado *Client ID*. Este identificador, composto por quatro bytes, é válido somente enquanto for mantida a conexão entre o cliente e o servidor.

O *Client ID* é dividido em dois conjuntos: (1) *High ID*, quando o cliente consegue estabelecer conexões TCP de entrada; e (2) *Low ID*, quando não é possível estabelecer

[9] Disponível em: `http://sourceforge.net/projects/emule/files/`

[10] A notação "`0x`" é usada para representar valores em hexadecimal.

uma conexão TCP de entrada com esse cliente. Caso seja fornecido um *Low ID*, o cliente não conseguirá receber dados de outros clientes com *Low ID*, pois nenhum dos dois conseguirá estabelecer uma conexão TCP com o outro, limitando as fontes dos arquivos aos clientes com *High ID*. Além disso, alguns servidores limitam o número de clientes conectados com *Low ID*, desconectando os usuários que excederem esse limite.

A forma de cálculo do *High ID* é baseada no endereço IP. Dado um endereço IP $A.B.C.D$, o *High ID* é o resultado de: $A + (2^8 \times B) + (2^{16} \times C) + (2^{24} \times D)$.

Quanto à forma de cálculo do *Low ID*, é um número sequencial atribuído pelo servidor ao qual esse cliente se conectou e sempre será menor que 16777216 (`0x1000000`), não guardando relação com o endereço IP do cliente.

Identificação de arquivos (*File ID*)

Conforme exposto anteriormente, desde seu lançamento, o *eMule* utiliza um código, denominado *File ID*, para identificação dos arquivos, ao invés do nome do arquivo, como utilizado por outros aplicativos P2P na época. Os resultados das buscas por arquivos são agrupados por *File ID*, independentemente de seu nome. Essa mudança aumentou a confiabilidade do resultado das buscas, pois usuários mal intencionados poderiam modificar o nome de um arquivo, levando outros usuários inadvertidamente a baixá-los equivocadamente.

Esse código, com 16 bytes de tamanho, é calculado sobre o conteúdo do arquivo e é baseado no algoritmo MD4. Para seu cálculo, os arquivos são divididos em pedaços (*chunks*) de 9728000 bytes (9500 KB ou 9,28 MB) cada. Se o arquivo não possuir um tamanho que seja múltiplo desse valor, o último *chunk* será menor. Caso o arquivo seja menor que 9,28 MB, o identificador é o *hash* MD4 do conteúdo. Caso contrário, é calculado o *hash* de cada *chunk* e esses valores, em formato hexadecimal, são concatenados em uma sequencia denominada *Hashset*, sendo novamente calculado o *hash* MD4 sobre esse valor para a obtenção do *File ID*.

Detecção de corrupção no recebimento de arquivos

O primeiro sistema de detecção de corrupção durante o recebimento de arquivos foi o *Intelligent Corruption Handling* (ICH). Este sistema verifica a integridade de cada pedaço (*chunk*) dos arquivos. Caso detecte que um *chunk* está corrompido, o eMule inicia novamente o *download* desse *chunk* e verifica o *hash* (*File ID*) a cada 180 KB, até que o *File ID* esteja correto. O problema é que esse sistema não consegue verificar

a integridade de blocos menores, sendo necessário baixar os blocos de 180 KB até que o *File ID* esteja correto.

Em virtude desse desperdício de recursos do ICH, foi criado o *Advanced Intelligent Corruption Handling* (AICH). Neste sistema, cada *chunk* de 9,28 MB é dividido em 53 blocos (52 blocos de 180 KB e 1 bloco de 140 KB) e sobre cada bloco é calculado um código *hash* com o emprego do algoritmo SHA1. O *hash* de cada bloco, denominado *Block Hash*, é armazenado no nível mais baixo de uma árvore de *hashes* (*Hash Tree*). É calculado novamente o *hash* SHA1 sobre dois *Block Hashes* adjacentes, armazenando esse *hash* de verificação em um nível mais alto da *Hash Tree*. Esse processo se repete até que se chegue a um único *hash*, denominado *Root Hash*. Esse conjunto de *hashes*, denominado *Hashset AICH*, é proporcional ao tamanho do arquivo – são gerados 108 *hashes* para cada *chunk* de 9,28 MB.

Se o *Hashset AICH* estiver disponível, o *eMule* pode calcular os *Block Hashes* para identificar qual bloco está corrompido e baixar novamente somente este bloco (de 180 KB ou 140 KB de tamanho), diminuindo sensivelmente o tempo de recuperação de corrupção de arquivos quando comparado com o ICH.

Se o *eMule* não possuir o *Root Hash* de um arquivo que está sendo baixado, são escolhidos, de forma aleatória, 10 clientes que estão compartilhando o arquivo e é solicitado o *Root Hash* desse arquivo. Se todos enviarem um *Root Hash* idêntico, ele será utilizado para a verificação de corrupção desse arquivo (a documentação oficial do *eMule* fala em 92%, mas como são apenas 10 clientes, a resposta deve ser unânime). Esse *Root Hash*, por não ter vindo no atalho utilizado para baixar o arquivo pelo *eMule* (*eD2k link*), não será armazenado em disco e somente será válido para a sessão atual.

Se for identificado que um *chunk* desse arquivo está corrompido, é enviado o pedido de um pacote de recuperação (*Recovery Packet*) desse *chunk* para um cliente do *eMule*, escolhido aleatoriamente, que esteja compartilhando esse arquivo e que possua o *Hashset AICH* completo. Quando for recebido o *Recovery Packet*, o *eMule* verifica se o *Root Hash* é o mesmo que foi recebido no passo anterior. Se for idêntico, são calculados os *Block Hashes* e comparados com o nível mais baixo do *Hashset AICH*, para a identificação de qual bloco deve ser baixado novamente. A Figura 4.2 apresenta um exemplo da resposta por um pacote de recuperação do *chunk* número 2.

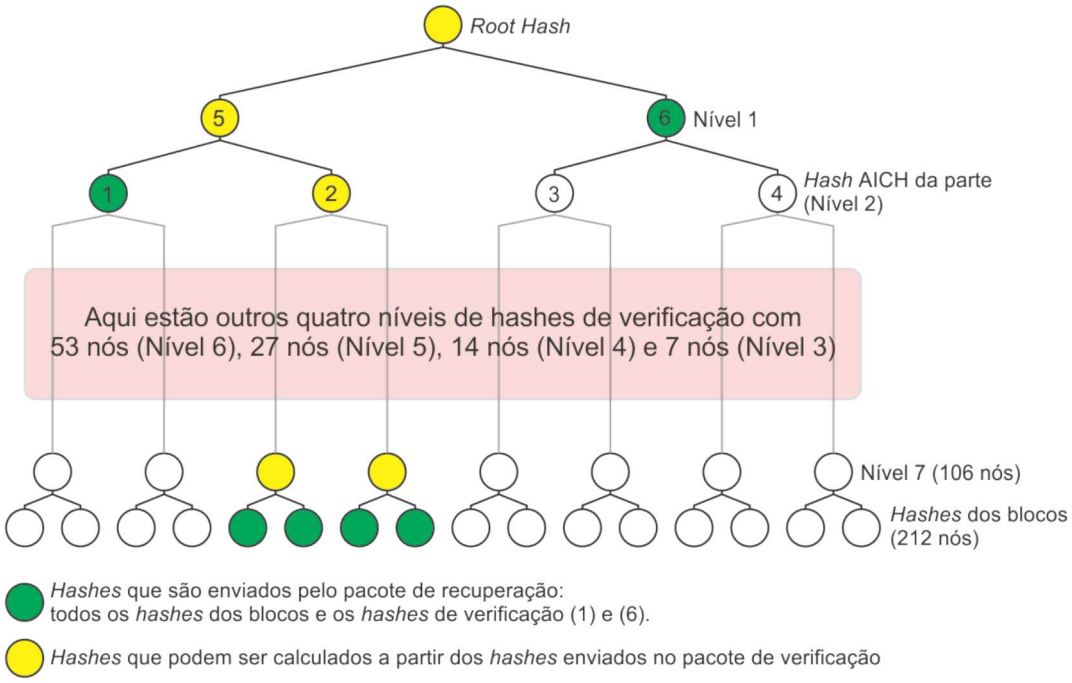


Figura 4.2: Exemplo de um pacote de recuperação de um arquivo com quatro *chunks*, adaptado da documentação oficial do eMule

Atalhos do eMule (*eD2k Links*)

Atalhos do *eMule* (*eD2k Links*) são atalhos especialmente criados para que operações possam ser realizadas diretamente pelo *eMule*. Os *eD2k Links*, cujos principais formatos são apresentados abaixo, sempre se iniciam com a sequência "`eD2k://|`".

1. Arquivo (formato básico) - os *eD2k Links* referentes a arquivos iniciam com "`eD2k://|file|`" e informam o nome do arquivo, seu tamanho e o *File ID*. Formato:
   `ed2k://|file|<nome>|<tamanho>|<File ID>|/`

2. Arquivo com *Hashset* - além das informações presentes no item anterior, esse atalho também contém o *Hashset*, inserido após os caracteres "`p=`", sendo os *hashes* dos *chunks* separados por dois pontos. Formato:
   `ed2k://|file|<nome>|<tamanho>|<File ID>|p=<Hashset>|/`

3. Arquivo com *Root Hash* - além das informações presentes no formato básico do *eD2k Link*, é informado o *Root Hash*, apresentado após os caracteres "`h=`". O *Root Hash* é utilizado no sistema de detecção AICH. Formato:
   `ed2k://|file|<nome>|<tamanho>|<File ID>|h=<Root Hash>|/`

4. Arquivo com fontes (endereço IP) - informa o endereço IP e porta de comunicação de um ou mais clientes que disponibilizam esse arquivo. Formato:
`ed2k://|file|<nome>|<tamanho>|<File ID>|/|sources,<IP:porta,...>|/`

5. Arquivo com fontes (nome de host) - informa o nome de *host* e porta de comunicação de um ou mais clientes que disponibilizam esse arquivo. Formato:
`ed2k://|file|<nome>|<tamanho>|<File ID>|/|sources,<host:porta,...>|/`

6. Arquivo com fontes (endereço HTTP) - informa o endereço HTTP onde são armazenadas fontes desse arquivo, após os caracteres "`s=`". Formato:
`ed2k://|file|<nome>|<tamanho>|<File ID>|s=http://sitio/arquivo|/`

7. Servidor do *eMule* - os *eD2k Links* que são utilizados para indicar servidores do *eMule* iniciam-se com "`eD2k://|server|`". Também são informados neste atalho o endereço IP e a porta de comunicação desse servidor. Formato:
`ed2k://|server|<IP>|<porta>|/`

8. Buscas - também podem ser utilizados *eD2k Links* para a realização de buscas por arquivos no *eMule*. Deve ser informado o termo utilizado para a busca. Formato:
`ed2k://|search|<termo>|/`

Sistema de créditos do eMule

Para incentivar o compartilhamento de arquivos e evitar que usuários utilizassem o *eMule* apenas para baixar arquivos, retirando-os do compartilhamento assim que fossem recebidos (*free riders*), o *eMule* passou a empregar um sistema de créditos a partir da versão 0.19a, lançada em setembro de 2002 (Li e Gruenbacher, 2010).

A largura de banda de envio de arquivos é dividido em n canais, onde os n clientes que tenham a maior pontuação na fila de envio são atendidos. O cálculo dessa pontuação é realizado pela multiplicação do número de segundos em que o cliente está na fila por 100 e o resultado é multiplicado pelos créditos que esse usuário possui.

Para identificar a quantidade de créditos que um cliente do *eMule* possui, são realizados dois cálculos (as medidas são em MB), conforme as Equações 4.1 e 4.2.

$$Taxa_1 = \frac{Quantidade\ de\ dados\ enviados\ pelo\ cliente \times 2}{Quantidade\ de\ dados\ recebidos\ pelo\ cliente} \tag{4.1}$$

$$Taxa_2 = \sqrt[2]{Total\ enviado\ pelo\ cliente + 2} \tag{4.2}$$

O menor valor entre $Taxa_1$ e $Taxa_2$ é utilizado como crédito desse usuário. Se o resultado for menor que 1, é utilizado 1, e se for superior a 10, será utilizado 10.

Para evitar fraudes, o *eMule* utiliza um sistema de autenticação com o emprego de chaves pública e privada para a identificação segura dos clientes.

Outro elemento importante é que os créditos dos outros clientes são armazenados localmente. Assim, se o arquivo que contém o registro da quantidade de dados que foram transferidos e recebidos de outros usuários do *eMule* for apagado ou se corromper, todos os créditos serão perdidos.

Compartilhamento de arquivos

Com base em Kulbak e Bickson (2005), o compartilhamento padrão de arquivos utilizado pelo *eMule* pode ser descrito, de forma simplificada, em um cenário fictício, onde um usuário (*Usuário A*) disponibiliza um arquivo para envio (*upload*) e outro usuário (*Usuário B*) solicita o recebimento desse arquivo (*download*):

1. O *Usuário A*, ao se conectar a um servidor do *eMule*, envia a lista de arquivos presentes nas pastas compartilhadas. O servidor armazena essa lista na sua base de dados juntamente com as listas de arquivos disponibilizados pelos outros usuários conectados a ele.

2. O *Usuário B*, ao realizar uma busca por arquivos cujos nomes contêm determinadas palavras, recebe a lista de usuários que estão disponibilizando esses arquivos. De posse dessa lista, o *Usuário B* escolhe o arquivo a ser baixado, realiza uma conexão com os usuários que possuem o arquivo (ou parte(s) do mesmo) e solicita que esse arquivo seja transferido (neste exemplo fictício, o *Usuário A* está disponibilizando um arquivo cujo nome contém o termo pesquisado e que foi escolhido pelo *Usuário B* para ser baixado). Deve-se ressaltar que podem ser utilizadas diversas fontes para o recebimento do arquivo simultaneamente, ou seja, é possível receber partes distintas de um arquivo de diversos usuários ao mesmo tempo.

3. O *Usuário A*, ao receber a solicitação de transferência de um arquivo do *Usuário B*, coloca esse pedido em uma fila de *download*. Se a fila estiver vazia, a transferência pode começar imediatamente. Caso contrário, o *Usuário B* deverá esperar até que os outros usuários que estão na sua frente terminem o *download* solicitado anteriormente ou a pontuação do *Usuário B* seja suficiente para ser atendido.

4. Em virtude da utilização do sistema de créditos, se o *Usuário B* anteriormente havia enviado dados para o *Usuário A*, ele terá créditos em relação a esse usuá-

rio, podendo passar à frente de outros usuários que não possuem créditos com o *Usuário A*. Dessa forma, a vantagem que o usuário tem de disponibilizar (compartilhar) arquivos é a possível diminuição do tempo de recebimento de arquivos, pois iria passar à frente, na fila de *downloads*, de outros usuários que não possuem créditos com o usuário que está fornecendo o arquivo.

5. Enquanto o arquivo está sendo recebido pelo *Usuário B*, ele é armazenado na pasta de Arquivos Temporários do *eMule*, sendo que cada *chunk* completo é compartilhado automaticamente (essa opção não pode ser desabilitada pelo usuário do *eMule*). Quando o arquivo estiver completo, ele será automaticamente movido para a pasta de Arquivos Completos do *eMule* e todos os *chunks* do arquivo serão compartilhados.

A Figura 4.3 apresenta, de forma visual, o processo básico de transferência de arquivos descrito anteriormente. As operações foram numeradas e ordenadas para facilitar sua identificação.

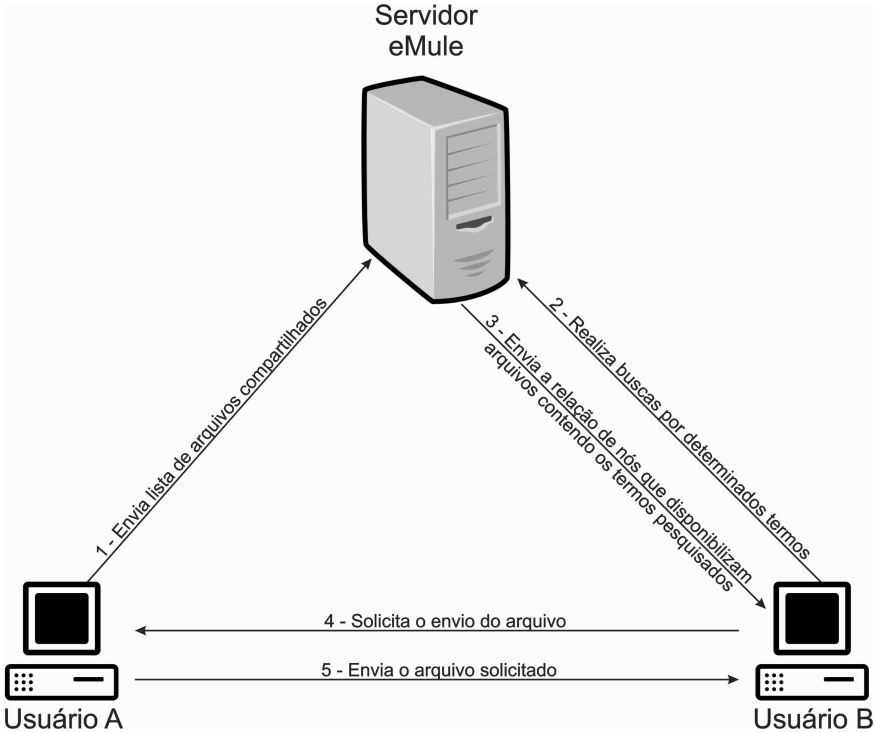
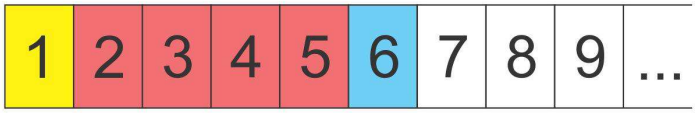


Figura 4.3: Esquema simplificado da busca e a transferência de arquivos realizado pelo eMule (adaptado de Kulbak e Bickson, 2005)

4.4.2 Comunicação entre clientes e entre clientes e servidores do eMule

O *eMule* utiliza, na comunicação onde são empregados pacotes TCP, a estrutura descrita na Figura 4.4. Como é possível observar na referida figura, os seis primeiros bytes dos pacotes TCP do *eMule* possuem posição fixa.

Payload de um pacote TCP do eMule

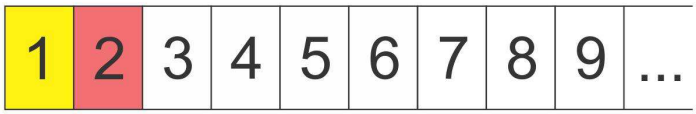


Deslocamento	Descrição
01	Indica o protocolo do eMule: 0xe3 - pacote antigo, baseado no eDonkey 0xc5 - novo pacote, criado pelo eMule 0xe4 - utilizado na rede KAD do eMule 0xd4 - utilizado pelo eMule para indicar que o conteúdo está compactado
02 - 05	Indica o tamanho do pacote do eMule, sem contar o 1º byte (protocolo) e os 4 bytes do tamanho
06	Indica o tipo de operação do pacote

Figura 4.4: Informações presentes nos primeiros bytes de um pacote TCP do eMule

A estrutura dos pacotes UDP do *eMule* é apresentada na Figura 4.5. Somente os dois primeiros bytes possuem posição fixa neste tipo de pacote. De acordo com Kulbak e Bickson (2005), o *eMule* não necessita do UDP para o funcionamento normal, tanto que o usuário do *eMule* pode desabilitar a utilização desse protocolo.

Payload de um pacote UDP do eMule



Deslocamento	Descrição
01	Indica o protocolo do eMule. Sempre é utilizado o código 0xe3
02	Indica o tipo de operação do pacote

Figura 4.5: Informações presentes nos primeiros bytes de um pacote UDP do eMule

Os protocolos que foram criados pelo eDonkey e que foram utilizados pelo *eMule* são apresentados nas Tabelas 6.2 e 6.3, localizadas no Anexo A. Os novos pacotes, que foram criados pelo *eMule* para disponibilizar outras funcionalidades, são apresentados na Tabela 6.4, presente no Anexo A. Os pacotes indicados nessas três tabelas são utilizados na rede eD2k. Os pacotes utilizados na rede KAD são apresentados na Tabela 6.1.

Ressalta-se que é possível que sejam enviados mais de um tipo de pacote do *eMule* em um mesmo pacote TCP. As mensagens são simplesmente concatenadas, mas cada uma delas possui a mesma estrutura apresentada na Figura 4.4, sendo que o tamanho refere-se à mensagem individual, e não ao tamanho total do pacote.

Os pacotes identificados pelo código `0xD4` são compactados com ZLIB para economizar largura de banda.

As comunicações entre os clientes do *eMule* e entre clientes e servidores do *eMule* podem ser divididas de acordo com a rede utilizada: (1) eD2k e (2) KAD.

Rede eD2k

A rede eD2k foi a primeira a ser utilizada pelo *eMule*, tendo sido baseada na rede utilizada pelo aplicativo eDonkey. Essa rede é composta por centenas de servidores e dezenas (ou centenas) de milhões de clientes.

Em redes P2P de arquitetura Parcialmente Centralizada ou Híbrida Descentralizada (como a eD2k), o resultado das consultas é gerenciado pelos servidores. Os clientes enviam as solicitações de buscas por arquivos aos servidores que retornam o conjunto de clientes que estão disponibilizando arquivos contendo os termos solicitados pelos usuários (Allali et al., 2009). Dessa forma, é possível a identificação das buscas, do oferecimento de arquivos e do pedido de mais fontes para baixar arquivos na comunicação entre o cliente e os servidores. A Tabela 6.2 apresenta os pacotes utilizados na comunicação entre os clientes e os servidores do *eMule*.

Em relação à comunicação entre os clientes do *eMule* na rede eD2k, são empregados, além dos pacotes utilizados pelo eDonkey (aplicativo do qual foi originado o *eMule*), novos pacotes de comunicação criados pelo *eMule*. A Tabela 6.3, apresentada no Anexo A, aponta os pacotes enviados entre os clientes do *eMule* que já existiam no *eDonkey* e a Tabela 6.4, também presente no Anexo A, apresenta os novos pacotes criados pelo *eMule*. Nessas comunicações entre clientes é possível encontrar informações referentes à troca de mensagens (*chat*) e o efetivo envio/recebimento de arquivos.

O cliente *eMule* deve se conectar a um servidor para acessar a rede de compartilhamento eD2k. Essa conexão de rede deve permanecer aberta para que seja possível a busca e troca de arquivos.

A Figura 4.6 apresenta o diagrama de alto nível da rede *overlay* eD2k criada pelo *eMule*. Ressaltam-se as conexões entres os nós e entre os servidores.

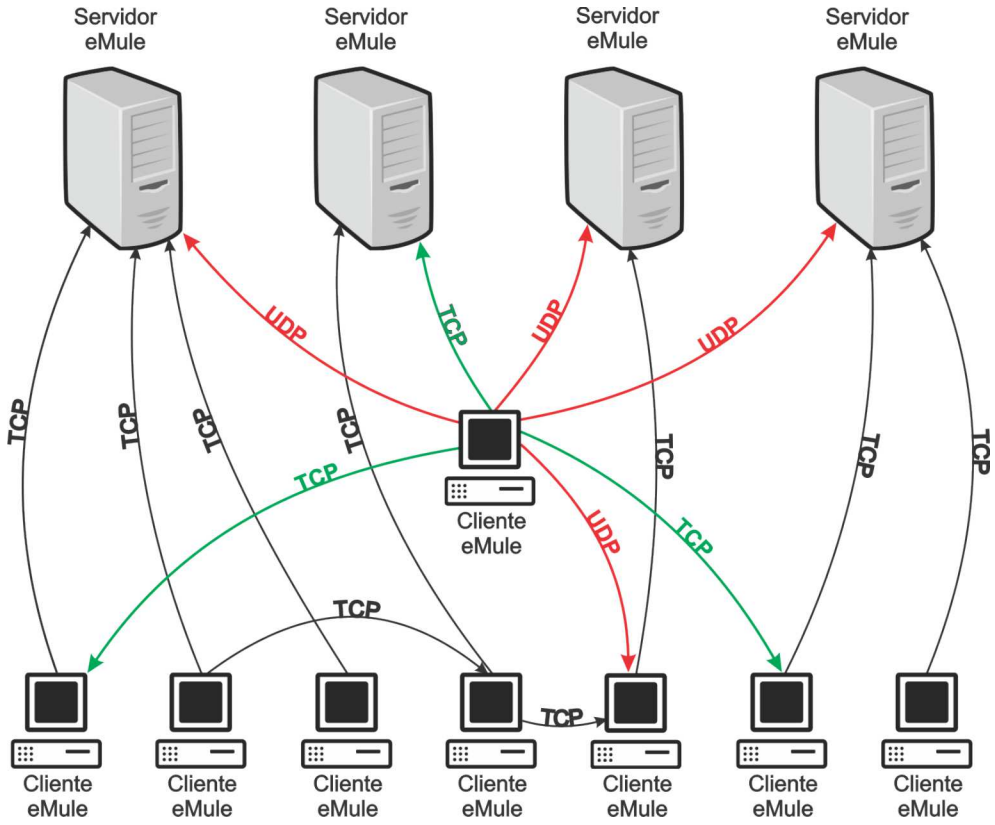


Figura 4.6: Diagrama de alto nível da rede eD2k do eMule, conforme proposto por Kulbak e Bickson (2005)

Rede KAD

A rede Kademlia (KAD) foi criada por Maymounkov e Mazieres (2002) como uma rede de computadores P2P descentralizada. Na rede KAD, cada nó é identificado por um *node ID*, de 128 bits de tamanho que é gerado, de forma aleatória através de uma função criptográfica de *hash*, na primeira vez que o *eMule* é executado, permanecendo o mesmo até a reinstalação do aplicativo ou quando o arquivo de preferências for apagado (Steiner et al., 2007).

A rede KAD utiliza a noção de distância entre dois pontos para fins de roteamento entre os nós. Dados x e y, a distância (d) entre esses dois pontos é o resultado do *ou exclusivo* (XOR) entre os identificadores de nó (*node ID*) de cada um deles: $d(x, y) = x \oplus y$. Como

exemplo, tendo $x = 1011$ e $y = 0111$, a distância entre eles é $d(x, y) = 1011 \oplus 0111 = 1100$. A vantagem é que esse sistema é simétrico, pois $a \oplus b = b \oplus a$, diferentemente de outros sistemas DHT, como o Chord, facilitando a localização de outros pontos (Steiner et al., 2007).

A rede KAD foi criada para diminuir a dependência nos servidores do *eMule*. Por este motivo, é empregada na comunicação entre clientes e, de acordo com Mysicka (2006), o principal protocolo de transporte utilizado é o UDP.

Na Tabela 6.1, presente no Anexo A, são apresentados os principais tipos de pacotes utilizados na comunicação entre clientes na rede KAD. Ressalta-se que a partir da versão atual (0.50a), o conteúdo dos pacotes TCP e UDP utilizados na comunicação nessa rede são cifrados, de forma automática, pelo *eMule*.

Comunicação com relevância pericial

Na Tabela 6.5, localizada no Anexo A, são apresentados os pacotes utilizados na comunicação entre clientes ou entre o cliente e servidores do *eMule*, que apresentam relevância pericial, conforme apresentado na Seção 5.3.2.

Esses pacotes possuem relevância pericial pois apresentam elementos para a comprovação da materialidade e obtenção de indícios de autoria – objetivos do inquérito policial. Além disso, também possuem importância para evidenciar o dolo (consciência do autor na prática delituosa) e existência (ou não) da transnacionalidade do delito, elementos que irão auxiliar no trâmite da persecução penal.

4.4.3 Opções de criptografia no eMule

A partir da versão 0.47b, o *eMule* suporta criptografia do *payload* dos pacotes de rede dos protocolos TCP e UDP utilizados na rede eD2k.

Com a chegada da versão atual (0.50a), o conteúdo dos pacotes TCP e UDP da rede KAD também passaram a ser criptografados, de forma automática.

Embora seja um recurso criado para ocultar a presença do *eMule*, a criptografia não é muito utilizada pelos clientes desse aplicativo, conforme pode-se observar na Tabela 4.5. Esta tabela, que apresenta a utilização de criptografia pelo *eMule*, conforme levantamento apresentado em Ipoque (2009), indica que apenas 16% dos usuários habilitaram a utilização de criptografia na Alemanha e, na região sul da Europa, esse percentual diminui para 7%.

Tabela 4.5: Utilização da criptografia pelo aplicativo eMule de acordo com Ipoque (2009)

Região	**Fluxo cifrado**	**Fluxo não cifrado**
Alemanha	16,08%	83,92%
Região sul da Europa	7,03%	92,97%

O aplicativo *eMule* possui três opções de configuração para utilização de criptografia (em idioma português e inglês) dos arquivos da rede eD2k (a rede KAD é criptografada automaticamente e não pode ser desabilitada a partir da versão 0.50a):

- "Habilitar protocolo de ofuscamento" ("*Enable protocol obfuscation*") - por padrão, esta opção está desligada. Dessa forma, o *eMule* não utiliza o protocolo de ofuscamento (criptografia do conteúdo dos pacotes).

- "Permitir somente conexões ofuscadas (não recomendado)" ("*Allow obfuscated connections only (not recommended)*") - por padrão, esta opção está desligada e somente pode ser habilitada se a opção anterior estiver ligada. Se estiver habilitada, o *eMule* exige que a conexão seja estabelecida com criptografia, rejeitando as conexões não cifradas.

- "Desabilitar suporte para conexões ofuscadas" ("*Disable support for obfuscated connections*") - Desabilita o suporte para conexões ofuscadas. Por padrão, esta opção é desabilitada. Caso habilitada, o *eMule* não responde a pedidos para estabelecimento de conexões que utilizem criptografia.

Embora a rede eD2k do *eMule* não seja criptografada por padrão, resultando na baixa identificação dessa proteção apresentada na Tabela 4.5, é razoável supor que esse índice é maior entre os criminosos que querem ocultar suas atividades ilícitas.

Capítulo 5

Modelo Proposto e Experimentos

O presente capítulo irá apresentar na Seção 5.1 os trabalhos correlatos relacionados à identificação do tráfego P2P com o uso de IA e, na Seção 5.2, o modelo proposto. Na Seção 5.3 é apresentada a forma de implementação da RNA utilizada no presente trabalho para detecção do tráfego criptografado e na Seção 5.4 a relação das heurísticas empregadas para detectar o tráfego não criptografado. Na Seção 5.5.1 são apresentados os experimentos realizados com a utilização da RNA para identificar o tráfego criptografado e na Seção 5.5.2 são apresentados os experimentos com as heurísticas sobre o tráfego não criptografado.

5.1 Trabalhos Correlatos

As técnicas utilizadas para a identificação do fluxo de dados gerado por aplicativos P2P foram apresentadas na Seção 4.3. A Tabela 4.4 compara as características de cada técnica de classificação, apresentando seus pontos fortes e fracos.

A Tabela 5.1 compara os trabalhos que utilizaram técnicas de IA para a identificação do fluxo de dados gerados por aplicativos P2P.

Até onde foi possível investigar, poucos trabalhos utilizaram IA para analisar o *eMule*, como Braga (2007) e Zhang et al. (2010), ou realizaram estudos sobre a identificação do tráfego criptografado de aplicativos P2P, como Bonfiglio et al. (2007). Alguns trabalhos, tais como Bernaille et al. (2006), Erman et al. (2007c) e Li et al. (2007), buscaram identificar o fluxo gerado pelo *eDonkey*, aplicativo a partir do qual originou-se o *eMule*.

Em relação à utilização de outras técnicas para a identificação do fluxo de rede gerado pelo *eMule*, mesmo quando criptografado, podem ser citados os trabalhos de Freire

Tabela 5.1: Operações realizadas no eMule e principais pacotes gerados

Trabalho	Técnica	Aplicativos P2P	Tráfego Identificado	
			Com criptografia	Sem criptografia
Bernaille et al. (2006)	K-médias	*eDonkey* e *Kazaa*	-	*eDonkey* (84,2%) e *Kazaa* (95,24%)
Erman et al. (2007c)	K-médias	*BitTorrent*, *eDonkey*, *Gnutella* e *Kazaa*	-	81,32%
Erman et al. (2006)	AutoClass (EM)	Limewire	-	80%
Zhang et al. (2010)	Árvore de decisão	*Skype*, *PPLive*, *PPStream*, *Bittorrent*, *eMule* e *Thunder*	-	96,7% [1]
Auld et al. (2007)	Rede Bayesiana	*Kazaa*, *BitTorrent* e *Gnutella*	-	97,2%
Park et al. (2008)	Rede Bayesiana e MLP	*BitTorrent*, *Fileguri*, *Soribada* e *Gample*	-	80% a 90%
Moore e Zuev (2005)	*Naïve Bayes*	*Kazaa*, *BitTorrent* e *Gnutella*	-	55,18%
Bonfiglio et al. (2007)	DPI e *Naïve Bayes*	*Skype*	$\approx$ 100%	$\approx$ 100%
Li et al. (2007)	SVM	*MSN*, *Soulseek*, *Skype*, *BitTorrent*, *eDonkey*, *Qq* e *100bao*	-	95,18% [2]
Este et al. (2009)	SVM	*BitTorrent* e *MSN*	-	*MSN* (91,2%) e *BitTorrent* (96,8%)
Chen et al. (2009)	MLP	*BitTorrent* e *QQLive*	-	96.5%
Braga (2007)	MLP	*BitTorrent* e *eMule*	-	85% [3]

[1] Foram classificadas apenas categorias de aplicativos P2P entre si.
[2] Não foram apresentadas informações sobre a configuração de cada aplicativo P2P.
[3] O tráfego dos aplicativos P2P foi comparado apenas com os protocolos HTTP e FTP.

et al. (2009) e de Carvalho (2009). Nesses trabalhos foram empregadas técnicas de detecção de intrusão para a identificação do fluxo de dados, mas não foi apresentada a taxa de sucesso obtida. Além disso, a versão do *eMule* que foi analisada foi a 0.49, anterior à versão atual (0.50a). Na versão 0.49, alguns pacotes utilizados pelo *eMule* ainda não eram criptografados e foram esses pacotes os identificados pela técnica de detecção de intrusão.

Em diversos trabalhos são analisados vários aplicativos P2P simultaneamente, mas não são apresentadas as taxas individuais de identificação. Ainda, poucos trabalhos detalham as configurações dos aplicativos ou a quantidade de pacotes gerados por cada um dos programas P2P do conjunto de treinamento e de testes, dificultando a comparação entre os resultados.

Dentre os trabalhos correlatos, não foi encontrado semelhante ao presente trabalho, no qual são empregadas heurísticas e RNA MLP para a identificação do tráfego não criptografado e criptografado do *eMule*.

5.2 Apresentação do Modelo Proposto

A Figura 5.1 apresenta a arquitetura do modelo proposto, a qual foi baseada no trabalho de Kaushik et al. (2010). Esses autores apresentaram uma arquitetura para a perícia em redes de computadores (*network forensics*) que tem por objetivo auxiliar o perito no processamento e análise de grande volume de dados de rede. Nota-se que a arquitetura possui três módulos distintos: (1) Módulo de Captura; (2) Módulo de Análise; e (3) Módulo de Apresentação.

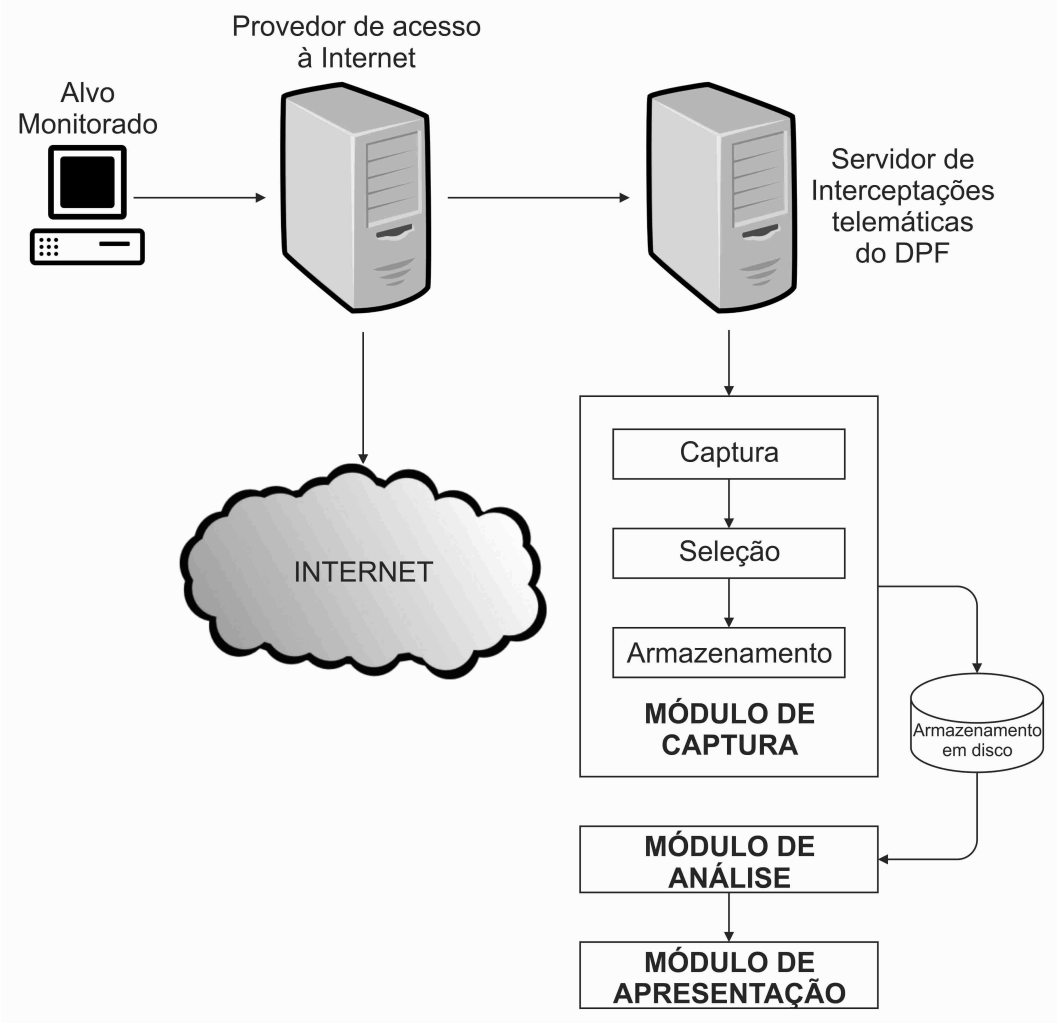


Figura 5.1: Arquitetura do modelo proposto (adaptado de Kaushik et al., 2010)

O Módulo de Captura, desenvolvido anteriormente a este trabalho pela Coordenação de Tecnologia da Informação (CTI) do DPF, se encontra no início de suas operações nesse órgão policial. Este módulo desempenha as tarefas de captura, seleção e armazenamento dos dados referentes às interceptações telemáticas, separando-os por alvo e operação, através do recebimento de uma cópia do fluxo de dados do investigado proveniente das empresas de telecomunicações. Esse módulo não foi desenvolvido especificamente para aplicativos P2P, mas sim para armazenar quaisquer dados obtidos através de interceptações telemáticas tais como navegação em páginas da Internet, acesso a mensagens eletrônicas e conversas instantâneas.

O Módulo de Análise, tema central da presente dissertação, tem como principal tarefa a identificação de informações de cunho pericial dentre os fluxos de dados interceptados que contêm pacotes relacionados ao aplicativo *eMule*. Essas informações podem ser obtidas dos fluxos de dados nos quais não foi empregada criptografia. Caso tenha sido utilizada essa proteção nos dados trafegados, o Módulo de Análise deve ser capaz de identificar se existem informações que foram geradas pelo *eMule*, mesmo que os dados tenham sido criptografados.

Relacionado ao Módulo de Apresentação, foi desenvolvido paralelamente ao presente trabalho uma ferramenta para obtenção, tratamento, importação e análise de dados de interceptações telemáticas denominada *CLIT* (Peron et al., 2011). Essa ferramenta interpreta e apresenta ao usuário informações de alguns protocolos de rede tais como HTTP, SMTP e POP, dentre outros, mas não reconhece o protocolo utilizado pelo *eMule*.

O aplicativo *CLIT* possibilita a integração com módulos externos de reconhecimento de protocolos de aplicação (filtros), entregando ao módulo externo o fluxo de dados da mesma forma que foi entregue à aplicação, remontando pacotes IP fragmentados e reconstruindo as sessões dos pacotes TCP e UDP.

Para apresentar as propostas desta dissertação, foi implementado o protótipo de um filtro para análise do protocolo não criptografado do *eMule*, permitindo a integração com a ferramenta *CLIT* e possibilitando a completude da arquitetura apresentada na Figura 5.1. Esse experimento é apresentado na Seção 5.5.2.

Como exemplos de informações que podem ser obtidas com a utilização do protótipo sobre o fluxo de dados sem criptografia do aplicativo *eMule*, podem-se citar:

1. Reconhecimento de identificadores de arquivos (*File IDs*) que possuam conteúdo ilícito, através da busca em tabelas previamente criadas pelas forças policiais;

2. Identificação de termos suspeitos nos nomes dos arquivos disponibilizados ou transferidos;

3. Reconstrução de arquivos transferidos, permitindo sua visualização e análise pelos operadores do sistema;

4. Gravação dos identificadores (*File ID*) arquivos que estão sendo disponibilizados para que, posteriormente ao cumprimento do mandado de busca e apreensão do disco rígido, seja possível confrontar com o conteúdo dos arquivos, aumentando a materialidade de delitos como, por exemplo, a disponibilização de material envolvendo pornografia infantojuvenil;

5. Identificação de outros usuários que enviam dados para o usuário monitorado e que solicitam dados desse usuário, tendo em vista uma ampliação do processo investigativo, podendo estender-se para outros países;

6. Registro de atividades realizadas de forma manual pelo usuário tais como buscas por arquivos e troca de mensagens (*chat*), possibilitando a determinação de autoria e dolo;

7. Parâmetros de configuração tais como o nome do usuário, endereço IP, servidores utilizados e data e horário da utilização do aplicativo .

Conforme exposto, também é necessário o desenvolvimento de um método para a identificação do fluxo de dados P2P para o Módulo de Análise quando o tráfego estiver criptografado. Como solução, esta dissertação propõe o emprego de aprendizagem supervisionada em uma RNA MLP, conforme apresentado na Seção 4.3.4

Para produzir o grupo de treinamento foram gerados pacotes e mensagens do *eMule* que possuem relevância pericial. Após o treinamento, foram aferidas as informações obtidas (validação) com o emprego da RNA sobre um subconjunto desses dados, apresentado na Seção 5.5.1.

Para obtenção dos dados utilizados para o treinamento e validação da RNA, foram empregados fluxos de dados que foram gerados com o emprego de criptografia e fluxos que foram gerados sem esse tipo de proteção por parte do aplicativo *eMule*, conforme a Seção 5.3.2.

5.3 Caracterização da RNA

Nesta dissertação é utilizada uma RNA MLP para a identificação do tráfego P2P criptografado. A escolha de redes neurais artificiais decorreu de sua capacidade de generalização, classificação e resolução de problemas não-lineares, conforme apresentado na Seção 3.2.

Para a implementação da RNA, foi utilizado o aplicativo MatLab, versão 7.12.0.635 (R2011a). Este programa, criado pela empresa MathWorks Inc., é um ambiente técnico e científico que permite a realização de tarefas que exijam intensa capacidade computacional numérica, trabalhando com dados em um formato organizado em matrizes (MathWorks, 2011b).

5.3.1 Identificação do ambiente

Para obtenção dos dados do tráfego do *eMule*, não foi possível utilizar um servidor real pois, de modo geral, os servidores do *eMule* implementam técnicas para evitar o comportamento anômalo dos clientes, colocando-os em listas negras (*blacklists*), nas quais o referido cliente é proibido de logar-se àquele servidor por determinado tempo. Desta forma foi realizada uma captura dos dados do tráfego do *eMule* conforme segue.

Para geração de diversos pacotes que apresentem relevância pericial devem ser executadas várias ações repetitivas que, sem dúvida, iriam colocar os clientes em *blacklists* nos servidores. Para contornar essa situação, foi criado um servidor *eMule* para fins de geração do tráfego P2P. Foi instalado o aplicativo mais utilizado para funcionar como servidor do *eMule* – o *eServer Lugdunum* [1], em sua última versão (17.5), que possui suporte à criptografia do *payload* dos pacotes. Em relação ao aplicativo utilizado no acesso às redes do *eMule* (eD2k e KAD), foi empregada a última versão desse aplicativo P2P – a 0.50a.

Para a criação de máquinas virtuais nas quais foram executados os clientes e o servidor do *eMule* foi utilizado o aplicativo de virtualização *VMware*, versão 7.0. Para execução dos clientes foi instalado o sistema operacional Microsoft *Windows XP*, atualizado com o *Service Pack 3*, e, para execução do servidor do *eMule*, foi instalado o Linux *Ubuntu* versão 11.4, com Kernel 2.6.38.

[1] A página oficial deste aplicativo (`http://lugdunum2k.free.fr/kiten.html`) não está atualmente disponível.

5.3.2 Coleta dos dados

Inicialmente os clientes do *eMule* foram executados nas máquinas virtuais com a opção de criptografia do conteúdo dos pacotes desabilitada. Essa é a configuração padrão do aplicativo, ou seja, para que seja utilizada a criptografia, é necessário que o usuário do *eMule* altere essa opção, habilitando o suporte à ofuscação de pacotes.

Foram coletadas informações referentes aos pacotes de dados empregados para a comunicação entre clientes e entre clientes e servidores do *eMule*. Para gravação do tráfego de rede gerado foi utilizado o aplicativo *Wireshark*, versão 1.6.0.

Para transferência dos arquivos, foram criados 60 arquivos com conteúdo randômico, sendo 10 arquivos com 1 KB, 10 com 10 KB, 10 com 100 KB, 10 com 1 MB, 10 com 10 MB e 10 com 100 MB, totalizando aproximadamente 1,2 GB.

Foram realizadas diversas ações que apresentam interesse pericial no aplicativo *eMule*. A escolha das ações com interesse pericial foi obtida da seguinte forma:

1. tendo como base o funcionamento de perícia em redes de computadores, apresentado na Seção 2.3;

2. com a análise das características envolvendo crimes cibernéticos, conforme Seção 2.4, principalmente nos delitos de compartilhamento de material pornográfico infantojuvenil, apresentado na Seção 2.4.1, e da atuação das forças policiais, de acordo com a Seção 2.4.2;

3. de acordo com os aspectos legais da atuação da perícia, apresentados na Seção 2.5, e do funcionamento da persecução penal no Brasil, de acordo com a Seção 2.6; e

4. com da análise do funcionamento do aplicativo *eMule*, apresentado na Seção 4.4.

A Tabela 5.2 apresenta as operações realizadas, o número de vezes que essas operações foram executadas e os principais pacotes que foram gerados. Para que essa tabela não ficasse excessivamente grande, o critério de inserção foi que a quantidade de pacotes gerados tenha sido superior a 100. Durante a captura dos dados, que teve duração aproximada de 20 horas, foram encontrados 93,75% de todos os tipos de pacotes utilizados na comunicação do aplicativo *eMule*.

Em seguida, foi habilitada a opção de utilização de criptografia nos clientes do *eMule*. Foram realizadas as mesmas operações elencadas na Tabela 5.2 mas, em decorrência do conteúdo dos pacotes estar cifrado, não foi possível determinar, com exatidão, os tipos de pacotes gravados nem sua quantidade. Entretanto, tendo em vista que a

Tabela 5.2: Operações realizadas no eMule e principais pacotes gerados

Operação	**Nº de execuções**	**Pacotes gerados**	**Quantidade**
Login no servidor (sem arquivos compartilhados)	1000	Hello	2000
		Hello answer	1000
		ID change	1000
		Server message	1000
		Server status	1000
Login no servidor (com arquivos compartilhados)	1000	Get list of servers	1000
		Hello	2000
		Hello answer	1000
		ID change	1000
		Offer files	1000
		Server identification	1000
		Server message	1000
		Server status	1000
Envio de mensagens	1000	Chat message	1000
Buscas por arquivos	1000	Search request	1000
		Search result	1000
Transferência de arquivos entre clientes	Arquivos transferidos: -10 com 1KB; -10 com 10KB; -10 com 100KB; -10 com 1MB; -10 com 10MB; -10 com 100MB. Total: 1,2 GB.	Accept upload request	101
		Cancel transfer	104
		Chat message	115
		Found sources	743
		Get sources	1245
		Hello	125
		Hello answer	116
		Offer files	134
		Request file parts	1350
		Sending file part	22572
		Server status	272
		Start upload request	101

documentação oficial do *eMule* informa que a habilitação da criptografia somente traz como *overhead* na rede o procedimento de troca da chave de sessão, é razoável supor que foram gerados os mesmos pacotes que os criados sem a utilização de criptografia.

Para filtrar apenas os pacotes que foram gerados pelo *eMule* com a utilização de criptografia, foi utilizada a filtragem por porta de comunicação, conforme Seção 4.3.1. A utilização dessa técnica é possível tendo em vista que existe total conhecimento sobre a configuração dos clientes e do servidor do *eMule* neste ambiente de coleta de dados.

5.3.3 Armazenamento dos dados coletados

O tráfego de rede gerado foi armazenado em diversos arquivos gravados no formato PCAP (*Packet CAPture*) pelo aplicativo Wireshark. Foram gravados aproximadamente 7 GB de dados, contendo mais de quarenta mil pacotes relacionados ao *eMule*.

Esses dados gravados foram separados em dois grupos: (1) sem utilização de criptografia, cujos pacotes gerados em maior número são apresentados na Tabela 5.2; e (2) com a utilização de criptografia. Conforme exposto anteriormente, não foi possível identificar, com base no conteúdo dos pacotes, de acordo com a Seção 4.3.2, os tipos de pacotes gerados nem a sua quantidade após a utilização de criptografia do conteúdo dos pacotes pelo *eMule*.

Foram gravadas todas as informações dos pacotes, inclusive o *payload* completo, para análise do conteúdo desses pacotes conforme descrito na Seção 4.3.2. Esse enfoque também é diferente da maioria dos trabalhos de monitoramento de tráfego P2P correlatos, onde questões sobre a privacidade dos usuários eram relevantes. No caso das investigações policiais, existe uma prévia autorização judicial para a interceptação telemática, inexistindo problemas relacionados à privacidade.

5.3.4 Filtragem dos dados para treinamento e validação

Durante a gravação dos dados de treinamento da RNA pelo *Wireshark*, apenas o aplicativo *eMule* estava em execução na máquina virtual. Entretanto, o sistema operacional utilizado (Microsoft *Windows XP*) executa diversos aplicativos em segundo plano (*background*). Dessa forma, além dos pacotes gerados pelo *eMule* também foram capturadas informações de rede provenientes de outros aplicativos que estavam sendo executados em *background*.

Dentre os dados gravados pelo aplicativo *Wireshark*, foram retirados os pacotes que não foram gerados pelo *eMule* ou que não apresentavam relevância. Dentre os pacotes retirados da amostra para treinamento e validação, encontram-se os pacotes de estabelecimento (*three-way handshake*) e encerramento da conexão TCP, pacotes TCP ou UDP sem dados (*payload* vazio), comunicação NetBIOS do sistema operacional Microsoft *Windows XP*, pacotes ARP e DHCP, dentre outros.

5.3.5 Seleção dos atributos da RNA

Para a identificação do fluxo de dados pela RNA, é necessária a criação de uma relação de atributos que serão analisados pela rede neural. Os atributos que serão analisados devem ser transformados em números para fins de treinamento, teste e validação da RNA.

Os atributos foram obtidos a partir de informações do fluxo de dados que foram gravados em formato PCAP. Para o cálculo desses atributos, foi utilizada a ferramenta *Tcptrace* [2] versão 6.6.0. Essa ferramenta foi desenvolvida por Shawn Ostermann, da Universidade de Ohio, para a análise de arquivos contendo fluxos de dados TCP gerados por diversas ferramentas como, por exemplo, *tcpdump*, *Windump* e *Wireshark*. São apresentados 138 atributos contendo informações relacionadas a cada conexão, como o tempo de conexão, número de bytes e pacotes que foram enviados e recebidos, retransmissões, RTT (*round trip times*), dentre outras informações (Tcptrace, 2011).

Diversos trabalhos utilizam vários atributos para identificação de fluxos de dados. No trabalho de Moore et al. (2005), embora tenham sido calculados 248 atributos de fluxos TCP, foram utilizados apenas 11 para a classificação dos fluxos.

Outros trabalhos utilizam um subconjunto ainda menor de atributos com boa taxa de identificação e com diminuição do custo computacional necessário para o processamento dos dados. Dentre esses trabalhos pode-se citar Erman et al. (2006), com cinco atributos; Erman et al. (2007d) com três grupos de atributos; Zhang et al. (2010) com três atributos e Bernaille et al. (2006) com atributos baseados apenas no tamanho dos pacotes. Dessa forma, mesmo com pequenos grupos de atributos é possível obter um alto índice de acerto na identificação do fluxo de dados.

Neste trabalho foram utilizados atributos relacionados ao tipo do protocolo de transporte e ao tamanho, bem como à quantidade de pacotes do fluxo de dados. Também é utilizada a contagem de pacotes com o *flag PUSH*. Esse *flag* é utilizado nos pacotes TCP para indicar que os dados presentes no *buffer* devem ser enviados para a aplicação que está recebendo as informações da rede. A relação dos atributos escolhidos neste trabalho é apresentada na Tabela 5.3.

A escolha desses atributos teve como base os trabalhos de Moore e Zuev (2005), Bonfiglio et al. (2007), Li et al. (2007) e Este et al. (2009). Foram analisadas as características com maior relevância nesses trabalhos e escolhidos empiricamente para a identificação dos fluxos. A combinação desses atributos, bem como a utilização de subconjuntos ou a inclusão de outros atributos poderia melhorar o desempenho da RNA. Essas modifi-

[2] Disponível em: `http://www.tcptrace.org/index.html`

Tabela 5.3: Atributos do fluxo de dados utilizados para treinamento, teste e validação da RNA

Atributo	Observação
Protocolo de transporte	Indica o número do protocolo de transporte utilizado pelo protocolo IP [1]
Tamanho do primeiro pacote	É calculado o tamanho total do primeiro pacote (cabeçalho e *payload*) após o estabelecimento da conexão TCP
Média de tamanho dos pacotes	É calculada a média do tamanho dos pacotes, excluindo-se os de estabelecimento de conexão TCP (*SYN*, *SYN+ACK*, *ACK*)
Quantidade de pacotes do fluxo	Também são excluídos os pacotes de estabelecimento de conexão TCP
Tamanho mínimo dos pacotes	É calculado o tamanho mínimo dos pacotes naquele fluxo
Pacotes com o *flag* PUSH	Número de pacotes nos quais o *flag* PUSH está setado

[1] O código do protocolo TCP é 6 e do protocolo UDP é 17. O número do protocolo de transporte utilizado pelo IP é atribuído pela IANA (disponível em: `www.iana.org/assignments/protocol-numbers/protocol-numbers.xml`).

cações no conjunto de atributos da RNA é uma das sugestões de trabalhos futuros da presente dissertação.

5.3.6 Parâmetros da RNA

Esta seção irá apresentar os parâmetros que foram utilizados para a criação da RNA. De acordo com Haykin (1998), quanto menor a taxa de aprendizagem, mais suave será a trajetória de correção dos pesos pelo algoritmo de treinamento. Entretanto, o custo será uma aprendizagem lenta. Por outro lado, se a taxa de aprendizagem for alta, as mudanças nos pesos podem tornar a rede instável. Dessa forma, foi escolhido empiricamente um valor inicial baixo (0,01) para a taxa de aprendizagem, por não existirem restrições quanto ao tempo de aprendizado da RNA.

A medida de desempenho utilizada foi o soma dos erros quadráticos (*Sum of squared error* - SSE), uma forma muito utilizada para a busca pela otimização dos pesos das conexões (Russel e Norvig, 2004). Nesta função de desempenho, o erro é caracterizado pelo somatório dos módulos das diferenças entre as saídas desejadas e as saídas obti-

das com o classificador, elevado ao quadrado. Essa função pode ser escrita conforme apresentado na Equação 5.1:

$$SSE = \sum_{j=1}^{n}(|Saída\ esperada_1 - Saída\ obtida_1| + |Saída\ esperada_2 - Saída\ obtida_2|)^2 \tag{5.1}$$

Como função de ativação foi utilizada a função sigmóide (*logsig*), conforme apresentado na Seção 3.2.2, Figura 3.4 (c) e Equação 3.4. Essa função, de acordo com Haykin (1998), é a função de ativação mais utilizada em RNA e exibe um comportamento balanceado entre linear e não-linear e é continuamente diferenciável. Esta função foi utilizada em todas as camadas da rede.

Foi escolhida uma topologia de RNA com duas camadas ocultas pois, de acordo com Russel e Norvig (2004), é possível a representação de funções contínuas ou descontínuas, com o número de neurônios crescendo exponencialmente com o número de entradas. Como exemplo, esses autores afirmam ser necessária a utilização de $\frac{2^n}{n}$ unidades ocultas para a codificação de todas as funções booleanas de n entradas. Com essa configuração de camadas ocultas, pode ser resolvida grande parte dos problemas de reconhecimento de padrões por RNAs.

A função de treinamento *trainlm* do MatLab utiliza o algoritmo Levenberg-Marquardt e foi escolhida por apresentar a maior velocidade de treinamento para redes de alimentação direta (*feedforward*) com até algumas centenas de pesos e, como aspecto negativo, possui o maior uso de memória (MathWorks, 2011a). A escolha dessa função de treinamento deveu-se à possibilidade do treinamento ser realizado em computadores com maior capacidade em termos de memória, não havendo impacto negativo relevante neste aspecto.

A função de aprendizagem tem por finalidade o ajuste dos pesos das sinapses da rede neural para que as saídas obtidas se aproximem das saídas desejadas. Foi utilizado o valor *learngdm*, que utiliza um gradiente descendente com momentum para esta finalidade (MathWorks, 2011a).

A rede de alimentação direta pode ser criada pelo comando *nntool* do MatLab, que disponibiliza uma interface gráfica para a configuração da rede neural. Conforme o exposto nesta seção, a rede neural foi criada com os seguintes parâmetros:

- Função de treinamento: *trainlm*
- Taxa de aprendizagem: 0,01
- Função de aprendizagem adaptativa: *learngdm*

- Medida de desempenho: SSE (*Sum of squared error*)
- Número de camadas ocultas: 2
- Função de ativação: sigmóide (*logsig*)
- Número de épocas: 1000
- Saída: P2P ou não-P2P

Em relação ao número de neurônios nas camadas ocultas, foram realizados experimentos para tentar identificar o número mais adequado. Esses experimentos são apresentados na Seção 5.5.1.

5.3.7 Normalização dos dados para treinamento

Para que a Rede Neural não atribua elevada importância a determinados atributos em detrimento de outros, os dados devem ser normalizados para que fiquem dentro da mesma faixa de valores. Foi utilizada a faixa de 1000 elementos para que cada valor fique entre 0 e 1.

Para normalização dos dados, foram identificados os valores máximo e mínimo para cada atributo e foi aplicada a Equação 5.2.

$$Atributo\ normalizado = \frac{Valor\ do\ atributo}{Valor\ máximo - Valor\ mínimo} \tag{5.2}$$

Dessa forma, foram obtidos os valores normalizados para cada atributo, os quais foram empregados no treinamento da RNA.

5.3.8 Processo de treinamento da RNA

Para treinamento da RNA, os fluxos de dados foram copiados para um arquivo que, por sua vez, foi dividido em três conjuntos, definidos empiricamente. Os nomes dos subconjuntos e o percentual de pacotes do arquivo original são apresentados na Tabela 5.4.

Como saída esperada, são criadas duas classes: (1) Tráfego P2P; e (2) Tráfego não-P2P. Dessa forma, a se o fluxo pertencer ao *eMule*, a saída da rede neural deve ser 1 para a classe Tráfego P2P e 0 para a classe Tráfego não-P2P. Caso o fluxo não estiver relacionado ao *eMule*, a saída deverá ser 0 e 1, respectivamente, para as classes Tráfego

Tabela 5.4: Divisão dos fluxos de dados

Conjunto	**Percentual dos fluxos**
Treinamento	70 %
Teste	15 %
Validação	15 %

P2P e Tráfego não-P2P. Se o resultado for 0 e 0 ou 1 e 1 para as classes Tráfego P2P e Tráfego não-P2P, esse fluxo será considerado não classificado.

5.3.9 Validação da RNA

Para validação do desempenho da RNA, foram utilizadas as métricas completude (*completeness*) e acurácia (*accuracy*). A completude de um sistema de identificação de fluxo de dados, de acordo com Karagiannis et al. (2005), mede o percentual do tráfego que foi classificado, ou seja, é a razão entre o número de fluxos classificados sobre o número total dos fluxos (Equação 5.3). Segundo esses autores, a acurácia indica o percentual do tráfego classificado que foi corretamente identificado. Em outras palavras, a acurácia indica a probabilidade que um fluxo classificado pertença à classe identificada pela RNA (Equação 5.4).

$$Completude = \frac{N^o \ de \ fluxos \ classificados}{N^o \ total \ de \ fluxos} \tag{5.3}$$

$$Acurácia = \frac{N^o \ de \ fluxos \ classificados \ corretamente}{N^o \ total \ de \ fluxos} \tag{5.4}$$

Como medida de desempenho, é apresentada a matriz de confusão, que contabiliza os erros e acertos da RNA MLP de classificação. Essa matriz é apresentada na Tabela 5.5.

Tabela 5.5: Matriz de confusão do resultado da classificação dos fluxos

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	Verdadeiro P2P (VP)	Falso não-P2P (FN)	Sem classificação em alguma classe
	Não-P2P	Falso P2P (FP)	Verdadeiro não-P2P (VN)	Sem classificação em alguma classe

Um fluxo é considerado não classificado quando apresenta 1 para as classes P2P e Não-P2P ou quando apresenta 0 para ambas classes.

Na Seção 5.5 são apresentados os experimentos realizados e os respectivos resultados obtidos são apresentados no Capítulo 6.

5.4 Heurísticas

A identificação do tráfego não criptografado do *eMule* foi realizada com base nas heurísticas apresentadas na Tabela 5.6. Essas heurísticas, baseadas no trabalho de Karagiannis et al. (2003); Sen et al. (2004); Freire et al. (2009); de Carvalho (2009); Feng (2010), foram expandidas, com a inserção de identificação das operações do eMule (conforme Seção 4.4.2), cálculo do tamanho dos pacotes com múltiplas mensagens e dos pacotes compactados, além da criação de novas regras para a identificação dos pacotes UDP.

As heurísticas foram obtidas através da verificação e estudo do funcionamento do aplicativo *eMule*, conforme a Seção 4.4, e da realização de testes experimentais com a análise dos pacotes de rede gerados pelo *eMule*, apresentados nas Tabelas 6.1 a 6.4 do Anexo A.

A utilização das heurísticas descritas permitiu identificar 100% do tráfego do *eMule* do conjunto de testes que não se encontra criptografado, conforme apresentado na Seção 5.5.2.

5.5 Experimentos Realizados

Para a realização dos experimentos foi gerado um conjunto próprio de aferição de desempenho, composto por fluxos de rede gerados pelo compartilhamento de arquivos (*download* e *upload*) com usuários reais do *eMule*. Não foi possível a utilização de dados reais obtidos através de interceptação telemática, pois o DPF não possui ferramenta adequada para a obtenção de informações do *eMule*. Dessa forma, a técnica de interceptação telemática não é, normalmente, utilizada durante a investigação desse tipo de delito.

O *eMule* foi configurado para permitir somente conexões ofuscadas, conforme descrito na Seção 4.4.3, para gerar pacotes criptografados. Foi escolhido um *software* livre para transferência com o objetivo de evitar problemas relacionados à infração de direitos autorais. Durante a coleta dos dados, além do *eMule* foram executados aplicativos que utilizam os protocolos HTTP, FTP e POP, comumente encontrados em monitoramen-

Tabela 5.6: Heurísticas utilizadas para identificação do tráfego não criptografado do *eMule*

Protocolo	Localização	Valor	Observação
TCP	1º byte (protocolo P2P)	O primeiro byte do *payload* é `0xE3`, `0xE4`, `0xD4` ou `0xE4`	Essas são as assinaturas dos procolos do eMule em pacotes TCP: `0xE3` (eDonkey), `0xE4` (eMule), `0xD4` (conteúdo compactado) e `0xE4` (KAD)
	2º ao 5º byte (tamanho)	O tamanho é igual ao tamanho do *payload*	É possível o envio de diferentes pacotes do eMule em um mesmo pacote TCP. Neste caso, é verificado o somatório dos tamanhos das mensagens do eMule presentes nesse pacote
	6º byte (operação)	Indica o código da operação do eMule	As operações são códigos que indicam a função do pacote TCP do eMule. As relações das operações foram apresentadas no Anexo A. Se os dados estiverem comprimidos (opção `0xD4`), os dados são descomprimidos antes da verificação da operação
UDP	1º byte (protocolo P2P)	O primeiro byte do *payload* é `0xE3`	Essa é a assinatura do eMule em pacotes UDP: `0xE3` (eDonkey)
	2º byte (operação)	Indica o código da operação do eMule	As operações são códigos que indicam a função do pacote UDP do eMule. As relações das operações são apresentadas no Anexo A

tos por interceptação telemática. Informações sobre os dados obtidos são apresentados na Tabela 5.7.

Como é possível observar, a maior parte do tráfego capturado é de pacotes nos quais foi utilizado o procolo TCP ou UDP na camada de transporte. Serão escolhidos, dentre esse conjunto de dados, 10000 fluxos para a realização dos experimentos apresentados na Seção 5.5.1, sendo 5000 fluxos pertencentes ao *eMule* e 5000 pertencentes a outros aplicativos. O critério de classificação foi a porta utilizada pelo *eMule* (Seção 4.3.1), ressaltando-se que essa classificação é válida tendo em vista o controle sobre a configuração desse aplicativo P2P durante a geração dos fluxos para aferição de desempenho. Em computadores de suspeitos monitorados em uma interceptação telemática, esse conhecimento sobre a porta utilizada pelo *eMule* não existe, impossibilitando a utilização da classificação baseada em portas nesses casos.

Tabela 5.7: Informações sobre os dados capturados de clientes do eMule

Informações	**Quantidade**
Número total de pacotes	2.580.466
Número de pacotes TCP ou UDP	2.579.320
Tamanho total dos pacotes (cabeçalho e *payload*)	2,28 GB
Tempo de duração da captura	6 horas

5.5.1 Experimento 1 - Fluxos criptografados

Após o treinamento da RNA com os parâmetros detalhados na Seção 5.3, foram realizados experimentos com o número de neurônios nas camadas ocultas para identificar o número que obtivesse o melhor desempenho da RNA. O número de neurônios testados e definidos empiricamente foi de 3, 5, 10, 20, 40 e 100. Os resultados, relativos ao conjunto de validação (conforme Seção 5.3.8) dos fluxos selecionados, são apresentados nas Tabelas 5.8 a 5.13.

Tabela 5.8: Matriz de confusão do resultado da classificação dos fluxos com 3 neurônios nas camadas ocultas

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	3873 (77,46%)	795 (15,90%)	332 (6,64%)
	Não-P2P	843 (16,86%)	3779 (75,58%)	378 (7,56%)

Com a utilização de 3 neurônios nas camadas ocultas foram obtidas as taxa de identificação mais baixas: 77,46% na classe P2P e 75,58% na classe Não-P2P. Também foi obtida a maior quantidade de fluxos não classificados (6,64% na classe P2P e 7,56% na classe Não-P2P). A acurácia medida foi de 76,52% e a completude foi de 92,90%.

Tabela 5.9: Matriz de confusão do resultado da classificação dos fluxos com 5 neurônios nas camadas ocultas

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	4272 (85,44%)	554 (11,08%)	174 (3,48%)
	Não-P2P	792 (15,84%)	4061 (81,22%)	147 (2,94%)

Com a utilização de 5 neurônios nas camadas ocultas foram obtidas as taxa de identificação de 85,44% na classe P2P e 81,22% na classe Não-P2P. Também foi obtida, nos fluxos não classificados, o percentual de 3,48% na classe P2P e 2,94% na classe Não-P2P. A acurácia medida foi de 83,33% e a completude foi de 96,79%.

Tabela 5.10: Matriz de confusão do resultado da classificação dos fluxos com 10 neurônios nas camadas ocultas

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	4278 (85,56%)	562 (11,24%)	160 (3,20%)
	Não-P2P	786 (15,72%)	4086 (81,72%)	128 (2,56%)

A mudança para 10 neurônios nas camadas ocultas aumentou a taxa de identificação em relação à utilização de 3 e 5 neurônios. Passou de 85,44% na classe P2P para 85,56% e, em relação à classes Não-P2P, passou de 81,22% para 81,72%. Além disso, a quantidade de tráfego não classificado diminuiu na classe P2P de 3,48% para 3,20% e de 2,94% para 2,56% na classe Não-P2P. A utilização de 10 neurônios nas camadas ocultas apresentou uma melhora na acurácia, passando de 83,33% para 83,64% e a completude passou de 96,79% para 97,12%.

Tabela 5.11: Matriz de confusão do resultado da classificação dos fluxos com 20 neurônios nas camadas ocultas

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	4282 (85,64%)	562 (11,24%)	156 (3,12%)
	Não-P2P	762 (15,24%)	4117 (82,34%)	121 (2,42%)

A utilização de 20 neurônios nas camadas ocultas da RNA apresentou melhoras em relação à utilização de 10 neurônios. A taxa de identificação na classe P2P foi de 85,64% e na classe Não-P2P foi de 82,34%. Não foram classificados corretamente 3,12% dos fluxos da classe P2P e 2,42% na classe Não-P2P. A acurácia passou de 83,64% para 83,99% e a completude aumentou de 97,12% para 97,23%.

Tabela 5.12: Matriz de confusão do resultado da classificação dos fluxos com 40 neurônios nas camadas ocultas

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	4336 (86,72%)	528 (10,56%)	131 (2,72%)
	Não-P2P	620 (12,40%)	4267 (85,34%)	113 (2,26%)

A RNA na qual foram utilizados 40 neurônios nas camadas ocultas apresentou o melhor desempenho. Foram identificados corretamente 86,72% dos fluxos da classe P2P, indicando um aumento de 1,08% em relação ao uso de 20 neurônios, e 85,34% da classe Não-P2P, indicando um aumento de 3% em relação à utilização de 20 neurônios. A acurácia, quando comparada com a RNA que utilizou 20 neurônios, passou de 83,99% para 86,03% e a completude passou de 97,23% para 97,51%.

Tabela 5.13: Matriz de confusão do resultado da classificação dos fluxos com 100 neurônios nas camadas ocultas

		Classe prevista		
		P2P	**Não-P2P**	**Não classificado**
Classe real	**P2P**	4292 (85,84%)	569 (11,38%)	139 (2,78%)
	Não-P2P	771 (15,42%)	4095 (81,90%)	134 (2,68%)

A utilização de 100 neurônios nas camadas ocultas apresentou um desempenho inferior ao da RNA na qual foram empregados 40 neurônios nas camadas ocultas. A taxa de identificação na classe P2P diminuiu de 86,72% para 85,84% e na classe Não-P2P passou de 85,34% para 81,64%. Essa configuração de RNA apresentou acurácia de 83,87% e completude de 97,27%.

Tabela 5.14: Resultados dos experimentos com os números de neurônios nas camadas ocultas

Nº de neurônios	**Nº de fluxos**	**Acurácia**	**Completude**
3 neurônios	7652	76,52%	92,90%
5 neurônios	8333	83,33%	96,79%
10 neurônios	8364	83,64%	97,12%
20 neurônios	8399	83,99%	97,23%
40 neurônios	8603	86,03%	97,51%
100 neurônios	8374	83,87%	97,27%

O comparativo de desempenho entre essas abordagens nas quais ocorre a variação do número de neurônios nas camadas ocultas é apresentada na Tabela 5.14. Com base nos dados apresentados nessa Tabela, o número de neurônios que obteve o melhor desempenho foi o de 40 neurônios, tanto na acurácia como na completude, conforme as Equações 5.3 e 5.4 da Seção 5.3.9.

Nesta RNA, a completude (percentual do tráfego que foi classificado), ou seja, para os quais foi atribuída uma classe, foi de 97,51% com acurácia (percentual do tráfego classificado que foi corretamente identificado) de 86,03%. Levando em conta que o objetivo é o de identificar a utilização do *eMule*, mesmo que o tráfego esteja criptografado por este aplicativo P2P para fins de validação das informações prestadas pelo ISP, o resultado foi considerado satisfatório.

5.5.2 Experimento 2 - Fluxos não-criptografados

Para a realização do experimento com a utilização das heurísticas apresentadas na Tabela 5.6 da Seção 5.4, foi desenvolvido um protótipo integrado à ferramenta CLIT.

Essa ferramenta, descrita na Seção 5.2, permite a utilização de módulos externos, denominados filtros, para a interpretação de protocolos de aplicação.

Foi utilizado o procedimento semelhante ao apresentado na Seção 5.5 para obtenção dos dados de rede, mas neste experimento o *eMule* foi configurado para desabilitar o suporte para conexões ofuscadas (conforme apresentado na Seção 4.4.3). Também foram executados aplicativos que utilizam os protocolos HTTP, FTP e POP durante a captura, para que fossem criados fluxos de dados normalmente encontrados em informações provenientes de interceptações telemáticas. Informações sobre os dados capturados são apresentados na Tabela 5.15.

Tabela 5.15: Informações sobre os dados capturados de clientes do eMule sem utilização de criptografia

Informações	**Quantidade**
Número total de pacotes	824.560
Número de pacotes TCP ou UDP	824.453
Tamanho total dos pacotes (cabeçalho e *payload*)	811 MB
Tempo de duração da captura	41 minutos

Tendo em vista que, normalmente, em cada fluxo de dados são trocados diversos pacotes, foram retirados apenas 200 fluxos dos dados capturados, sendo 100 fluxos do *eMule* e 100 fluxos que não são relacionados a este aplicativo.

Após a seleção dos fluxos de rede, esses dados foram processados pela ferramenta CLIT e pelo protótipo desenvolvido. Foram identificados todos os 100 fluxos do *eMule* com as heurísticas propostas. Para exemplificar os dados encontrados com o protótipo proposto, são apresentadas as principais informações obtidas nas Figuras 5.2 a 5.6. Para preservar a privacidade dos usuários, as referências aos endereços IP e nomes e identificadores (*User ID*) de usuários serão apagadas.

Na Figura 5.2 pode-se observar um pacote de conexão (*Hello*), seguido por um de solicitação do recebimento do arquivo cujo *File ID* é `F9E204DA56268FA7212CCBB957B35EDF` (*Start upload request*) e por uma solicitação de três fragmentos (*Request file parts*), no deslocamento 148684800 a 149237760, totalizando 552.960 bytes

Na Figura 5.3 é apresentado um pedido de login (*Hello*) o qual foi aceito pelo servidor do *eMule*. O cliente também solicita o recebimento da lista de servidores (*Get list of servers*) e que sejam fornecidas mais fontes (*Get sources*) para o arquivo cujo *File ID* é `F9E204DA56268FA7212CCBB957B35EDF`. Por fim, o cliente disponibiliza um arquivo incompleto, cujo *File ID* é `F9E204DA56268FA7212CCBB957B35EDF`, para que seja baixado

FLUXO ENVIADO	
PACOTE	Hello
CÓDIGO DO PACOTE	0x01
DESCRIÇÃO DO PACOTE	HELLO (CONEXÃO COM OUTRO USUÁRIO DO EMULE)
INFORMAÇÕES DO USUÁRIO QUE INICIOU A CONEXÃO	
HASH DO USUÁRIO	[redacted]
ID DO USUÁRIO	[redacted]
IP DO USUÁRIO	[redacted]
PORTA DO USUÁRIO	10000
INFORMAÇÕES DO USUÁRIO/SERVIDOR	
NOME	[DreaMule]dreamule.org
VERSÃO	60
IP DO SERVIDOR	[redacted]
PORTA DO SERVIDOR	1
PACOTE	Start upload request
CÓDIGO DO PACOTE	0x54
DESCRIÇÃO DO PACOTE	OP_STARTUPLOADREQ (PEDIDO PARA RECEBIMENTO DE ARQUIVO)
HASH DO ARQUIVO SOLICITADO	F9E204DA56268FA7212CCBB957B35EDF
PACOTE	Request file parts
CÓDIGO DO PACOTE	0x47
DESCRIÇÃO DO PACOTE	OP_REQUESTPARTS (PEDIDO DE TRANSFERÊNCIA DE ARQUIVOS)
HASH DO ARQUIVO	F9E204DA56268FA7212CCBB957B35EDF
PEDIDO 01	DESLOCAMENTO DE 148684800 A 148869120
TAMANHO:	184320 BYTES
PEDIDO 02	DESLOCAMENTO DE 148869120 A 149053440
TAMANHO:	184320 BYTES
PEDIDO 03	DESLOCAMENTO DE 149053440 A 149237760
TAMANHO:	184320 BYTES
PACOTE	Request file parts

Figura 5.2: Contato com outro usuário do *eMule* e solicitação de *upload*

por outros clientes do *eMule* (*Offer files*). Ressalta-se que o arquivo sendo disponibilizado possui o mesmo *File ID* do arquivo para o qual estão sendo solicitadas mais fontes, indicando que esse arquivo ainda está na fila de recebimento. Caso esse arquivo esteja na lista de arquivos cujos *hashes* (*File ID*) sejam reconhecidamente ilícitos como no caso de pornografia infantojuvenil, pode ser comprovada a materialidade do delito de disponibilização de material contendo exploração sexual de crianças ou adolescentes.

A Figura 5.4 apresenta as informações dos pacotes que indicam que um cliente do *eMule* aceitou enviar um arquivo (*Accept upload request*) e inicia a transferência de três trechos do arquivo cujo *File ID* é `F9E204DA56268FA7212CCBB957B35EDF`. O primeiro fragmento vai do deslocamento 154214400 até 154219520, o segundo vai de 154219520 até 154224640 e o terceiro vai de 154224640 até 154229760, totalizando 15360 bytes transferidos. Os fragmentos transferidos entre clientes do *eMule* podem ser armazenados no disco rígido para que se possa reconstruir os arquivos, possibilitando uma análise de seu conteúdo. Também podem ser registrados previamente em uma tabela os identificadores de arquivos (*File ID*) de arquivos que comprovadamente são ilícitos como, por exemplo, de material que envolve pornografia infantojuvenil. Dessa forma,

FLUXO ENVIADO	
PACOTE	Hello
CÓDIGO DO PACOTE	0x01
DESCRIÇÃO DO PACOTE	OP_LOGINREQUEST (LOGIN EM UM SERVIDOR DO EMULE)
INFORMAÇÕES DO USUÁRIO QUE INICIOU A CONEXÃO	
HASH DO USUÁRIO	[redacted]
ID DO USUÁRIO	[redacted]
IP DO USUÁRIO	[redacted]
PORTA DO USUÁRIO	10000
NOME DO USUÁRIO	[DreaMule]dreamule.org
VERSÃO	60
FLAGS	281
VERSÃO DO EMULE	51200
PACOTE	Get list of servers
CÓDIGO DO PACOTE	0x14
DESCRIÇÃO DO PACOTE	OP_GETSERVERLIST (PEDIDO PARA RECEBIMENTO DA LISTA DE OUTROS SERVIDORES DO EMULE)
PACOTE	Get sources
CÓDIGO DO PACOTE	0x19
DESCRIÇÃO DO PACOTE	OP_GETSOURCES (SOLICITA FONTES DE UM DETERMINADO ARQUIVO)
HASH DO ARQUIVO	F9E204DA56268FA7212CCBB957B35EDF
Nº DE FONTES DO ARQUIVOS	0
PACOTE	Offer files
CÓDIGO DO PACOTE	0x15
DESCRIÇÃO DO PACOTE	OFFERFILES (DISPONIBILIZAÇAO DE ARQUIVOS PARA OUTROS CLIENTES DO EMULE)
Nº DE ARQUIVOS OFERECIDOS	1
ARQUIVO 1/1	
HASH	F9E204DA56268FA7212CCBB957B35EDF
TIPO DE COMPARTILHAMENTO	ARQUIVO INCOMPLETO SENDO DISPONIBILIZADO
NOME DO ARQUIVO	ubuntu-11.04-desktop-i386.iso
TAMANHO DO ARQUIVO	718583808 bytes

Figura 5.3: Login do usuário do *eMule*, solicitação de fontes de um arquivo e disponibilização de um arquivo incompleto

a materialidade do delito de transmissão desse tipo de material estaria comprovada, inclusive quanto à transnacionalidade ou não, já que os endereços IP do remetente e do destinatário são conhecidos.

A Figura 5.5 apresenta o pacote de login em um servidor *eMule* (*Hello*), seguido por duas buscas por arquivos, sendo a primeira pelo termo "ubuntu" e a segunda pelo termo "linux". Poderia ser utilizada uma tabela que contivesse termos comumente encontrado em materiais ilícitos, auxiliando o trabalho do policial responsável pela análise do fluxo de dados. Esse fluxo poderia indicar a autoria, já que as buscas são realizadas de forma manual pelo usuário, indicando a data e horário em que o usuário está na frente do computador, facilitando a identificação do criminoso principalmente se o computador é compartilhado por várias pessoas em uma mesma residência. Também

PACOTE	Accept upload request
CÓDIGO DO PACOTE	0x55
DESCRIÇÃO DO PACOTE	OP_ACCEPTUPLOADREQ (PEDIDO ACEITO PARA O ENVIO DE ARQUIVO)
PACOTE	Sending file part
CÓDIGO DO PACOTE	0x46
DESCRIÇÃO DO PACOTE	OP_SENDINGPART (TRANSFERÊNCIA DE ARQUIVOS)
HASH DO ARQUIVO	F9E204DA56268FA7212CCBB957B35EDF
DESLOCAMENTO INICIAL NO ARQUIVO	154214400
DESLOCAMENTO FINAL NO ARQUIVO	154219520
QUANTIDADE DE DADOS TRANSFERIDOS	5120 BYTES
PACOTE	Sending file part
CÓDIGO DO PACOTE	0x46
DESCRIÇÃO DO PACOTE	OP_SENDINGPART (TRANSFERÊNCIA DE ARQUIVOS)
HASH DO ARQUIVO	F9E204DA56268FA7212CCBB957B35EDF
DESLOCAMENTO INICIAL NO ARQUIVO	154219520
DESLOCAMENTO FINAL NO ARQUIVO	154224640
QUANTIDADE DE DADOS TRANSFERIDOS	5120 BYTES
PACOTE	Sending file part
CÓDIGO DO PACOTE	0x46
DESCRIÇÃO DO PACOTE	OP_SENDINGPART (TRANSFERÊNCIA DE ARQUIVOS)
HASH DO ARQUIVO	F9E204DA56268FA7212CCBB957B35EDF
DESLOCAMENTO INICIAL NO ARQUIVO	154224640
DESLOCAMENTO FINAL NO ARQUIVO	154229760
QUANTIDADE DE DADOS TRANSFERIDOS	5120 BYTES

Figura 5.4: Informa que o pedido de *upload* foi aceito e inicia a transferência do arquivo

pode ser utilizado para a comprovação do dolo no recebimento de arquivos ilícitos, pois os termos utilizados na busca podem demonstrar a intenção do criminoso em receber determinado tipo de material.

A Figura 5.6 apresenta o pacote com uma mensagem entre dois usuários do *eMule* (*Chat message*). Da mesma forma que a busca apresentada na Figura 5.5, a troca de mensagens é realizada de forma manual pelo usuário do *eMule*, indicando a presença do usuário no computador cujo fluxo está sendo interceptado, auxiliando na determinação de autoria. Além disso, dependendo do teor das mensagens, também pode-se determinar o dolo na troca de material ilícito.

A análise dos pacotes do aplicativo *eMule* pode comprovar a autoria, materialidade e dolo de crimes envolvendo o compartilhamento de arquivos através desse aplicativo P2P. A integração com a ferramenta CLIT facilita a atuação do investigador, pois na mesma ferramenta é possível a análise de protocolos de outros aplicativos da Internet

FLUXO ENVIADO

PACOTE	Hello
CÓDIGO DO PACOTE	0x01
DESCRIÇÃO DO PACOTE	OP_LOGINREQUEST (LOGIN EM UM SERVIDOR DO EMULE)
INFORMAÇÕES DO USUÁRIO QUE INICIOU A CONEXÃO	
HASH DO USUÁRIO	[illegible]
ID DO USUÁRIO	[illegible]
IP DO USUÁRIO	[illegible]
PORTA DO USUÁRIO	62041
NOME DO USUÁRIO	http://emule-project.net
VERSÃO	60
FLAGS	281
VERSÃO DO EMULE	51200
PACOTE	Search request
CÓDIGO DO PACOTE	0x01
DESCRIÇÃO DO PACOTE	OP_SEARCHREQUEST (BUSCAS POR ARQUIVOS)
TIPO DA BUSCA	PELO NOME DO ARQUIVO
TERMO PESQUISADO	ubuntu
PACOTE	Search request
CÓDIGO DO PACOTE	0x01
DESCRIÇÃO DO PACOTE	OP_SEARCHREQUEST (BUSCAS POR ARQUIVOS)
TIPO DA BUSCA	PELO NOME DO ARQUIVO
TERMO PESQUISADO	linux

Figura 5.5: Realização de buscas por arquivos cujos nomes contêm determinadas palavras chave

FLUXO ENVIADO

PACOTE	Chat message
CÓDIGO DO PACOTE	0x4E
DESCRIÇÃO DO PACOTE	OP_MESSAGE (MENSAGEM DE CHAT)
MENSAGEM DE CHAT	teste2

Figura 5.6: Realização de troca de mensagens entre usuários do *eMule*

como HTTP, FTP e POP, além do *eMule*. Além disso, as heurísticas utilizadas permitiram a identificação de 100% do conjunto de testes, resultado considerado satisfatório.

Capítulo 6

Análise dos Resultados e Conclusões

Neste trabalho foi projetada uma RNA para classificação do tráfego criptografado das redes eD2k e KAD, utilizadas pelo aplicativo *eMule*, conforme Seção 5.5.1. Também foi projetado um conjunto de heurísticas para a identificação do tráfego no qual não foi empregada criptografia e desenvolvido um protótipo para aplicar esse conjunto, apresentado na Seção 5.5.2.

Nos fluxos de dados em que não houve a utilização de criptografia, o sistema obteve informações com relevância pericial, conforme apresentado na Seção 5.5.2, que podem possibilitar a comprovação da autoria (Figuras 5.5 e 5.6), materialidade (Figuras 5.3 e 5.4), intenção do agente (dolo) (Figuras 5.5 e 5.6) e a delimitação geográfica do delito (Figuras 5.3 e 5.4). Essa comprovação pode permitir um trâmite mais célere do inquérito policial e do processo penal, haja vista a produção de provas em um momento anterior ao da execução do mandado de busca e apreensão.

Ainda, a obtenção dessas informações pode ser crucial para a persecução penal, já que o criminoso pode utilizar sistemas de criptografia no disco rígido, dificultando ou, até mesmo, inviabilizando a perícia no computador apreendido, resultando na impunidade do criminoso.

Caso seja utilizada criptografia no tráfego de rede, a RNA pode confirmar a utilização do aplicativo P2P *eMule* em um determinado local, aumentando a confiabilidade dos dados obtidos dos ISPs. A confirmação é importante, haja vista a ocorrência de erros na informação do endereço do suspeito que utilizou determinado endereço IP por parte dos ISPs. Caso não seja detectada a utilização do *eMule*, seriam necessárias maiores investigações em relação à efetiva utilização (ou não) desse aplicativo no endereço for-

necido pelo ISP, podendo inocentar pessoas que não estavam utilizando esse aplicativo e que haviam sido vítimas de informações incorretas do ISP.

Neste trabalho foi apresentado um conjunto de heurísticas conforme a Tabela 5.6 para identificação de pacotes não criptografados do *eMule*, conseguindo como resultado 100% de identificação no conjunto de testes apresentados na Tabela 5.15.

Para o tráfego criptografado do *eMule*, foi utilizada uma RNA MLP que identificou 86,03% de acurácia com 40 neurônios nas camadas ocultas, conforme Tabela 5.14.

Esses resultados apontam que o sistema apresentado pode auxiliar o processo de persecução penal, possibilitando a identificação de criminosos, a obtenção de elementos para comprovação da materialidade delituosa e a proteção a inocentes que tiveram seu endereço informado erroneamente pelos ISPs.

Para comparar o desempenho entre o presente trabalho e outros sistemas correlatos de classificação de fluxo de rede, o ideal seria a utilização do mesmo conjunto de dados e verificação do desempenho de cada sistema. Entretanto, o armazenamento de pacotes de rede, inclusive com seu conteúdo, é uma solução utópica, tendo em vista os problemas concernentes à privacidade dos usuários (Salgarelli et al., 2007).

Foram encontrados alguns conjuntos de arquivos contendo capturas de fluxos de rede como, por exemplo, os arquivos disponibilizados pelo *Mawi Working Group* [1]. Entretanto, o conteúdo dos pacotes (*payload*) foi suprimido, impedindo, dessa forma, a comparação com outros estudos que utilizaram esses dados.

Além disso, para fins realísticos de comparação, esse conjunto de dados deveria apresentar fluxos de dados gerados pela última versão do *eMule* (0.50a), lançada em abril de 2010, tendo em vista as modificações nos protocolos de comunicação e a utilização de criptografia dos pacotes. Essa necessidade de contemporaneidade dos fluxos de dados é outro fator limitador a sua utilização para comparação entre sistemas de classificação.

Até onde foi possível investigar, poucos trabalhos analisam a detecção do tráfego do *eMule* em que é utilizada a criptografia. Dentre esses trabalhos, é possível citar Freire et al. (2009) e de Carvalho (2009), nos quais foram apresentadas adaptações de regras de detecção de intrusão para a identificação do fluxo de dados do *eMule*, mesmo que o conteúdo estivesse criptografado. Entretanto, esses trabalhos foram realizados sobre a versão 0.49 do *eMule*, quando o protocolo UDP da rede KAD ainda não era cifrado. A detecção ocorreu justamente sobre esse protocolo fato que, com a versão atual do *eMule* (0.50a), não seria mais factível.

[1] Disponível em: `http://mawi.wide.ad.jp/mawi/`

6.1 Trabalhos Futuros

Eventuais alterações promovidas por subsequentes versões do *eMule* podem ser implementadas no sistema criado, tendo em vista a disponibilidade de acesso ao código-fonte do *eMule*.

Também deixa-se como sugestão de trabalhos futuros a complementação da análise do fluxo de dados por ferramentas que possam obter, de forma automática, as informações encontradas no disco rígido apreendido e que possam corroborar os dados obtidos no fluxo de rede. Um passo nesse sentido foi a apresentação do artigo referente à identificação de artefatos periciais do *eMule*, no *6th International Conference on Forensic Computer Science*(ICoFCS) (Lange e Ralha, 2011).

Uma possibilidade para obtenção de maiores evidências periciais seria o desenvolvimento de estudos para decifrar os dados criptografados pelo *eMule* após a apreensão do computador do suspeito. O desenvolvimento é factível, tendo em vista que o *eMule* armazena, no arquivo CRIPTKEY.DAT, a chave pública e privada do usuário.

Podem ser testados outros algoritmos de IA, como *Bayesian Networks* ou árvores de decisão (*decision trees*) ou outros algoritmos de classificação. Além disso, pode-se ampliar ou modificar os atributos selecionados para a identificação do fluxo de dados com o objetivo de melhorar os resultados obtidos com a RNA MLP definida.

Com os resultados apresentados no presente trabalho, é possível sua adaptação para outros aplicativos P2P como, por exemplo, o *BitTorrent* e o *LimeWire*, com as respectivas modificações nos módulos de Análise e Apresentação conforme apresentado na Figura 5.1 (pág. 82). Também podem ser incluídas as particularidades das *mods* (modificações) do aplicativo *eMule* original e que são populares no Brasil como, por exemplo, o *DreaMule* e o *aMule*.

Essas sugestões demonstram a potencialidade do uso de técnicas de IA na área forense, permitindo maior celeridade e obtenção de provas robustas com vistas à combater a criminalidade, através do exemplo de punição do criminoso.

Referências Bibliográficas

AccessData (2011). Forensic Toolkit (FTK) - versão 3. Disponível em: `http://accessdata.com/products/computer-forensics/ftk`. Acessado em: 15 set. 2011.

Allali, O.; Latapy, M. e Magnien, C. (2009). Measurement of edonkey activity with distributed honeypots. In *IPDPS '09 Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, pages 1–9.

Androutsellis-Theotokis, S. e Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371.

Arnold, R. e Miklos, P. (2010). Character recognition using neural networks. In *Computational Intelligence and Informatics (CINTI), 2010 11th IEEE International Symposium on*, pages 311–314.

Auld, T.; Moore, A. e Gull, S. (2007). Bayesian neural networks for internet traffic classification. *Neural Networks, IEEE Transactions on*, 18(1):223–239.

Azzouna, N. e Guillemin, F. (2004). Impact of peer-to-peer applications on wide area network traffic: an experimental approach. *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, pages 1544–1548.

Baryamureeba, V. e Tushabe, F. (2004). The enhanced digital investigation process model. In *Proceedings of the 4th Annual Digital Forensic Research Workshop, Baltimore, MD*, pages 1–9.

Beebe, N. L. (2009). Digital forensic research: The good, the bad and the unaddressed. *Advances in Digital Forensics V*, 306:17–36.

Beebe, N. L. e Clark, J. (2005). Dealing with terabyte data sets in digital investigations. *Advances in Digital Forensics*, pages 3–16.

Bernaille, L.; Teixeira, R.; Akodkenou, I.; Soule, A. e Salamatian, K. (2006). Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23.

Bessadok, F.; Bessaoud, K.; Latapy, M. e Magnien, C. (2009). Measurement of paedophile activity in eDonkey using a client sending queries. In *Advances in the Analysis of Online Paedophile Activity*, pages 91–92.

Bleul, H.; Rathgeb, E. e Zilling, S. (2006a). Advanced P2P multiprotocol traffic analysis based on application level signature detection. In *Telecommunications Network Strategy and Planning Symposium, 2006. NETWORKS 2006. 12th International*, pages 1–6.

Bleul, H.; Rathgeb, E. e Zilling, S. (2006b). Evaluation of an efficient measurement concept for P2P multiprotocol traffic analysis. In *Software Engineering and Advanced Applications, 2006. SEAA'06. 32nd IEEE EUROMICRO Conference on*, pages 414–423.

Bolla, R.; Canini, M.; Rapuzzi, R. e Sciuto, M. (2008). On the Double-Faced Nature of P2P Traffic. *16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008)*, pages 524–530.

Bonfiglio, D.; Mellia, M.; Meo, M.; Rossi, D. e Tofanelli, P. (2007). Revealing skype traffic: when randomness plays with you. In *ACM SIGCOMM Computer Communication Review*, pages 37–48.

Braga, R. A. d. S. (2007). Reconhecimento de Tráfego peer-to-peer utilizando redes neurais. Dissertação de Mestrado, Universidade Federal de Itajubá.

Brasil (1940). Decreto Lei n. 2848 – Código Penal. Disponível em: `http://www.planalto.gov.br/CCIVIL/Decreto-Lei/Del2848.htm`. Acessado em: 22 fev. 2011.

Brasil (1941). Decreto Lei n. 3689 – Código de Processo Penal. Disponível em: `http://www.planalto.gov.br/ccivil_03/decreto-lei/Del3689Compilado.htm`. Acessado em: 22 fev. 2011.

Brasil (1973). Lei n. 5869 – Código de Processo Civil. Disponível em: `http://www.planalto.gov.br/ccivil_03/leis/L5869.htm`. Acessado em: 22 fev. 2011.

Brasil (1988). Constituição da República Federativa do Brasil de 1988. Disponível em: `http://www.planalto.gov.br/ccivil_03/constituicao/constitui%C3%A7ao.htm`. Acessado em: 22 fev. 2011.

Brasil (1990). Lei n. 8069/90 – Estatuto da Criança e do Adolescente (ECA). Disponível em: `http://www.planalto.gov.br/ccivil_03/leis/L8069.htm`. Acessado em: 26 fev. 2011.

Brasil (2004). Superior Tribunal de Justiça (5^a Turma). Recurso Especial n^o 617.221/RJ. Relator: Ministro GILSON DIPP. Brasília, DF. Julgado em: 19 out. 2004.

Brasil (2010). Programa nacional de Direitos Humanos (PNDH-3) / Secretaria de Direitos Humanos da Presidência da República - rev. e atual. Disponível em: `http://http://portal.mj.gov.br/sedh/pndh3/pndh3.pdf`. Acessado em: 29 set. 2011.

Brownlee, N. e Claffy, K. (2002). Understanding Internet traffic streams: Dragonflies and tortoises. *Communications Magazine, IEEE*, 40(10):110–117.

Bryant, R. P., editor (2008). *Investigating Digital Crime*. John Wiley & Sons, Ltd.

Buchanan, B. (2005). A (very) brief history of artificial intelligence. *AI Magazine*, 26(4):53.

Buford, J.; Yu, H. e Lua, E. (2009). *P2P networking and applications*. Morgan Kaufmann, Burlington, MA.

Burji, S.; Liszka, K. e Chan, C. (2010). Malware analysis using reverse engineering and data mining tools. In *System Science and Engineering (ICSSE), 2010 International Conference on*, pages 619–624.

Calazans, J. (2008). Pesquisa IBOPE/Netrating sobre navegação na Internet em setembro de 2008. Mensagem eletrônica enviada em 22 de junho de 2009.

Callado, A.; Kamienski, C.; Szabo, G.; Gero, B.; Kelner, J.; Fernandes, S. e Sadok, D. (2009). A survey on internet traffic identification. *Communications Surveys & Tutorials, IEEE*, 11(3):37–52.

Caloyannides, M. A. (2001). *Computer Forensics & Privacy (Artech House Computer Security Series)*. Artech House Publishers.

Carrier, B. (2005). *File System Forensic Analysis*. Addison-Wesley Professional, 1st edition.

Carrier, B. e Spafford, E. (2003). Getting physical with the digital investigation process. *International Journal of Digital Evidence*, 2(2):1–20.

Casey, E. (2009). *Handbook of Digital Forensics and Investigation.* Academic Press, 1st edition.

Caviglione, L. e Davoli, F. (2008). Traffic volume analysis of a nation-wide eMule community. *Computer Communications*, 31(10):2485–2495.

Chen, H.; Hu, Z.; Ye, Z. e Liu, W. (2009). Research of P2P Traffic Identification Based on Neural Network. In *2009 International Symposium on Computer Network and Multimedia Technology*, pages 1–4.

Chen, Z.; Wang, H.; Peng, L.; Yang, B. e Chen, Y. (2006). A Novel Method of P2P Hosts Detection Based on Flexible Neural Tree. In *Sixth International Conference on Intelligent Systems Design and Applications*, pages 556–561.

Cheng, W.; Gong, J. e Ding, W. (2008). Identifying file-sharing P2P traffic based on traffic characteristics. *The Journal of China Universities of Posts and Telecommunications*, 15(4):112–120.

Choo, K.-K. R. (2008). Organised crime groups in cyberspace: a typology. *Trends in Organized Crime*, 11(3):270–295.

Cohen, F. (2010). Toward a Science of Digital Forensic Evidence Examination. In *Advances in Digital Forensics VI*, pages 17–35. Springer.

Crotti, M.; Dusi, M.; Gringoli, F. e Salgarelli, L. (2007). Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5.

Crowcroft, J.; Pias, M.; Sharma, R. e Lim, S. (2004). A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93.

Daeid, N. N. e Houck, M. (2010). *Interpol's forensic science review.* Taylor & Francis.

Dai, L.; Yang, J. e Lin, L. (2010). A comprehensive system for P2P classification. In *Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on*, pages 561–563.

Dale, W. M. e Becker, W. S. (2007). *The Crime Scene: How Forensic Science Works.* Kaplan Publishing, New York, New York, USA.

Dalpian, G. M. e Benites, C. A. A. (2007). Ferramenta Para Monitoramento de Redes P2P-EspiaMule. In *The Second International Conference of Forensic Computer Science*, pages 70–72.

DARPA (2011). Defense Advanced Research Projects Agency. Disponível em: `http://www.darpa.mil/`. Acessado em: 08 ago. 2011.

de Carvalho, D. A. M. (2009). *Towards the Detection of Encrypted Peer-to-Peer File Sharing Traffic and Peer-to-Peer TV Traffic Using Deep Packet Inspection Methods*. Master of science, University of Beira Interior.

de Oliveira, J. R. S. e da Silva, E. E. (2009). EspiaMule e Wyoming ToolKit: Ferramentas de Repressão à Exploração Sexual Infanto-Juvenil em Redes Peer-to-Peer. In *The Fourth International Conference on Forensic Computer Science*, pages 108–113, Natal, Rio Grande do Norte, Brasil.

Erman, J.; Mahanti, A. e Arlitt, M. (2006). Internet Traffic Identification using Machine Learning. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1–6, San Francisco, CA.

Erman, J.; Mahanti, A. e Arlitt, M. (2007a). Byte me: a case for byte accuracy in traffic classification. In *Proceedings of the 3rd annual ACM workshop on Mining network data*, pages 35–38.

Erman, J.; Mahanti, A.; Arlitt, M.; Cohen, I. e Williamson, C. (2007b). Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, 64(9-12):1194–1213.

Erman, J.; Mahanti, A.; Arlitt, M. e Williamson, C. (2007c). Identifying and discriminating between web and peer-to-peer traffic in the network core. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*, page 883, New York, New York, USA.

Erman, J.; Mahanti, A.; Arlitt, M. e Williamson, C. (2007d). Identifying and discriminating between web and peer-to-peer traffic in the network core. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*, page 883, New York, New York, USA. ACM Press.

Este, A.; Gringoli, F. e Salgarelli, L. (2009). Support Vector Machines for TCP traffic classification. *Computer Networks*, 53(14):2476–2490.

Fafinski, S. (2008). UK Cybercrime Report. Technical report, Garlik.

Fagundes, P. (2009). Fighting Internet Child Pornography - The Brazilian Experience. *The Police Chief*, LXXVI(9):48–55.

Farmer, D. e Venema, W. (1999). Murder on the Internet Express.

Feng, J. (2010). Research on the technology of peer-to-peer traffic classification. In *Computer Communication Control and Automation (3CA), 2010 International Symposium on*, volume 1, pages 491–494.

Finkelhor, D.; Mitchell, K. e Wolak, J. (2006). Online Victimization of Youth: Five Years Later. Technical report, U.S. Department of Justice.

Freire, M. M.; Carvalho, D. a. e Pereira, M. (2009). Detection of Encrypted Traffic in eDonkey Network through Application Signatures. *2009 First International Conference on Advances in P2P Systems*, pages 174–179.

Gerber, A.; Houle, J.; Nguyen, H.; Roughan, M. e Sen, S. (2003). P2P, the gorilla in the cable. In *National Cable & Telecommunications Association (NCTA) 2003 National Show*, pages 8–11.

Ghiassi, M. e Burnley, C. (2010). Measuring effectiveness of a dynamic artificial neural network algorithm for classification problems. *Expert Systems with Applications*, 37(4):3118–3128.

Gorge, M. (2007). Lawful interception: key concepts, actors, trends and best practice considerations. *Computer Fraud & Security*, 2007(9):10–14.

Grobler, C. e Louwrens, B. (2006). Digital Forensics: A Multi-Dimensional Discipline. In *Proceedings of the ISSA 2006 from Insight to Foresight Conference. Pretoria: University of Pretoria.*

Guidance (2011). EnCase Forensic - versão 7. Disponível em: `http://www.guidancesoftware.com/forensic.htm`. Acessado em: 15 set. 2011.

Haffner, P.; Sen, S.; Spatscheck, O. e Wang, D. (2005). ACAS: automated construction of application signatures. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 197–202.

Hankins, R.; Uehara, T. e Liu, J. (2009). A Comparative Study of Forensic Science and Computer Forensics. *2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement*, pages 230–239.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation (2nd Edition).* Prentice Hall.

Hebb, D. O. (1949). *The Organization of Behaviour.* John Wiley & Sons Inc.

Heiser, J. e Kruse, W. (2002). *Computer Forensics: Incident Response Essentials.* Addison-Wesley, Indianapolis, IN, United States, 1st edition.

Hoelz, B. W. P. (2009). *MADIK: Uma Abordagem Multiagente para o Exame Pericial de Sistemas Computacionais*. Dissertação de mestrado em informática, Universidade de Brasília (UnB).

Hoelz, B. W. P.; Ralha, C. G. e Geeverghese, R. (2009). Artificial intelligence applied to computer forensics. *Proceedings of the 2009 ACM symposium on Applied Computing - SAC '09*, page 883.

Houghton Mifflin Company (2001). *The American heritage dictionary of the English language*. Houghton Mifflin Company, 4.ed. edition.

Hu, Y.; Chiu, D.-M. e Lui, J. C. (2009). Profiling and identification of P2P traffic. *Computer Networks*, 53(6):849–863.

Huebner, E.; Bem, D. e Bem, O. (2003). Computer Forensics - Past, Present And Future. *Information Security Technical Report*, 8(2):32–36.

Hughes, D.; Walkerdine, J.; Coulson, G. e Gibson, S. (2006). Peer-to-Peer: Is Deviant Behavior the Norm on P2P File-Sharing Networks? *IEEE Distributed Systems Online*, 7(2):1–1.

Hunton, P. (2009). The growing phenomenon of crime and the internet: A cybercrime execution and analysis model. *Computer Law & Security Review*, 25(6):528–535.

Hurley, J.; Garcia-Palacios, E. e Sezer, S. (2011). Classifying network protocols: a two-way flow approach. *IET Communications*, 5(1):79–89.

Iacovazzi, A. e Baiocchi, A. (2010). Optimum packet length masking. In *Teletraffic Congress (ITC), 2010 22nd International*, pages 1–8.

Ieong, R. S. C. (2006). FORZA - Digital forensics investigation framework that incorporate legal issues. *Digital Investigation*, 3:29–36.

Inman, K. e Rudin, N. (2000). *Principles and Practice of Criminalistics: The Profession of Forensic Science (Protocols in Forensic Science)*. CRC Press.

Internet World Stats (2011). Internet usage and population statistics. Disponível em: `http://www.internetworldstats.com/stats.htm`. Acessado em: 15 set. 2011.

Interpol (2009). Crimes Against Children. Technical report, INTERPOL.

Ipoque (2009). Internet Study 2008/2009. Disponível em: `http://www.ipoque.de/userfiles/file/ipoque-Internet-Study-08-09.pdf`. Acessado em: 22 jul. 2011.

John, W.; Tafvelin, S. e Olovsson, T. (2010). Passive internet measurement: Overview and guidelines based on experiences. *Computer Communications*, 33(5):533–550.

Kala, R.; Shulkla, A. e Tiwari, R. (2009). Fuzzy Neuro Systems for Machine Learning for Large Data Sets. In *2009 IEEE International Advance Computing Conference*, pages 541–545.

Karagiannis, T.; Broido, A.; Brownlee, N.; Claffy, K. e Faloutsos, M. (2003). File-sharing in the Internet: A characterization of P2P traffic in the backbone. *University of California, Riverside, USA, Tech. Rep.*

Karagiannis, T.; Broido, A.; Brownlee, N.; Claffy, K. e Faloutsos, M. (2004). Is P2P dying or just hiding? In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, volume 3, pages 1532–1538.

Karagiannis, T.; Papagiannaki, K. e Faloutsos, M. (2005). BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM Computer Communication Review*, 35(4):229–240.

Kaushik, A. K.; Pilli, E. S. e Joshi, R. (2010). Network forensic system for port scanning attack. *2010 IEEE 2nd International Advance Computing Conference (IACC)*, pages 310–315.

Keralapura, R.; Nucci, A. e Chuah, C.-N. (2009). Self-Learning Peer-to-Peer Traffic Classifier. In *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, pages 1–8.

Keralapura, R.; Nucci, A. e Chuah, C.-N. (2010). A novel self-learning architecture for P2P traffic classification in high speed networks. *Computer Networks*, 54(7):1055–1068.

Kulbak, Y. e Bickson, D. (2005). The eMule protocol specification. Technical report, Hebrew University of Jerusalem, Jerusalem.

Lange, R. e Ralha, C. G. (2011). Identificação de Artefatos Periciais do eMule. In *The Sixth International Conference on Forensic Computer Science (ICoFCS 2011)*, pages 43–53, Florianópolis - Brasil. DOI: 10.5769/C2011004.

Latapy, M.; Magnien, C. e Fournier, R. (2011). Quantifying paedophile activity in a large P2P system. In *IEEE Infocom Mini-Conference*, pages 1–24. IEEE.

Leiner, B. M.; Cerf, V. G.; Clark, D. D.; Kahn, R. E.; Kleinrock, L.; Lynch, D. C.; Postel, J.; Roberts, L. G. e Wolff, S. (2009). A brief history of the internet. *ACM SIGCOMM Computer Communication Review*, 39(5):22.

Li, Y. e Gruenbacher, D. (2010). Analysis of P2P file sharing network's credit system for fairness management. In *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pages 88–95.

Li, Z.; Yuan, R. e Guan, X. (2007). Accurate classification of the internet traffic based on the SVM method. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 1373–1378.

Liang, J. e Kumar, R. (2005). The kazaa overlay: A measurement study. *Computer Networks Journal (Elsevier)*, pages 1–25.

Lopes Júnior, A. (2006). *Sistemas de investigação preliminar no processo penal*. Lumen Juris, Rio de Janeiro, 4. ed. rev edition.

Lungarella, M.; Iida, F.; Bongard, J. e Pfeifer, R. (2007). AI in the 21 st Century - With Historical Reflections. *50 years of artificial intelligence*, pages 1–8.

Malin, C. H.; Casey, E. e Aquilina, J. M. (2008). *Malware Forensics: Investigating and Analyzing Malicious Code*. Syngress.

Marshall, A. M. (2008). *Digital Forensics: Digital Evidence in Criminal Investigations*. John Wiley & Sons, Ltd, 1st edition.

MathWorks (2011a). MatLab R2011a Documentation - Neural Network Toolbox. Disponível em: `http://www.mathworks.com/help/toolbox/nnet/ug/bss330n-1.html`. Acessado em: 22 jul. 2011.

MathWorks (2011b). Matlab, versão 7.12.0.635 (R2011a). Disponível em: `http://www.mathworks.com/products/matlab/`. Acessado em: 22 jul. 2011.

Maymounkov, P. e Mazieres, D. (2002). Kademlia: A peer-to-peer information system based on the XOR metric. *Peer-to-Peer Systems*, pages 53–65.

McCulloch, W. e Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133.

McGregor, A.; Hall, M.; Lorier, P. e Brunskill, J. (2004). Flow clustering using machine learning techniques. In *Passive and Active Network Measurement*, pages 205–214. Springer Berlin / Heidelberg.

McKemmish, R. (1999). What is forensic computing. *Trends and issues in crime and criminal justice*, 118(118).

Minsky, M. e Papert, S. (1969). *Perceptrons*, volume 1988. MIT press Cambridge, MA, 1st edition.

Mohanty, S. e Bhattacharya, S. (2008). Recognition of voice signals for Oriya language using wavelet neural network. *ACM International Journal of Expert Systems with Applications*, 34(3):2130–2147.

Moore, A. e Zuev, D. (2005). Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 50–60.

Moore, A.; Zuev, D. e Crogan, M. (2005). Discriminators for use in flow-based classification. *Technical report, University of Cambridge, Computer Laboratory*, (August).

Mysicka, D. (2006). *Reverse Engineering of eMule An Analysis of the Implementation of Kademlia in eMule*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich.

Nucci, G. d. S. (2008). *Código Penal Comentado*. Editora Revista dos Tribunais, São Paulo, 9th edition.

Nucci, G. d. S. (2009). *Leis Penais e Processuais Penais Comentadas*. Editora Revista dos Tribunais, São Paulo, 4th edition.

Oram, A. (2001). *Peer-to-peer: Harnessing the power of disruptive technologies*. O'Reilly & Associates, Inc.

Organização das Nações Unidas (1989). Convenção Internacional sobre os Direitos da Criança. Disponível em: `http://www.gddc.pt/direitos-humanos/textos-internacionais-dh/tidhuniversais/dc-conv-sobre-dc.html`. Acessado em: 17 jun. 2011.

Palmer, G. (2001). A road map for digital forensic research. In *First Digital Forensic Research Workshop*, page 40.

Park, B.; Won, Y.; Choi, M.; Kim, M. e Hong, J. (2008). Empirical analysis of application-level traffic classification using supervised machine learning. In *APNOMS '08 Proceedings of the 11th Asia-Pacific Symposium on Network Operations and Management: Challenges for Next Generation Network Operations and Service Management*, pages 474–477.

Peron, A.; de Deus, F. E. e de Souza Júnior, R. T. (2011). Ferramentas e Metodologia para Simplificar Investigações Criminais Utilizando Interceptação Telemática. In *The Sixth International Conference on Forensic Computer Science (ICoFCS 2011)*, Florianópolis - Brasil.

Pinto, C. A. F. (2009). Pedofilia: Uma abordagem essencialmente jurídica. Disponível em: `http://www.recantodasletras.com.br/textosjuridicos/1405178`. Acessado em: 22 jan. 2011.

Plissonneau, L.; Costeux, J. e Brown, P. (2006). Detailed analysis of edonkey transfers on ADSL. *2006 2nd Conference on Next Generation Internet Design and Engineering, 2006. NGI '06.*, 00(c):255–262.

Pollitt, M. (2010). A History of Digital Forensics. In *Advances in Digital Forensics VI*, volume 337 of *IFIP Advances in Information and Communication Technology*, chapter Chapter 1, pages 3–15. Springer Boston.

Reith, M.; Carr, C. e Gunsch, G. (2002). An examination of digital forensic models. *International Journal of Digital Evidence*, 1(3):1–12.

Rocha, J. a.; Domingues, M.; Callado, A.; Souto, E.; Silvestre, G.; Kamienski, C. e Sadok, D. (2004). Peer-to-peer: Computação colaborativa na internet. *Minicurso, Simpósio Brasileiro de Redes de Computadores.*

Rosenblatt, F. (1958). *The perceptron: A theory of statistical separability in cognitive systems (Project Para).* Cornell Aeronautical Laboratory.

Ruibin, G.; Yun, T. e Gaertner, M. (2005). Case-relevance information investigation: binding computer intelligence to the current computer forensic framework. *International Journal of Digital Evidence*, 4(1):1G13.

Rumelhart, D. E.; Hinton, G. E. e Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, DTIC Document.

Rumelhart, D. E. e McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* A Bradford Book.

Russel, S. e Norvig, P. (2004). *Inteligência Artificial: tradução da segunda edição.* Elsevier, Rio de Janeiro, 2nd edition.

Salgarelli, L.; Gringoli, F. e Karagiannis, T. (2007). Comparing traffic classifiers. *ACM SIGCOMM Computer Communication Review*, 37(3):65–68.

Saroiu, S.; Gummadi, K.; Dunn, R.; Gribble, S. e Levy, H. (2002). An analysis of internet content delivery systems. *ACM SIGOPS Operating Systems Review*, 36(SI):315–327.

Sen, S.; Spatscheck, O. e Wang, D. (2004). Accurate, scalable in-network identification of p2p traffic using application signatures. *Proceedings of the 13th conference on World Wide Web - WWW '04*, page 512.

Sen, S. e Wang, J. (2002). Analyzing peer-to-peer traffic across large networks. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 137–150. ACM.

Sen, S. e Wang, J. (2004). Analyzing Peer-To-Peer Traffic Across Large Networks. *IEEE/ACM Transactions on Networking*, 12(2):219–232.

Sheng, L.; Dong, X.; Song, J. e Xie, K. (2010). Traffic Locality in the eMule System. *2010 First International Conference on Networking and Distributed Computing*, pages 387–391.

Shukla, A. e Tiwari, R. (2007). Fusion of Face and Speech Features with Artificial Neural Network for Speaker Authentication. *The Institution of Electronics and Telecommunication Engineers*, 24(5):359–368.

Steel, C. M. S. (2009). Child pornography in peer-to-peer networks. *Child abuse & neglect*, 33(8):560–8.

Steiner, M.; En-Najjary, T. e Biersack, E. W. (2007). A global view of kad. *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement - IMC '07*, page 117.

Sun, D.; Li, X.; Liu, W. e Wu, J. (2010). The New Architecture of P2P-Botnet. In *2010 Second Cybercrime and Trustworthy Computing Workshop*, pages 34–40.

Tanenbaum, A. S. e Wetherall, D. J. (2010). *Computer Networks (5th Edition)*. Prentice Hall.

Taur, J. e Tao, C. (2000). A new neuro-fuzzy classifier with application to on-line face detection and recognition. *The Journal of VLSI Signal Processing*, 26(3):397–409.

Taylor, M.; Haggerty, J.; Gresty, D. e Fergus, P. (2010). Forensic investigation of peer-to-peer networks. *Network Security*, 2010(9):12–15.

Tcptrace (2011). Shawn Ostermann – Universidade de Ohio. Disponível em: `http://www.tcptrace.org/index.html`. Acessado em: 20 jul. 2011.

Tutschku, K. (2004). A measurement-based traffic profile of the eDonkey filesharing service. In *Passive and Active Network Measurement*, pages 12–21.

Wang, J.; Qian, C.; Che, C.-H. e He, H. (2010a). Study on Process of Network Traffic Classification Using Machine Learning. In *The Fifth Annual ChinaGrid Conference*, pages 262–266.

Wang, K. (2010). Using a Local Search Warrant to Acquire Evidence Stored Overseas via the Internet. In *Advances in Digital Forensics VI*, pages 37–48.

Wang, Y.; Xiang, Y. e Yu, S.-Z. (2010b). Automatic Application Signature Construction from Unknown Traffic. *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pages 1115–1120.

Weiss, G., editor (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. The MIT press, Cambridge, Massachusetts.

Whitcomb, C. M. (2002). An Historical Perspective of Digital Evidence: A Forensic Scientist's View. *International Journal of Digital Evidence*, 1(1):1–9.

Williams, N.; Zander, S. e Armitage, G. (2006). Evaluating machine learning algorithms for automated network application identification. Technical Report March, Centre for Advanced Internet Architectures (CAIA).

Wolak, J.; Finkelhor, D.; Mitchell, K. e (US), N. C. M. . E. C. (2003). Internet sex crimes against minors: The response of law enforcement. Technical report, U.S. Department of Justice.

Wyoming DCI ICAC (2008). Wyoming Toolkit: Instalation and Usage Manual. Disponível em: `http://www.icactraining.org/`. Acessado em: 21 ago. 2011.

Xu, B.; Chen, M. e Fei, L. (2009). Distributed P2P Traffic Identification Method. In *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4.

Yar, M. (2006). *Cybercrime and society*. Sage Publications Ltd, Londres.

Yegnanarayana, B. (2004). *Artificial Neural Networks*. Prentice-Hall of India.

Zhang, Y.; Wang, H. e Cheng, S. (2010). A method for real-time peer-to-peer traffic classification based on C4.5. In *2010 IEEE 12th International Conference on Communication Technology*, pages 1192–1195.

Anexo A - Identificadores do eMule

Os identificadores dos pacotes utilizados para a comunicação do *eMule* são apresentadas neste anexo.

Tabela 6.1: Pacotes do eMule utilizados para a comunicação entre clientes na rede KAD, adaptado de Mysicka (2006)

Nome do Pacote	**Identificador**
Hello	0x10 e 0x11
Hello answer	0x18 e 0x19
Hello answer ACK	0x22
Bootstrap	0x00 e 0x01
Bootstrap answer	0x08 e 0x09
Search	0x30
Search answer	0x38 e 0x3B
Search notes	0x32 e 0x35
Search notes answer	0x3A
Search key	0x33
Search source	0x34
Publish	0x40
Publish answer	0x48 e 0x4B
Publish answer ACK	0x4C
Publish notes	0x42 e 0x45
Publish notes answer	0x4A
Publish key	0x43
Publish source	0x44
Firewall	0x50 e 0x53
Firewall answer	0x58
Firewall answer ACK	0x59
Find buddy	0x51
Find buddy answer	0x5A
Kademlia request	0x20 e 0x21
Kademlia answer	0x28 e 0x29
Call back	0x52
Kademlia ping	0x60
Kademlia pong	0x61

Tabela 6.2: Pacotes do eMule utilizados na comunicação entre o cliente e servidores na rede eD2k, adaptado de Kulbak e Bickson (2005)

Nome do Pacote	**Identificador**	**Protocolo**
Login	0x01	TCP
Server message	0x38	TCP
ID change	0x40	TCP
Offer files	0x15	TCP
Get list of servers	0x14	TCP
Server status	0x34	TCP
List of servers	0x32	TCP
Server identification	0x41	TCP
Search request	0x16	TCP
Search result	0x33	TCP
Get sources	0x19	TCP
Found sources	0x42	TCP
Callback request	0x1C	TCP
Callback requested	0x35	TCP
Callback failed	0x36	TCP
Message rejected	0x05	TCP
Get sources	0x9A	UDP
Found sources	0x8B	UDP
Status request	0x96	UDP
Status response	0x97	UDP
Search request	0x98 ou 0x92	UDP
Search response	0x99	UDP
Server description request	0xA2	UDP
Server description response	0xA3	UDP
Global get sources	0x94	UDP
Global get sources response	0x9B	UDP
Global call back	0x9C	UDP
Invalid low ID	0x9E	UDP
Server list request	0xA0 ou 0xA4	UDP
Server list response	0xA1	UDP

Tabela 6.3: Pacotes do eMule utilizados para a comunicação entre clientes na rede eD2k, adaptado de Kulbak e Bickson (2005)

Nome do Pacote	**Identificador**	**Protocolo**
Hello	0x01	TCP
Hello answer	0x4C	TCP
Sending file part	0x46	TCP
Request file parts	0x47	TCP
End of download	0x49	TCP
Change client ID	0x4D	TCP
Chat message	0x4E	TCP
Part hashset request	0x51	TCP
Part hashset reply	0x52	TCP
Start upload request	0x54	TCP
Accept upload request	0x55	TCP
Cancel transfer	0x56	TCP
Out of part requests	0x57	TCP
File request	0x58	TCP
File request answer	0x59	TCP
File not found	0x48	TCP
Requested file ID	0x4E	TCP
File status	0x50	TCP
Change slot	0x5B	TCP
Queue rank	0x5C	TCP
View shared files	0x4A	TCP
View shared files answer	0x4B	TCP
View shared folders	0x5D	TCP
View shared folders answer	0x5F	TCP
View content of a shared folder	0x5E	TCP
View shared folder content answer	0x60	TCP
View shared folder or content denied	0x61	TCP

Tabela 6.4: Pacotes do eMule utilizados para a comunicação entre clientes na rede eD2k, adaptado de Kulbak e Bickson (2005)

Nome do Pacote	**Identificador**	**Protocolo**
eMule info	0x01	TCP
eMule info answer	0x02	TCP
Sending compressed file part	0x40	TCP
Queue ranking	0x60	TCP
File info	0x61	TCP
Sources request	0x81	TCP
Sources answer	0x82	TCP
Secure identification	0x87	TCP
Public key	0x85	TCP
Signature	0x86	TCP
Preview request	0x90	TCP
Preview answer	0x91	TCP
Re-ask file	0x90	UDP
Re-ask file ack	0x91	UDP
Queue full	0x93	UDP

Tabela 6.5: Pacotes de rede utilizadas pelo eMule na rede eD2k que apresentam relevância pericial

Nome do Pacote e identificador	**Comunicação**	**Protocolo**	**Descrição**	**Momento em que é utilizado**	**Relevância na investigação**			
					Autoria	**Dolo**	**Materialidade**	**Transnacionalidade**
Offer files (`0x15`)	cliente → servidor	TCP	Relaciona os arquivos que estão disponíveis para outros usuários	Automaticamente durante o login ou quando ocorrem mudanças na lista de arquivos compartilhados	-	-	Sim (disponibilizar) artigo 241-A (ECA)	Sim
Search request (`0x16`)	cliente → servidor	TCP	Pedido de busca por arquivos tendo como critério dados digitadas pelo usuário	A busca por termos é realizada manualmente pelo usuário	Sim	Sim	-	-
Get sources (`0x19`)	cliente → servidor	TCP	Pedido de clientes que estão compartilhando os arquivos selecionados para serem baixados	Automaticamente quando um arquivo é selecionados para ser baixado	Sim	Sim	Sim (adquirir) artigo 241-B (ECA)	Sim
Get sources (UDP) (`0x9A`)	cliente → servidor	UDP	Solicitação por outros clientes que estão compartilhando esse arquivo	É enviada de forma automática quando o arquivo solicitado para ser baixado possui poucas fontes	-	Sim	-	-
Enhanced file search request (`0x98` e `0x92`)	cliente → servidor	UDP	Pedido de busca por arquivos enviado para todos os servidores da lista de servidores	A busca por termos é realizada manualmente pelo usuário	Sim	Sim	-	-
Chat message (`0x4E`)	cliente → cliente	TCP	Envia mensagens de chat para outro cliente	De forma manual	Sim	Sim	-	-
Sending file part (`0x46`)	cliente → cliente	TCP	Envia parte do arquivo que está sendo transferido	Automaticamente quanto um arquivo é transferido	-	-	Sim (transmitir) artigo 241-A (ECA)	Sim
Sending compressed file part (`0x40`)	cliente → cliente	TCP	Envia parte do arquivo que está sendo transferido	Automaticamente quando um arquivo é transferido	-	-	Sim (transmitir) artigo 241-A (ECA)	Sim
Hello (`0x10` e `0x11`)	cliente → servidor	TCP	Envia o *hash* do usuário do eMule e o endereço IP	Automaticamente quando é feito o login no servidor	Sim	-	-	Sim

Glossário

eMule é um dos aplicativos P2P mais utilizados no Brasil. 1, 3, 5–8, 24, 29, 47, 49, 50, 52, 53, 57, 59, 64, 67–88, 92, 94, 95, 98–106, 120

hash é uma função unidirecional que aceita uma entrada de tamanho variável e produz uma saída de tamanho fixo. Essa saída é dependente de todos os bits da entrada. Assim, mesmo que somente um bit da entrada seja modificado, a saída será diferente. 67, 69–71, 77, 100

internet service provider é a empresa que comercializa o acesso à Internet por meio um sistema de telecomunicações. 4, 5, 7, 27, 28, 53, 55–57, 98, 104, 105

overlay é uma rede virtual superposta a outra rede. 50, 52, 68, 77

interceptação telemática é a gravação da comunicação entre equipamentos ligados à informática ou telemática (misto de computadores e meios de comunicação). Necessita de autorização judicial e é regida pela Lei nº 9.296, de 24 de julho de 1996. 5, 7, 55, 83, 88, 94, 95, 99

mandado de busca e apreensão é a ordem judicial para que se procure uma coisa ou pessoa e realize sua apreensão. Pode ser expedido tanto em processos criminais como em processos cíveis. 2, 4, 5, 7, 19, 27, 28, 104