Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Caracterizando Distorções em Redes *Ad Hoc*

Osvaldo Corrêa do Nascimento Júnior

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientador
Prof. Dr. Jacir Luiz Bordim

Brasília
2008

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Informática

Coordenador: Prof. Dr. Li Weigang

Banca examinadora composta por:

Prof. Dr. Jacir Luiz Bordim (Orientador) – CIC/UnB
Prof. Dr. Georges Amvame Nze – ENE/UnB
Prof. Dr. Mario Antônio Ribeiro Dantas – INE/UFSC

CIP – Catalogação Internacional na Publicação

Nascimento Júnior, Osvaldo Corrêa do.

Caracterizando Distorções em Redes *Ad Hoc* / Osvaldo Corrêa do Nascimento Júnior. Brasília: UnB, 2008.
77 p.: il.; 29,5 cm.

Tese (Mestre) – Universidade de Brasília, Brasília, 2008.

1. Reputação, 2. Confiança, 3. Ad Hoc, 4. IEEE 802.11, 5. Métodos Estatísticos, 6. Camada de Enlace, 7. Cross-Layer

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro – Asa Norte
CEP 70910–900
Brasília – DF – Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Caracterizando Distorções em Redes *Ad Hoc*

Osvaldo Corrêa do Nascimento Júnior

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof. Dr. Jacir Luiz Bordim (Orientador)
CIC/UnB

Prof. Dr. Georges Amvame Nze
ENE/UnB

Prof. Dr. Mario Antônio Ribeiro Dantas
INE/UFSC

Prof. Dr. Li Weigang
Coordenador do Mestrado em Informática

Brasília, 1º de agosto de 2008

Dedicatória

Dedico este trabalho à minha esposa *Damara Santos Ribeiro*, e aos meus familiares em especial minha mãe *Eulália M. do Nascimento*, que me ensinou os valores e responsabilidades, que sem os quais, eu não seria quem sou hoje.

Agradecimentos

A Deus por estar sempre comigo em todos os momentos.

Ao professor doutor *Jacir Luiz Bordim*, pela força e incentivo na orientação deste trabalho.

A direção do Instituto de Ciências Exatas da Universidade de Brasília, pelo apoio na figura de seus Diretores.

Ao professor doutor *Rui Seimetz*, pelo apoio e incentivo.

Ao amigo *M.F. Caetano* pela colaboração nas pesquisas.

A minha esposa *Damara S. Ribeiro do Nascimento* a quem dedico todo meu amor, pelo apoio e compreensão.

Resumo

Redes *ad hoc* dependem da cooperação dos seus membros para a troca de dados. Nós mal intencionados, com objetivo de obterem vantagens, podem explorar falhas causando distorções, como alteração do processo de acesso ao meio. Isso exige medidas eficazes para a identificação e exclusão desses nós da rede.

Na literatura, há grande diversidade de sistema de detecção dessas distorções, adotando metodologias baseadas na coleta de dados em apenas uma camada do modelo TCP, mais precisamente a camada de rede. Poucas, entretanto, tratam de problemas que podem ser gerados na camada de enlace, como a burla do intervalo de tempo para acessar o meio, conhecido como *backoff*, ou alteração intencional do tempo de espera, enquanto outras estações utilizam o canal, ou NAV.

Devido ao caráter aleatório que envolve o processo de acesso ao meio, identificar distorções ou falhas causadas por um nó mal intencionado com a intenção de aumentar sua taxa de transmissão não é tarefa simples. Para verificar alterações é necessário obter amostras dos intervalos praticados pelo nó suspeito, para realizar deduções a respeito do comportamento. Fazer levantamento de quantas amostras são de fato necessárias para fazer inferência de maliciosidade torna-se algo desejável para um sistema eficiente.

Neste trabalho foi realizado levantamento do padrão IEEE 802.11, explorando as principais deficiências que podem ocorrer, além de apresentar estudo baseado em métodos estatísticos para proposição de solução eficiente para um modelo de sistema de reputação. Esse sistema utiliza informações da camada de enlace e tem por objetivo identificar distorções causadas por burla da intervalo de *backoff*, e o estabelecimento intencional do NAV. O modelo proposto conta com sistema de gerenciamento de índices de confiança que fornece informações às diversas camadas do modelo TCP, tornando possível a tomada de decisão independente nas camadas sobre o tráfego de pacotes.

Palavras-chave: Reputação, Confiança, Ad Hoc, IEEE 802.11, Métodos Estatísticos, Camada de Enlace, Cross-Layer

Abstract

Ad hoc networks work under the assumption that the devices composing the network are willing to collaborate with one another. Nevertheless, in a collaborative environment, selfish and misbehaving nodes may have a negative impact on the whole network. Hence, means to characterize and identify such behavior is necessary as its impact may affect the entire network.

In the literature, there are a number of works that deal with the problem of identifying and characterizing misbehavior on ad hoc network. However, these work focus on higher layer of the TCP/IP stack. In this work we focus on characterization and identification of misbehavior on the lower layers, more precisely, on the medium access control sub-layer, also known as MAC layer. At the MAC layer, nodes may change their behavior via modifications on the backoff window and other informations, such as the network allocation vector - NAV.

Owing to the nature of ad hoc networks, which are distributed, asynchronous and self organizing, identifying misconducting nodes in such environment is not a trivial task. In order to verify whether or not a node's conduct has been deviating from its normal or expected course, a number of samples need to be collect. Which enough samples, one can assess the conduct of a given node. In other words, the more one knows about its neighboring activity, the better the accuracy of its judgment will be. From the above, it is clear that one needs to know the number of samples necessary to collect in order to obtain a certain level of accuracy.

This work attempts to first identify the weakness of the IEEE802.11 that might be explored by a misconducting nodes. After that, we try to verify which means can be used to correct identify the exploitation of such vulnerabilities. With that in mind, a reputation system model is proposed, which is based on statistical methods used to characterize misconducting nodes. The proposed reputation system works by collecting information on the MAC layer and providing means to take action and impose restrictions on misconducting nodes.

Keywords: Reputation, Trust, Ad Hoc, IEEE 802.11, Statistical Methods, Data Link Layer, Cross-Layer

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

ACK	*Acknowledges*
AIFS	*Access Category Inter Frame Spacing*
ANATEL	*Agencia Nacional de Telecomunicações*
AODV	*Ad Hoc On-Demand Distance Vector*
CCA	*Clear Channel Assessment*
CONFIDANT	*Cooperation Of Nodes: Fainess In Dynamic Ad Hoc Network*
CORE	*Collaborative Reputation Mechanism*
CSMA/CA	*Carrier Sense Multiple Access With Collision Avoidance*
CSMA/CD	*Carrier Sense Multiple Access With Collision Detection*
CTS	*Clear to Send*
CW	*Contention Window*
DCF	*Distributed Coordination Function*
DIFS	*DCF Inter Frame Spacing*
DOMINO	*System for Detection of Greedy Behavior in the Mac Layer*
DoS	*Deny of Service*
DSDV	*Destination-Sequenced Distance Vector Routing*
DSR	*Dynamic Source Routing*
DSSS	*Direct Sequence Spread Spectrum*
EIFS	*Extended Interframe Space*
FTP	*File Transfer Protocol*
GPS	*Global Positioning System*
HTTP	*Hypertext Transfer Protocol*
ICMP	*Internet Control Message Protocol*
IEEE	*Institute of Electrical and Electronics Engineers*
IFS	*Inter Frame Spacing*
IP	*Internet Protocol*
ISO	*International Standards Organization*
LAN	*Local Area Network*
LLC	*Logical Link Control*
MAC	*Medium Access Control*
MAN	*Personal Area Network*
MANET	*Mobile Ad Hoc Networks*
MMPU	*MAC Management Protocol Data Unit*

MPDU	*Medium Access Control (MAC) Protocol Data Unit*
MSDU	*Medium Access Control (MAC) Service Data Unit*
NAV	*Network Allocation Vector*
OLSR	*Optimized Link State Routing*
OSI	*Open Systems Interconnection*
P2P	*Peer-to-Peer*
PAN	*Personal Area Network*
PC	*Point Coordinator*
PCF	*Point Coordination Function*
PDA	*Personal Digital Assistants*
PHY	*Physical Layer*
RREP	*Route Reply*
RREQ	*Route Request*
RRER	*Route Error*
RTS	*Request to Send*
SIFS	*Shorter Inter Frame Spacing*
SLRC	*Station Long Retry Count*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
SNR	*Signal-to-Noise Ratio*
SORI	*Secure and Objective Reputation-Based Incentive Scheme*
SRP	*Secure Routing Protocol*
SSRC	*Station Short Retry Count*
STA	*Station*
TCP	*Transmission Control Protocol*
TELNET	*Telecommunication Network*
TLC	*Teorema do Limite Central*
TTL	*Time to Live*
UDP	*User Datagram Protocol*
WAN	*Wide Area Network*

Capítulo 1

Introdução

A miniaturização dos componentes de comunicação, juntamente com a disponibilidade de processadores mais rápidos e de baixo custo, possibilitaram a disseminação desses dispositivos em escala global. De acordo com a ANATEL[4], no último mês de junho, existem no Brasil mais de 133 milhões de celulares, o que equivale acerca de 72,68% da população. Segundo o mesmo estudo, só no Distrito Federal a taxa é de 1,27 celulares por pessoa[4], o que mostra forte uso de dispositivos móveis. A utilização de novos serviços e o potencial de expansão desse parque de dispositivos são, de fato, promissores. As empresas de telefonia há muito vêm buscando disponibilizar serviços móveis, não apenas como meio de comunicação, mas de entretenimento.

Nessa mesma vertente, não só o uso de celulares, mas de computadores e PDAs também tem seu espaço. A conexão desses dispositivos de forma independente, ou seja sem ponto concentrador é chamada de redes *ad hoc*. A palavra *ad hoc* é uma expressão que vem do latim e significa feito para isso, mas na linguagem da computação a palavra *ad hoc* expressa algo que não tem ponto concentrador, ou seja, não necessita de ponto onde todos os dispositivos estejam conectados para trocarem informações entre si. Em uma rede, a cooperação entre os membros é de fundamental importância, uma vez que não há ponto central de conexão.

Esses dispositivos, por si sós, não são capazes de provocar nenhum mal; pelo contrário, contribuem para o desenvolvimento do bem comum. Mas quando se pensa nas pessoas que estão operando tais dispositivos vem à tona uma questão: até que ponto as informações trocadas nessas redes são confiáveis? Será que uma informação na origem é a mesma no destino? A taxa de transmissão é a mais justa? Até que ponto as pessoas estão dispostas a cooperar? Uma vez que a cooperação é de fundamental relevância em redes *ad hoc*, isso, de fato, precisa ser levado em conta.

Pensando no conceito de cooperação, a maioria das pessoas tem tendência natural a cooperar com intenção de obter algo em troca. A relação entre indivíduos e o conceito de cooperação baseado no conceito das relações humanas e de comunidade são tratadas por Gambetta (1990) [11]. Gambetta apresenta idéia dos conceitos da cooperação racional e os reais motivos pelos quais os indivíduos tendem a cooperar, além de definir o que seria certo

nível de confiança entre dois indivíduos ou *Trust*. A confiabilidade das informações, assim como a sua confidencialidade, é assunto tratado por diversos pesquisadores [32, 17, 24, 41] quando se referem a redes de dispositivos em modo *ad hoc*.

O uso de métodos que estabeleçam algum nível de confiança ou segurança no processo de comunicação entre dispositivos em redes *ad hoc* é de fato necessário devido a grande dificuldade de estabelecer meios que garantam aos indivíduos membros a contribuição dos demais em cem por cento dos casos de forma justa. Esses métodos podem ser classificados como *Hard Security* (uso de métodos criptográficos ou que requeiram autenticação) e o *Soft Security* (uso de métodos estabelecidos apenas na reputação e confiança) [27]. Pensando no uso de criptografia, esta poderia ser saída óbvia para muitos. No entanto, há casos em que esse processo pode se tornar muito caro do ponto de vista do processamento ou do uso da energia, o que pode tornar o uso da criptografia custoso em dispositivos móveis. Já métodos baseados na análise do comportamento, definidos simplesmente como CONFIANÇA - *Trust* [21], podem servir como solução para dispositivos que não possuem tanto poder computacional para trabalhar com chaves criptográficas mas são capazes de monitorar e avaliar o comportamento do seus vizinhos a fim de definir o seu grau de confiabilidade. De fato, a *confiança - trust* não estabelece níveis de segurança como as soluções que envolvem a criptografia, pois essas visam a campos distintos: a primeira busca cooperação e reputação enquanto a segunda busca identificação e integridade. Porém, do ponto de vista do estabelecimento de confiança, pode-se aproximar, dependendo do nível da reputação, tanto quanto se queira o grau de confiabilidade a que se deseja.

De forma geral, as soluções que estabelecem segurança por meio de sistemas de confiança têm como núcleo mecanismos responsáveis pela coleta, avaliação e mensuração da análise do comportamento observado a fim de prover confiabilidade ou nível de confiança, e estes mecanismos recebem a definição de sistemas de reputação Gambetta (1998) [11].

Neste ponto, deve-se tomar cuidado para que não haja mistura dos conceitos de *confiança* e *reputação*, pois apesar de possuírem interdependência, ambos são distintos. Para exemplificar essa relação, pode-se ficar com a seguinte definição de Jousang (2006)[3]: "Eu confio em você porque você tem uma boa reputação"e "Eu confio em você apesar da sua má reputação". Essas afirmações deixam claro que a questão da confiança não depende só da reputação, mas de outros fatores que podem ser desde observações, reputações anteriores ou, dependendo do grau de má reputação, o nível que se pretende de confiabilidade.

As redes que usam ondas de rádio para se formarem têm por natureza sérios problemas de segurança quanto aos dados trafegados nelas. Uma vez propagadas, as ondas de rádio não fornecem meios por si sós para inferir quais os indivíduos estão recebendo seu sinal, isto é, não se pode definir com precisão que de dois dispositivos em um mesmo raio de alcance um possa receber o sinal e outro não. Observando deste prisma, tornaram-se necessárias medidas para que se limitasse o acesso às informações propa-

gadas. Para facilitar a implementação dessas redes e criar padrão a ser seguido foi que em 1997 a IEEE criou grupo de pesquisadores responsável em gerar novo padrão. Daí deu-se origem ao padrão IEEE 802.11. Esse padrão é responsável por duas camadas fundamentais na arquitetura das redes sem fio: enlace e física. Conseqüentemente, a crescente demanda por tais dispositivos não poderia evoluir sem que fossem estabelecidos padrões para isso. Imagine se cada fabricante resolvesse criar dispositivos cada um operando em uma faixa de freqüência diferente, ou desenvolvendo métodos de acesso ao meio de maneira diferente em relação a outro. Isso acabaria causando verdadeira desordem. O padrão IEEE 802.11 foi criado da necessidade de estabelecer regras para a criação de dispositivos sem fio. Dentre as inúmeras definições criadas pelo padrão IEEE 802.11, ele estabelece de que maneira o nó de uma rede deve cessar o meio da forma mais justa possível, além de definir quais são as regras para a disputa do acesso. Aqui, fica notório que o estabelecimento de confiança em redes sem fio passa necessariamente pelo estudo desse padrão, e aqui fica claro que rompimento das regras de acesso ao meio ou outras que definam compartilhamento da rede dependem também de cooperação.

O estudo dos padrões, protocolos e sistemas de reputação em redes *ad hoc* é de grande relevância para os casos em que se tem dispositivos que possuem poder computacional limitado, como é o caso de sensores responsáveis pela coleta de dados importantes ou até a utilização de dispositivos para fins militares. Para estabelecer modelo que utilize informações das camadas de enlace, é preciso realizar estudo do comportamento do *backoff*, de maneira a considerar o seu caráter aleatório. Assim, serão utilizadas ferramentas baseadas em inferência estatística, em que por meio da coletas das informações aleatórias ou amostras, pode-se ter projeção do comportamento da população, que neste caso são os intervalos de *backoff - Processo aleatório de acesso ao meio*.

Apesar da maioria dos trabalhos pesquisados fazerem alusão a problemas da camada de rede, as informações contidas em outras camadas podem fornecer indícios de distúrbios causados por mal comportamento. Exemplo disso seria o caso de um nó transmitir mensagem para outro nó e esse não responder. Logo, não se pode inferir se este está descartando os pacotes ou está com limitações de bateria ou se está havendo colisões. Mas se outros nós recebem mensagens do nó em questão fica claro problema de negação de serviço.

1.1 Objetivos

Este trabalho visa estudar as diversas características do padrão IEEE 802.11, além das distorções causadas por mal comportamento em uma rede *ad hoc*, resultando na abordagem de sistema de reputação que estabeleça níveis de confiança para as diversas camadas do modelo TCP utilizando, para isso, técnicas de *cross-layer* para obter informações entre as camadas.

1.2 Estrutura da Monografia

O estudo a seguir está organizado em cinco capítulos e um apêndice. No capítulo 2, estão as características das redes sem fio, serão definidos os conceitos das arquiteturas baseadas em camadas além do padrão IEEE 802.11 e será apresentado comparativo entre os protocolos de roteamento existentes. No capítulo 3, é feito levantamento das possíveis distorções que podem ser geradas por mal comportamento, além de levantar o estado da arte em soluções propostas para os problemas abordados. No capítulo 4, por meio de estudo estatístico é proposta abordagem de sistema de reputação baseado em multicamadas ou *Cross-Layer*. Na sequência, estão as considerações finais e são estabelecidas metas para trabalhos futuros.

Capítulo 2

Redes sem Fio

Os dispositivos sem fio nunca tomaram tanto espaço como nos últimos anos. Hoje, os dispositivos sem fio (*wireless*) já fazem parte do dia-a-dia das pessoas, formando os mais diversos tipos de rede. De celulares, passando por PDAs, *laptops*, receptores de GPS em carros, entre outros. Das diversas vantagens das redes sem fio, a mobilidade e a facilidade de implementação sem dúvida são as que mais atraem usuários. Para que esta facilidade seja posta em prática, as redes sem fio são classificadas em tipos que serão tratados na seção a seguir[15].

2.1 Classificação das Redes sem Fio

As redes baseadas em ondas de rádio estão sujeitas a algumas limitações: interferências, baixa banda de comunicação, limitação de processamento devido ao pequeno tamanho e restrição quanto ao uso da energia. Partindo das peculiaridades das redes baseadas em ondas de rádio, pode-se classificá-las em dois modos:

- *Infra-Estruturada*
  A rede sem fio é conectada à rede física por meio de ponto de acesso (*Access Point - Gateway*). Este também é responsável em controlar o uso do meio (ondas de rádio) entre as estações. Exemplo de rede sem fio que adota este modelo são as redes de acesso público *hotspots*, populares em locais como aeroportos, hotéis e restaurantes. Devido a popularização do uso de computadores portáteis e o barateamento do *hardware Access Point*, este modelo de rede está se tornado opção barata e de fácil implementação, muito empregada em pequenas redes de escritórios ou residências.

- *Sem Infra-Estrutura - ad doc*
  Estas redes não dependem de rede estruturada. Os dispositivos, também definidos como nós, formam de maneira independente redes dinâmicas de comunicação. Diferente das redes cabeadas que possuem *gateway* responsável por rotear os pacotes, em redes *ad hoc* cada dispositivo é um roteador em potencial, isto é, são responsáveis por trans-

ferir pacotes para outros nós da rede. Celulares por meio de conexões *bluetooth*, trocando imagens ou toques (*ringtones*), são um bom exemplo de redes sem infra-estrutura ou redes *ad hoc*.

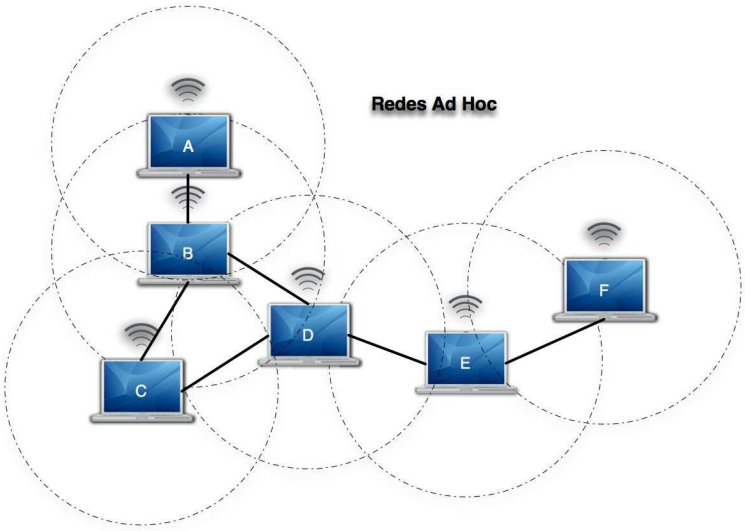

Figura 2.1: Rede *Ad Hoc*

As redes sem fio são classificadas e recebem nomenclaturas dependendo do raio de alcance das redes sem fio e do tipo de dado trafegado.

- *Wireless Wide Area Network (Wireless WANs)*
  São redes baseadas em infra-estrutura onde estações base cobrem grande área, fornecendo conexões para dispositivos móveis. Ex: Rede de Celulares.

- *Wireless Metropolitan Area Networks (Wireless MANs)*
  Essas redes geralmente são usadas para ligações entre pontos fixos, como a ligação entre dois edifícios de escritórios.

- *Wireless Local Area Networks (Wireless LANs)*
  Estabelecidas em pequenos locais, geralmente com cem metros de alcance em área livre (*outdoor*) baseadas em modo infra-estrutura tendo como exemplos as redes em pequenos escritórios e redes de acesso público em aeroportos. Este tipo é normatizado pelo padrão IEEE 802.11[15] que será visto em mais detalhes na seção 2.3. Redes que adotam as especificações IEEE 802.11 recebem o termo WI-FI.[10, 20]

- *Wireless Personal Area Networks (Wireless PANs)*
  São redes de pequeno alcance, em média dez metros, muito utilizada em dispositivos que adotam tecnologia *bluetooth* [34], como *mouse* sem fio ou até mesmo telefones celulares para troca de mensagens. Em sua maioria são redes baseadas no modo *ad hoc* e mantém interoperabilidade com redes IEEE 802.11.

Nota-se que há gama de definições e classificações que normatizam as redes sem fio. Porém, a partir de agora, o foco são redes *ad hoc* baseadas no padrão IEEE 802.11, onde está o interesse deste trabalho. Sendo assim, nas próximas seções estão mais detalhes dos padrões e protocolos adotados para este tipo de rede sem fio. As redes *ad hoc* também recebem outra nomenclatura chamada de MANET (*Mobile Ad Hoc Network*).

2.2 Arquiteturas

A forma como membros de uma rede trocam informações são regidas por diversidade de protocolos que necessitam de organização. Dependendo das características de uma rede, podem haver várias formas de realizar esta organização, gerando diversas arquiteturas de rede. A seguir, verifica-se como podem ser formadas essas arquiteturas, por meios dos modelos OSI e TCP.

2.2.1 Modelo OSI

Com o advento das primeiras implementações de redes sem fio, as redes físicas já estavam estabelecidas e com suas tecnologias mais amadurecidas. Imagine-se uma rede onde cada membro fale ou transmita mensagens sem padrão definido. Naturalmente seria difícil estabelecer troca eficiente de dados. Para que os membros dessas redes, pudessem comunicar-se de forma harmoniosa, a ISO (*International Standards Organization*) estabeleceu camadas de forma a abstrair a funcionalidade dos diversos protocolos de comunicação existentes. Este conjunto de camadas recebeu o nome de *Modelo OSI*, o qual é composto de sete camadas - figura 2.2 -, sendo cada uma responsável por uma função, de acordo com conjunto de protocolos. Essas camadas trabalham de forma independente, isto é, o que uma camada acima faz as camadas abaixo não interferem. A seguir, o que cada uma deve fazer, lembrando que o modelo OSI não implementa protocolos, apenas diz o que deve ser feito em cada camada [5].

- **Aplicação**
  A camada de aplicação é a que fica mais perto do usuário final. É nesta camada que ficam os protocolos responsáveis por tratar as informações e entregá-las para o usuário final FTP (*File Transfer Protocol - Protocolo de Transferência de Arquivos*) é um exemplo.

- **Apresentação**
  Tem por função regulamentar e abstrair as semânticas das estruturas de dados entre diferentes dispositivos.

- **Sessão**
  Estabelecer sessão entre usuários de diversos dispositivos.

- **Transporte**
  A função básica da camada de transporte é estabelecer comunicação entre os membros da rede, incluindo controlar o fluxo de dados. Outra função da camada de transporte é abstrair qualquer diferença de tecnologia de *hardware*.

- **Rede**
  A camada de rede é responsável pela forma, como os pacotes serão roteados pela subrede, pegando os pacotes recebidos e definindo origem e destino.

- **Enlace**
  Esta camada tem como função receber os dados da camada física e transformá-los em *bits* organizados em quadros que serão entregues à camada de rede. A camada de enlace também é responsável por controlar o fluxo de dados entre dispositivos a fim de confirmar a entrega. Nesta camada existe subcamada que fica responsável por controlar o acesso do canal compartilhado, ou seja, controlar o acesso ao meio.

- **Física**
  Nesta camada, encontra-se toda parte de *hardware* envolvida na comunicação dos dados.

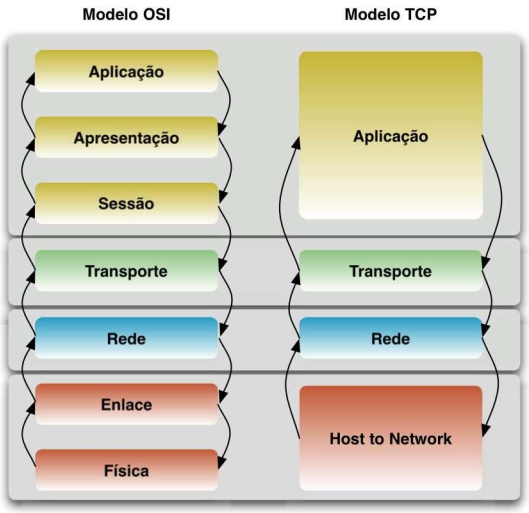

Figura 2.2: Camadas do Modelo OSI Comparadas ao Modelo TCP

2.2.2 Modelo TCP/IP

Apesar da sua organização, a arquitetura baseada no modelo OSI não aborda questões de intercomunicação entre redes diferentes. Com o advento das primeiras redes de pesquisa, mais precisamente a ARPANET - *Advanced*

Research Projects Agency Network, houve a necessidade de resolver questões de interligações entre redes, de maneira que determinado dado fosse da origem ao destino sem maiores problemas. Em maio de 1974, foi proposta por dois pesquisadores do IEEE, Cerf e Kahn, outra arquitetura baseada em modelo de camadas chamado modelo TCP/IP. Esse modelo, diferentemente do OSI, possui apenas quatro camadas: Aplicação, Transporte, Internet, *Host-to-Network*, conforme figura 2.3.

- **Aplicação**
  Semelhante ao Modelo OSI, é na camada de aplicação que estão estabelecidos os protocolos de mais alto nível [5], ou seja protocolos como HTTP (*HyperText Transfer Protocol* - Protocolo de Transferência de Hipertexto) responsáveis por serviços *web*, entre outros, como SMTP (*Simple Mail Transfer Protocol* - Protocolo de Transferência Simples de Mail) usado na transmissão de mensagens eletrônicas.

- **Transporte**
  Esta camada é responsável por estabelecer comunicação entre dispositivos da rede. Nesta camada estão definidos o TCP (*Transport Control Protocol* - Protocolo de Controle de Transmissão), seu objetivo principal é estabelecer conexão e entrega de dados (pacotes) entre dispositivos, e o UDP (*User Datagram Protocol*), entre outros. Enquanto o TCP tem a preocupação com a entrega dos dados e que esses cheguem ao destino de maneira correta, o UDP já se preocupa com o fluxo de dados que deve ser rápido e contínuo. O UDP não possui verificação de entrega de dados como no TCP.

- **Internet**
  Os dados quando passam pela camada de transporte são organizados em pacotes e entregues para esta camada. O objetivo dessa camada é definir como esses pacotes serão entregues entre origem e destino, independentemente do caminho percorrido. Isso é definido pelo protocolo IP (*Internet Protocol*) que estabelece sistema de endereços de origem e destino para cada pacote.

- ***Host-to-Network***
  Esta camada não foi detalhada quando da criação do modelo TCP/IP. No entanto, posteriormente, foi definida em mais detalhes e dividida em três subcamadas: LLC (*Logical Link Control* - Controle de Link Lógico), MAC (*Media Access Control* - Controle de Acesso ao Meio), PHY (*Physical* - Física). As duas primeiras subcamadas LLC e MAC formam o que se chama de camada de enlace.

  – A subcamada LLC é padronizada pelo IEEE 802.2 [1] e é responsável por colocar informações na forma de cabeçalho (números de seqüência e de confirmação) nos pacotes passados pela camada internet. Essas informações podem ser usadas para estabelecer conexões confiáveis em nível de camada de enlace.

– Nas subcamadas MAC e PHY está o divisor de padrões quando se trabalha com redes cabeadas e redes sem fio. É notório que os dois tipos de redes possuem características próprias, exigindo padrões distintos. As subcamadas MAC e PHY são estabelecidas pelo IEEE 802.3 [2] para as redes cabeadas e o IEEE 802.11 [15] para redes sem fio.

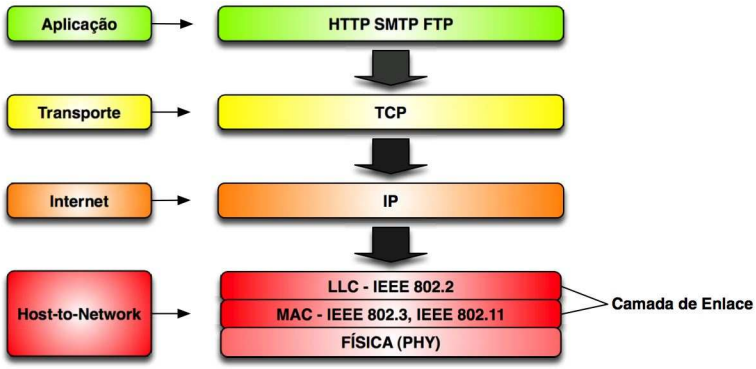

Figura 2.3: Modelo TCP/IP e Protocolos por Camadas

O objetivo dos modelos apresentados é estabelecer camadas onde estão organizados os diversos protocolos necessários ao funcionamento de uma rede. Assim, para que um dado saia de um nó e chegue em outro, ele passa rigorosamente de uma camada para outra sempre obedecendo a ordem entre elas, isto é, um dado gerado na camada de aplicação é passado para a de transporte que organiza os dados em pacotes que são passados para a de internet, que introduz endereços de origem e destino, depois são passados para a camada de enlace e, conseqüentemente, para a camada física.

As redes sem fio quando foram idealizadas adotaram a arquitetura baseada em camadas. As redes cabeadas funcionam de forma satisfatória neste tipo de arquitetura; no entanto, as redes sem fio possuem particularidades, que tornam esse modelo inadequado. Questões como mobilidade, por exemplo, onde rotas mudam constantemente poderiam ser mais bem resolvidas se informações da camada física pudessem ser vistas pela camada de aplicação e vice-versa sem passar pelas outras camadas. Para resolver estas e outras questões, pesquisadores [35] propuseram nova idéia, chamada de *Cross-Layer*.

2.2.3 *Cross-Layer*

O uso das arquiteturas baseadas em camadas contribuiu, de fato, para a popularização das redes com conexão ponto a ponto. O motivo que impulsionou a evolução desses modelos veio com o desenvolvimento das redes *ethernet*. As redes *ad hoc*, por sua vez, adotam o modelo TCP/IP baseado em camadas rígidas, onde não é permitida a troca de mensagens entre camadas não adjacentes, podendo, em alguns casos, ter sua performance

degradada. Isto acontece devido a natureza das conexões entre dispositivos de uma rede *ad hoc*.

Diferentemente das redes cabeadas, em que a conexão é estabelecida ponto a ponto por meio físico, as redes sem fio têm suas conexões por meio de ondas magnéticas (sinais de rádio)[35]. Conexões estabelecidas por ondas de rádio estão sujeitas a diversos fatores que podem comprometer sua eficiência, como interferências, obstáculos físicos, além da pouca capacidade de provisão de energia pelos dispositivos móveis. A troca de informações entre as camadas pode contribuir para que haja aumento considerável na performance dessas redes.

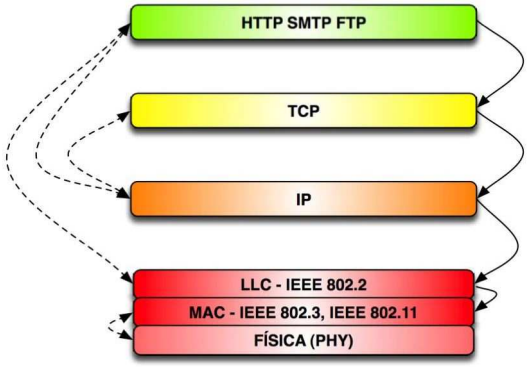

Figura 2.4: Conceito Geral de *Cross-Layer*

As camadas do modelo TCP/IP trocam informações por meio de chamada e resposta de forma independente, ou seja, as camadas apenas repassam as informações para as camadas adjacentes e isso ocorre porque essas camadas não foram projetadas de forma a tratar interações entre outras camadas ou entender o que as outras fazem. Para suprir essa deficiência são propostas algumas alternativas que são mostradas no estudo do Srivastava [35], em que são definidas algumas técnicas. De acordo com a figura 2.5, pode-se ver, de forma simplificada, diferentes tipos de projetos propostos para o *cross-layer*.

Nos três primeiros projetos da figura 2.5, verifica-se que as camadas trocam informações sem necessariamente serem adjacentes. Esta troca ocorre por meio da criação de interfaces entre as camadas.

O quarto projeto usa a idéia da criação de supercamada, em que os dados entre as camadas são trocados. Nesta abordagem não é necessária a criação de interfaces.

O quinto projeto apresentado pela figura 2.5 estabelece que as camadas são capazes de serem adaptadas para receber informações de maneira a dar maior desempenho nas trocas de dados, como ter camadas capazes de receber mais de um pacote ao mesmo tempo. Esta abordagem possui a vantagem de não necessitar da criação de camadas, nem interfaces.

O sexto projeto apresentado mostra a idéia da calibração vertical. Esta abordagem baseia-se no ajuste de parâmetros nas diversas camadas para melhorar o desempenho dependendo do dado transmitido pelas outras ca-

madas. A vantagem desta abordagem é que não é necessário alteração do modelo TCP. No entanto, os protocolos necessitariam de adaptações.

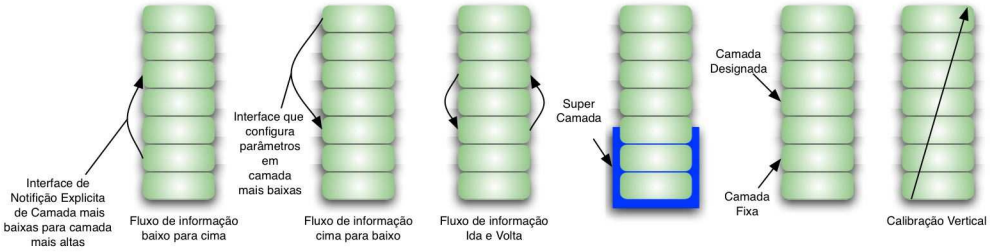

Figura 2.5: Propostas de Interações entre Camadas [35]

Apesar de o conceito de *cross-layer* ser inovador, não existe modelo padrão a ser seguido entre os diversos publicados [36, 29, 26, 38]. Em geral, as pesquisas publicadas sobre *cross-layer*, buscam estabelecer performance nas redes *ad hoc*, tratando especialmente as interações entre as camadas física, MAC e transporte. Conexões baseadas em ondas de rádio têm variações que as tornam diferentes das estabelecidas em redes cabeadas. Saber o que ocorre nestas camadas ajuda outras, como a de transporte a controlar supostos congestionamentos causados especialmente por protocolos com o TCP, que controla o tráfego de pacotes e depende de resposta do envio desses. É claro que vários cenários podem ser criados, com o objetivo de ganho de desempenho em redes móveis.

Diferentemente das pesquisas que visam ao desempenho, este trabalho busca utilizar informações coletadas em camadas, por exemplo às física e MAC, de forma a mensurar comportamentos e estabelecer confiabilidade entre membros de uma rede *ad hoc*, e neste ponto o conceito de *cross-layer* é de fundamental contribuição.

2.3 Padrão IEEE 802.11

Com a criação das redes sem fio, a camada MAC e a camada física precisavam ser redefinidas para atuarem em redes com conexões baseadas em ondas de rádio. O controle de acesso ao meio é bom exemplo da necessidade desta redefinição. Se adotado o mesmo controle de acesso ao meio feito para redes cabeadas teriam-se os conhecidos problemas de terminal escondido e terminal exposto.

O padrão IEEE 802.11 vem com a proposta de estabelecer todas as regras que atuam nas camadas físicas e MAC do modelo TCP/IP para redes sem fio. Na seqüência estão as principais características de 802.11.

2.3.1 A Camada Física

É responsável pelas interfaces elétrica e mecânica que propiciam a telecomunicação (comunicação a distância). É chamada de camada física por utilizar os recursos naturais do planeta, não tendo qualquer relação com a lógica criada para gerenciar a comunicação, isto é, responsabilidade dos protocolos. Os protocolos de rede nada mais são que algoritmos utilizados para que, de forma lógica, permitam a comunicação de uma estação A com uma estação B.

- Rádios

  Potência de Transmissão do Sinal

  Tipo de Antenas

2.3.2 A Camada de Enlace

A camada de enlace de dados existe sempre que há conexão física entre dois pontos, seja através de um *link* com fio (*wired*) ou sem fio (*wireless*), não importa: um *software* vai ter de dar sentido a toda esta transmissão física. A camada de enlace de dados fica responsável por informar onde começa uma informação, onde termina o endereço de origem e destino (afinal, não basta transmitir e receber, tem de informar para qual máquina se quer transmitir), é responsável também por identificar se algum *bit* foi danificado assim como corrigir a informação em alguns tipos de protocolo, outra funcionalidade é dizer se alguma informação não chegou ao receptor. Para executar essa função, ela utiliza algumas lógicas, como quando o receptor receber seqüência de 8 *bits* com o valor 1, isto pode significar que é o início do quadro que transporta a informação, quando tiver outra determinada seqüência pode significar o fim, existem várias formas lógicas de se resolver um mesmo problema.

A camada de enlace de dados também possui subcamada que é a de acesso ao meio. A necessidade dessa camada surge do problema que se tem quando existe um meio compartilhado, no caso de comunicação multiponto. Verifique-se o exemplo: um AP possui vários clientes que vão ter de se comunicar através de um mesmo canal, o problema é que se um transmitir junto do outro eles irão colidir, então é necessário *software* para otimizar o uso deste canal, no caso de redes 802.11.a/b/g, o objetivo dos projetistas foi que os clientes tivessem a mesma prioridade para transmitir o padrão 802.11e [15] que veremos no futuro, terá acesso ao meio com prioridades diferentes. O objetivo é que exista QoS (qualidade de serviço) para que o tráfego de voz e imagens, por exemplo, possa ter prioridade na rede.

MAC é uma subcamada da camada de enlace responsável por controlar o acesso ao meio. A arquitetura da MAC é composta de duas funções, *Distributed Coordination Function* (DCF) e o *Point Coordination Function*

(PCF), podendo as duas coexistirem em uma mesma IEEE 802.11 LAN.

2.3.2.1 DCF

O principal método de acesso utilizado pelo 802.11 é o DCF conhecido como um *Carrier Sense Multiple Access With Collision Avoidance* (CSMA/CA). Para que uma estação possa transmitir, ela deve ter certeza que o meio está ocioso, o que pede inicialmente verificação para checar se o mesmo está ocupado ou não; se estiver ocioso, o processo de transmissão é realizado. Se o meio estiver ocupado, a estação deve permanecer em silêncio até que ocorra o fim da transmissão. Depois de aguardar a transmissão ou antes de tentar transmitir novamente e imediatamente depois de uma transmissão bem-sucedida deve ser escolhida de forma randômica um intervalo de *backoff*. Este intervalo deve ser decrementado enquanto o meio estiver ocioso até que seu valor seja zerado, onde será feita nova tentativa de acesso ao meio. Depois de verificada a ociosidade do meio, o total decaimento do *backoff* e antes do envio de qualquer *frame* são trocadas curtas mensagens de controle de *frames* - *Request to Send* (RTS) *Clear to Send* (CTS). É atribuído intervalo *gap* entre os *frames* - IFS *Interframes Spaces*.

Cada *frame* CTS e RTS contém um campo *Duration/ID* que define o período de tempo que o meio deverá ficar reservado até que a estação de destino retorne um *ACK frame*. Dessa forma, as outras estações podem calcular o NAV e saber quanto tempo elas devem ficar sem transmitir e, posteriormente, tentar outro acesso ao meio obedecendo ao decaimento do seu *backoff*. Para os casos de seqüência de fragmentos, a ACK *frame* é seguida de outro *frame*.

2.3.2.2 PCF

Usado apenas em configurações de rede em modo infra-estrutura, o PCF é um modo de acesso ao meio opcional do MAC que pode ser incorporado. Esse método de acesso usa um *Point Coordination* (PC) que controla o ponto de acesso que determinará quais estações terão acesso ao meio. O PCF usa *carrier-sense* virtual - funções usadas para verificar o estado do meio, sendo virtuais ou físicas - somado a mecanismo de prioridade. Para o controle do meio frente às estações setando o NAV, o PCF administra a distribuição de *Beacon Frames*. O PCF utiliza IFS menor (DIFS) que no DCF para ter prioridade de controle do meio frente às estações.

2.3.2.3 Fragmentação e Defragmentação

Fragmentação é o processo de particionamento de um *MAC service data unit* (MSDU) ou *MAC management protocol data unit* (MMPDU) em pequenos *frames* chamados de *MAC protocol data unit* (MPDUs). A fragmen-

tação de um MSDU em pequenos MPDUS é para aumentar a probabilidade de sucesso de transmissão em canais que limitam a recepção de *frames* longos. O processo de fragmentação e defragmentação é permitido para cada imediata transmissão e recepção. Somente MPDUs com endereço de *unicast receiver* devem ser fragmentados, os *frames* de *Broadcast*/*Multicast* não devem ser fragmentados mesmo que seu tamanho exceda o limite do *aFragmentation Threshold*. Entretanto, se um MSDU for recebido com o tamanho maior que o limite *aFragmentation Threshold* o MSDU deverá ser fragmentado e cada fragmento não deverá ser menor que o limite *aFragmentationThreshold*. Cada fragmento é transmitido independentemente separados com um *ack*. Pode ocorrer a transmissão dos fragmento de um MSDU em um *Burst* durante o período de contenção usando uma única solicitação do meio pelo procedimento DCF.

O RTS/CTS não pode ser usado para MPDUs com *broadcast* e *multicast* pelo fato de existirem múltiplos destinos para o RTS, o que causaria a ocorrência de respostas de CTS como conseqüência. O RTS/CTS está sob controle do atributo *dot11RTSThresold*, o qual pode ser configurado em cada estação. Apesar de cada estação poder operar em diferentes taxas de transmissão, o RTS/CTS é sempre transmitido obedecendo a taxa especificada no atributo *aBasicRateSet*

2.3.2.4 O Mecanismo de *Carrier-Sense*

Os *carrier-sense* são usados para verificar o estado do meio e são definidos como físico e virtual. Quando essas funções indicam que o meio está ocupado, esse deve estar, caso contrário, ocioso. O *carrier-sense* físico é provido pela camada física enquanto o *carrier-sense* virtual é provido pela camada MAC. O *carrier-sense* virtual depende do NAV calculado baseando-se nos valores informados pelos *frames* RTS/CTS no seu campo *Duration*/*ID*. O NAV é um contador que decresce até zero partindo do valor encontrado pelos RTS/CTS. O mecanismo de *carrier-sense* combina os valores do NAV e as informações vindas da camada física para definir se uma estação está ocupada ou ociosa, fazendo com que os dois mecanismos trabalhem em conjunto.

2.3.2.5 Espaço Interframes (IFS)

O intervalo de tempo entre os *frames* é chamado de IFS. As estações determinam se o meio está ocioso utilizando as funções do *carrier-sense* para intervalo especificado. Quatro diferentes IFS, *Shot interframe space* (SIFS), *PCF interframe space* (PIFS), DCF *interframes space* (DIFS) e *extended interframes space* (EIFS), definidos do mais curto até o mais longo, respectivamente, devem ser independentes da taxa de transmissão de cada estação.

- *Short* IFS (SIFS) É o interframes mais curto e é utilizado depois de um ACK, CTS *frames*, a seqüência de um MPDU nos casos de fragmentação *burts*. Nos casos em que a estação precisa modificar o tempo de transmissão e manter o acesso ao meio com o objetivo de transmitir uma seqüência de *frames*, as estações usam um *gap* menor entre os *frames*, evitando que outras estações tomem o acesso ao meio.

- Short IFS (SIFS)

- PCF IFS (PIFS) O PIFS é usando apenas por estações (*Access Point*) operando em PCF para ganhar prioridade de acesso ao meio.

- DCF IFS (DIFS) O DIFS é usado por estações que estão operando sob o DCF para transmitir *data frames* (MPDUs) e administrar *frames* (MMPDUs).

- *Extended* IFS (EIFS) O EIFS é usado por estações sob DCF sempre que a camada física PHY indicar para o MAC que a transmissão do *frame* foi iniciada e que não resultou em recepção correta. O intervalo EIFS começa seguido pela indicação da PHY que o meio está ocioso depois da detecção do erro no *frame* recebido sem considerar o mecanismo do *carrier-sense* virtual. O EIFS provê tempo suficiente para a outra estação reconhecer que recebeu um *frame* com erro antes de iniciar outra transmissão.

2.3.2.6 *Backoff Time* Randômico

Uma estação, antes de começar a transmitir, evoca o mecanismo de *carrier-sense* para definir se o meio está ocioso ou ocupado. Se o meio estiver ocupado, a estação deve ficar em silêncio até que o meio seja determinado como ocioso sem interrupção por um período de tempo igual a DIFS quando o último *frame* detectado foi recebido corretamente ou quando o meio seja determinado como ocioso sem interrupção por período de tempo igual ao EIFS quando o último *frame* detectado pelo meio não foi recebido corretamente.

Depois desse período, a estação deve gerar, de forma randômica, um período de tempo adicional chamado *backoff*, no qual a estação deve permanecer em silêncio sem tentar acesso ao meio, a menos que o *backoff* já tenha valor diferente de zero, onde a geração randômica não é necessária.

BackoffTime = Random() x aSlotTime

Random() = Inteiro pseudo-randômico distribuído no intervalo [0,CW], onde $aCWMin \leq CW \leq aCWmax$

aSlotTime = Valor que depende das características PHY

A janela de contenção (CW) pega valor inicial de aCWmim. Cada estação deve manter dois contadores: um STA *Short retry count* (SSRC) e um STA *long retry count* (SLRC) iniciados com valor zero. O SSRC deve ser incrementado sempre que qualquer contagem de tentativa curta associada com qualquer MSDU é incrementada, assim como o SLRC deve ser incrementado sempre que qualquer contagem de tentativa longa for associada com qualquer MSDU é incrementado. O CW deve pegar o próximo valor no intervalo [aCWmin,aCWmax] toda vez que a estação não tiver sucesso na tentativa de transmissão e, conseqüentemente, os contadores de tentativas SSRC SLRC também são incrementados. Esse processo avança até que o CW alcance o valor de aCWmax. Uma vez alcançado o valor *aCWmax*, a janela de contenção deve manter este valor até que seja reiniciado.

O CW é reiniciado para o *aCWmim* sempre que ocorra sucesso na tentativa de transmissão de um MSDU ou MMPDU, quando o contador SLRC alcança o valor do parâmetro *aLongRetryLimit*, ou quando o contador SSRC alcançar o valor do parâmetro *dot11ShortRetryLimit*. O SSRC é configurado para zero sempre que um *frame* CTS é recebido em resposta de um *frame* RTS, sempre que um ACK *frame* é recebido em resposta a transmissão de um MPDU ou MMPDU. O contador SLRC é configurado para zero sempre que um ACK *frame* for recebido em resposta de um MPDU ou MMPDU de tamanho maior que o *dot11RTSThresold*.

A janela de contenção em cada incremento tem seu intervalo alterado em função de uma potência de 2 menos 1, onde o *aCWmin* e o *aCWmax* é definido pelo PHY.

2.3.3 IEEE 802.11 Sob os Aspectos da Confiança

Em um sistema de confiança, em que a camada de enlace deve ser observada, vários aspectos podem indicar ou não o mal comportamento das estações envolvidas. Uma estação pode estar transmitindo vários *frames* após um SIFS para tentar transmitir um MSDU uma vez que o tempo de transmissão não foi suficiente, mas por outro lado pode ser um mal comportamento querer ganhar o acesso ao meio por mais tempo que o devido. Uma maneira de detectar esse suposto mal comportamento seria analisar se os *frames* que são transmitidos fazem parte do mesmo MSDU.

Cada *frame* de RTS/CTS informa o tempo em que as estações devem ficar sem acessar o meio, isso pode ser usado para medir o tempo em que determinada estação ocupa o meio, de forma a detectar tomada de forma egoísta.

A seguir, serão vistos os protocolos de roteamento, responsáveis por estabelecer as rotas em uma rede *ad hoc*.

2.4 Protocolos de Roteamento

As redes *ad hoc*, devido as suas particularidades, onde cada nó é um roteador em potencial, precisam estabelecer métodos para que seja possível encontrar rotas entre seus membros. Para isso, são estabelecidos protocolos que regem a forma como estas rotas são descobertas e organizadas em tabelas. Cada nó mantém em seu poder essas tabelas que são geradas e atualizadas de acordo com o tipo de protocolo adotado. Esses protocolos são classificados em dois tipos: proativo e reativo.

Tabela 2.1: Definição dos Protocolos de Roteamento em Redes *Ad Hoc*

Tipo	Descrição
Proativo	Cada nó mantém tabela com rotas para todos os membros da rede. Essas rotas são atualizadas em intervalo de tempos periódicos. A vantagem deste tipo de abordagem é que assim que a rota e necessária, a mesma já está disponível sem a necessidade de buscá-la. Em redes com grande mobilidade, este tipo de abordagem torna-se desvantajosa uma vez que a atualização das rotas gera tráfego considerável.
Reativo	Também conhecido como *on-demand*, as rotas são descobertas apenas quando são exigidas, isto é, as tabelas de rotas são geradas e atualizadas à medida que as rotas são necessárias. Essas rotas são descobertas por processo chamado descobrimento de rotas (*Route Discovery*). A vantagem é a redução do fluxo na rede (*overhad*), diferentemente da abordagem proativa, mas este benefício só se mantém se a rede for de baixa para média mobilidade.

- **Protocolos Proativos**

  OLSR - *Optimized Link State Routing*
  DSDV - *Destination-Sequenced Distance Vector Routing*

- **Protocolos Reativos**

  DSR - *Dynamic Source Routing*
  AODV - *Ad Hoc On-Demand Distance Vector Routing*

Dos diversos protocolos de roteamento existentes, serão tratados apenas os reativos, também chamados de *on-demand*, uma vez que estes impõem menor custo de comunicação. Além disso, esses protocolos possuem extensões que visam a estabelecer segurança de comunicação entre os nós da

rede.

2.4.1 AODV - *Ad Hoc On-demand Distance Vector Routing*

O AODV [7] define que as estações, a princípio, não precisam ter em suas tabelas todas as rotas existentes. As rotas são estabelecidas apenas quando uma estação necessita alcançar a outra, na qual não exista rota em sua tabela, isto é, quando uma estação deseja enviar um pacote a outra, primeiro ela verifica em sua tabela de rotas, se há entrada referente a estação destino. Se houver, o pacote é enviado; caso contrário, é efetuado o procedimento de descoberta dessa rota.

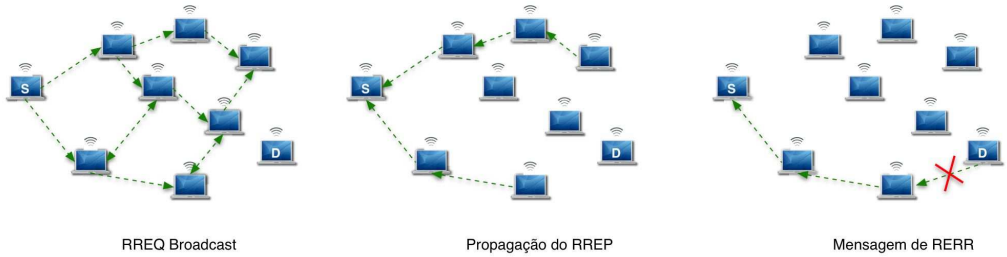

Figura 2.6: Funcionamento do AODV [7]

Esse procedimento inicia-se com o envio de pacote *Route Request* (RREQ), que contém o IP da origem e do destino, incluindo contador de saltos, que sai da estação origem com o valor zero atribuído e um RREQD ID usado para identificar os pacotes de RREQ e evitar duplicações. Ao receber um RREQ, um vizinho gera uma *reverse route* que futuramente será utilizada no estabelecimento da rota. Depois disto, ele verifica em sua tabela se há uma rota já estabelecida; se houver, ele utiliza a informação de rota reversa e responde a estação origem. Se não, ele incrementa o contador de saltos do RREQ e reenvia para o próximo vizinho. Quando o RREQ chega a uma estação que possui rota para a estação destino, primeiro é verificado se essa rota não expirou, senão um pacote *route replay* (RREP) é gerado, onde contém o IP de origem e destino, um contador de saltos e um número de seqüência do destino. Depois de gerado, o RREP é enviado à estação mais próxima que por sua vez gera nova entrada *forwoard route* para a estação destino e utiliza a entrada *reverse route* gerada quando do recebimento do RREQ, para enviar o RREP à estação origem. Mais detalhes podem ser vistos em [30, pp. 281 – 283].

2.4.2 DSR - *Dynamic Source Route*

O DSR[17] é muito semelhante ao AODV. No entanto, o mecanismo de descoberta de rotas difere na forma como a informação de rota é tratada pelas estações.

Inicialmente, ao notar que não há entrada em sua tabela para determinada rota, a estação origem envia um pacote de RREQ para o seu vizinho e esse verifica em suas tabela se há alguma entrada para a rota desejada, senão a estação vizinha atualiza a rota para a estação origem e adiciona no RREQ o seu IP reenviado para a próxima estação. À medida que o RREQ vai se propagando entre as estações, a rota fica pré-estabelecida de forma a construir o caminho até o destino. Quando uma estação que possui rota para o destino recebe o RREQ, ela concatena o caminho criado pelo RREQ com a rota para o destino e coloca-os em um RREP que é então transmitido em *unicast* para a origem. Já se a estação que recebe o RREQ é a estação destino, essa apenas gera o RREP com o caminho criado pelo RREQ e reenvia em *unicast* para a origem.

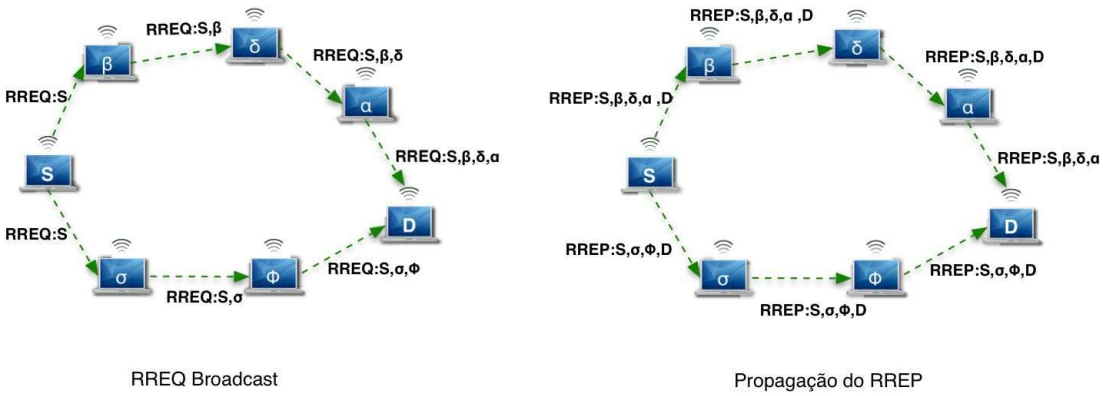

Figura 2.7: Funcionamento do DSR [17]

Tabela 2.2: Comparação entre os Protocolos DSR e AODV

Protocolos	Multirotas	Uso de *Flood*	Tempo de Vida	Prevenção de *Loops*
DSR		•	•	•
AODV	•	•		

Apesar de serem protocolos reativos e adotarem a mesma abordagem sobre como obter rotas, tanto o DSR quanto o AODV possuem particularidades que os tornam vantajosos em relação ao outro. A partir dessas diferenças, a seguir tem-se quadro comparativo das principais características entre os dois protocolos.

Há casos em que o DSR possui maior performance do que o AODV e vice-versa. Uma comparação mais detalhada do desempenho do dois protocolos pode ser vista no estudo do Perkins em [8].

Neste capítulo, foi estudado como são classificadas as redes sem fio, além do padrão IEEE 802.11 e seus protocolos de roteamento. A partir de agora, a atenção será focalizada nas redes sem infra-estrutura ou redes *ad hoc* e explorar quais são os pontos críticos neste tipo de rede. No próximo capítulo, serão tratados os principais problemas que podem ocorrer quando alguns parâmetros vistos são alterados ou manipulados gerando distorções no funcionamento da rede.

Capítulo 3

Identificando Distorções

A cooperação sempre foi um dos pilares que sustentam o convívio em sociedade. Conforme descrito no capítulo anterior, o conceito de sociedade e cooperação pode ser empregado facilmente quando se pensa em redes *ad hoc*. Em sociedade dita justa, todos os indivíduos vivem e trabalham com objetivo de igualdade entre os membros. Porém, no mundo real, sabe-se que isso não ocorre, sempre há alguns que querem levar vantagem sobre os outros. Frente a essa tendência natural, foram criadas leis que visam a tornar a igualdade mais justa entre os membros de uma sociedade.

Em redes *ad hoc*, assim como no convívio em sociedade, as coisas devem ser justas para o seu perfeito funcionamento. Para isso foram criados os protocolos de comunicação que servem como as leis desta sociedade digital e, assim como na vida real, quando os membros desta sociedade não respeitam as leis propostas, o bem-estar de todos pode ficar prejudicado.

Neste capítulo, serão analisadas possíveis distorções que podem ocorrer causadas pela exploração de eventuais falhas ou alteração nos protocolos estabelecidos pelo padrão IEEE802.11, nas diferentes camadas do modelo TCP e as soluções propostas para saná-las.

3.1 Camadas de Aplicação e Transporte

Apesar de não serem tratadas pelo IEEE802.11, vamos falar das camadas de aplicação e transporte apenas para efeito de informação, além de avaliar onde podem influenciar no desempenho das redes *ad hoc*.

Nas camadas de aplicação, estão os protocolos que estabelecem as regras para o oferecimento de diversos serviços. Falhas nos protocolos da camada de aplicação podem gerar problemas que vão do roubo de senhas até o envio de dados corrompidos, além de ser possível fazer um nó ter comportamento egoísta usando protocolos P2P.

A camada de transporte em redes *ad hoc* é basicamente tratada com redes cabeadas, ou seja, os programas entram em contato com a camada de aplicação pelos seus protocolos e estes, por sua vez, entregam os dados para a camada de transporte, que codifica estes dados no formato de pacotes, e

adicionam cabeçalhos de acordo com o protocolo da camada de transporte adotado.

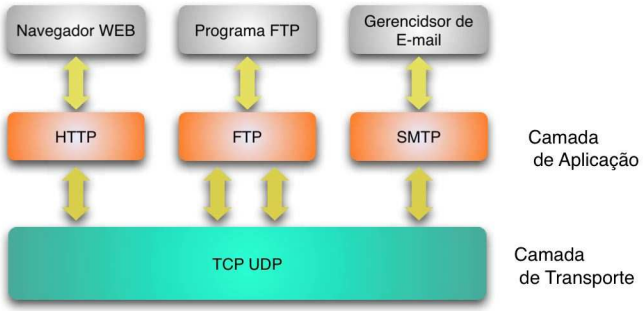

Figura 3.1: Camada de Transporte

O TCP é um protocolo orientado à conexão, no entanto, não foi projetado para redes móveis e conseqüentemente a sua característica principal, que é efetuar controle sobre a comunicação entre nós de uma rede, fica prejudicada devido a mobilidade dos mesmos. O TCP de um nó transmissor sempre que envia um pacote pela rede espera o recebimento de um ACK como resposta. Esta mensagem ACK informa um número seqüencial de 32 bits referente ao próximo pacote a ser recebido[5]. Se a mensagem ACK não for retornada após período de tempo pré-definido, o TCP do nó transmissor entende que o pacote foi perdido e transmite novamente. Em uma rede *ad hoc*, onde há grande mobilidade, a probabilidade que determinado nó não responda as mensagens de ACK é considerável.

Existem algumas soluções propostas com o objetivo de sanar estas e outras deficiências do modelo TCP 2.3 em redes *ad hoc*. Exemplo desta tentativa de mudança é o ATCP -*Ad hoc TCP* [16], onde os autores a criação de uma subcamada entre a camada de transporte e a camada internet.

A idéia do ATCP é monitorar a rede e informar por meio de mensagens ECN -*explicit congestion notification* e ICMP *"Destination Unreachable"* ao TCP do nó transmissor onde este fica em estado de espera até que nova rota seja definida. O conceito utilizado visa não a modificar o TCP para garantir sua compatibilidade com o padrão já existente das redes, mas adicionar novas funcionalidades.

3.2 Camada de Rede

A principal tarefa da camada de rede é fornecer meios para garantir que as informações geradas pela camada de transporte sejam enviadas por meio de rotas até um nó destino. Essas rotas podem ser definidas por meio de circuitos virtuais ou por datagramas [5].

Os circuitos virtuais partem da idéia da não necessidade da descoberta de nova rota toda vez que um dado for transmitido, ou seja, a rota fica definida e é a mesma para todos os pacotes da conexão estabelecida. O circuito

virtual fica ativo até que a transmissão termine.

Os datagramas, ao contrário dos circuitos virtuais, não são orientados à conexão, onde cada transmissão tem seus pacotes roteados independentemente, isto é, os pacotes podem trafegar por rotas distintas ou não. Roteamento por datagramas tem a vantagem de tratar congestionamento e é mais fácil lidar com eventuais quebras de rotas.

Redes heterogêneas e complexas, como a rede mundial de computadores (internet), onde a probabilidade de quebra de rotas é considerável, utilizam roteamento por circuitos virtuais, o que deixaria a rede muito vulnerável, uma vez que havendo quebra de algum roteador no meio do caminho toda a rota se perde.

As redes *ad hoc*, devido a sua característica móvel, necessitam de meios eficientes para definição de suas rotas. É natural perceber que roteamentos por meio de circuitos virtuais não seriam muito indicados para esse tipo de rede devido a fragilidade de suas rotas. Assim, como a internet, as redes *ad hoc* adotam o protocolo IP - *Internet Protocolo* para estabelecer conexão entre seus nós. O protocolo IP define a forma como os dados são transmitidos entre dois nós de uma rede, organizando-os em datagramas. Mas para que o IP possa realizar esta transmissão de dados, ele precisa saber o endereço dos nós de origem e destino. Para isso, são definidos os protocolos de roteamento.

Na subseção 3.2.1, serão tratadas as possíveis fraquezas encontradas nos protocolos de roteamento das redes *ad hoc*, assim como suas classificações. Na subseção 3.4, vamos tratar das soluções propostas por alguns protocolos, frente aos problemas apresentados.

3.2.1 Protocolos de Roteamento

Os protocolos que estabelecem rotas em redes cabeadas têm sua funcionalidade prejudicada, se adotados de forma direta em uma rede *ad hoc*, devido a sua mobilidade. Em uma rede *ad hoc*, cada nó se comporta como um roteador, ou seja, ele é responsável por repassar aos pacotes destinados outro membro da rede onde o caminho para tal membro passe por ele. Além, dependendo do protocolo adotado, manter tabelas com suas possíveis rotas. As rotas, no entanto, são compartilhadas com a rede tornando possível um nó encontrar um caminho para qualquer membro conectado, mesmo que este esteja a vários saltos de distância do transmissor. A figura 3.2 exemplifica bem essas situações.

Um ponto fundamental nos protocolos é a cooperação entre os nós. Há situações em que esta cooperação simplesmente não é possível, seja por motivos justos, como bateria, interferência ou simplesmente por mal comportamento como descarte intencional de pacotes ou outros distúrbios que podem ser causados de forma proposital. Esses distúrbios ou fraquezas nos protocolos de roteamento são explorados por meio de ataques [30] que podem ser classificados em:

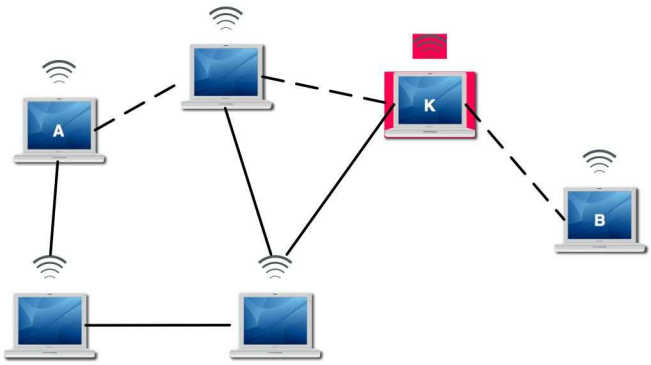

Figura 3.2: Roteamento com Falhas de Cooperação

- **Ativos** - ocorre quando há gasto de energia para explorar a fraqueza do protocolo. Alteração de tabela de rotas com objetivo de confundir outros nós é um exemplo deste tipo de ataque.

- **Passivo** - ocorre quando o agente causador não altera os dados explorados, isto é, não há gasto de energia. Estse tipo de ataque é caracterizado principalmente pela falta de cooperação, o que pode trazer sérios prejuízos em redes *ad hoc*.

Por meio dos problemas relacionados na tabela 3.1, pesquisadores [23, 19, 31] propuseram diversos melhoramentos nos protocolos de roteamento até então existentes, gerando gama de novos protocolos. Se observadas as falhas exploradas, nota-se que boa parte delas podem ser resolvidas trabalhando conceitos de cooperação. Visando isto, a maioria das soluções propostas seguem nessa vertente. Na subseção 3.4, veremos os principais protocolos de roteamento que tentam resolver os problemas relatados, usando relação de confiança ou recompensa, e na subseção 3.5 veremos outras soluções propostas baseadas no uso de métodos criptográficos.

3.3 Camada de Enlace

Estudamos no capítulo anterior que a camada *host to network* do modelo TCP é definida por padrões criados pelo IEEE [15], que estabelece duas camadas, uma de enlace e outra física. A camada de enlace se divide em duas subcamadas LLC e de acesso ao meio, também chamada de subcamada MAC. A figura 3.3 mostra como esta divisão é feita.

Com exceção da subcamada LLC, a subcamada MAC, juntamente com a física, são regulamentadas por padrões que variam conforme a rede adotada. No caso das redes sem fio, o padrão é o IEEE 802.11. Este define como se dá todo o processo de compartilhamento do meio assim como as características técnicas que regem as transmissões de rádio envolvidas. O objetivo desta seção é avaliar as características da subcamada MAC que podem sofrer alterações por parte dos membros de uma rede *ad hoc*.

Tabela 3.1: Principais Fraquezas dos Protocolos de Roteamento em Redes *Ad Hoc* [30]

Ataques	Descrição
Modificação	Para que seja possível estabelecer rota, os protocolos utilizam troca de mensagens entre os nós. Alterações nas mensagens trocadas podem gerar rotas inexistentes ou descarte indevido de pacotes transmitidos, causando negação de serviços - DoS *Deny of Service*.
Personificação	A personificação ocorre quando um nó se passa por outro. Como os protocolos de roteamento não tratam a autenticidade dos nós fica fácil falsificar os endereços de origem de determinado pacote, causando *spoofing*. Este tipo de ataque causa sérios problemas para a topologia da rede, uma vez que pacotes seguem por rotas erradas ou são passados para nós inexistentes.
Fabricação	Os protocolos de roteamento dependem da troca de mensagens para estabelecer suas tabelas de rotas. A geração de falsas mensagens de rotas, faz com que estes protocolos estabeleçam rotas inexistentes. Esse tipo de ataque é difícil de ser identificado uma vez que os nós têm processamento local e não podem inferir o que ocorre no restante da rede.
Buraco de verme - *Wormhole*	Este tipo de ataque é causado pelo conluio de dois nós maliciosos, gerando entre eles túnel capaz de fazer troca de mensagens de roteamento, causando cancelamento de determinadas rotas.
Cooperação	Com o objetivo de economizar bateria, um nó pode agir de maneira egoísta rejeitando solicitações de troca de mensagens para a construção de rotas, podendo ainda não rotear pacotes. Esse tipo de ataque passivo é difícil de ser detectado, pois o nó pode estar com problemas e não necessariamente com atitudes maliciosas.

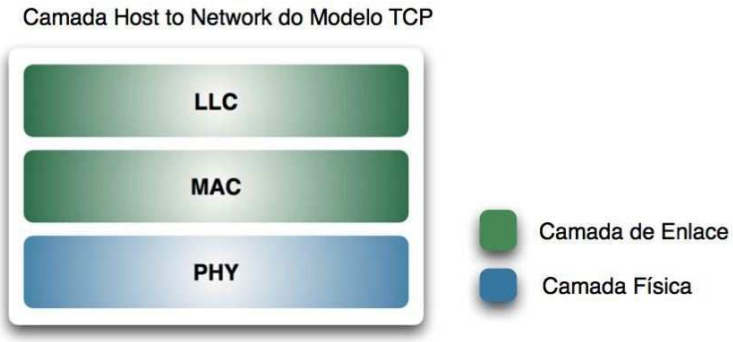

Figura 3.3: Estrutura Proposta pelo IEEE

3.3.1 Subcamada MAC

A função principal da subcamada MAC é definir regras para que o processo de acesso ao meio ocorra de forma justa e com o menor número de colisões possíveis. O padrão IEEE 802.11 estabelece que o processo de disputa de acesso seja feito utilizando o mecanismo de *backoff* discutido na seção 2.3.2.6. A seguir, veremos os principais problemas que podem ser causados por mal comportamento envolvendo processos da subcamada MAC.

3.3.1.1 Redução da Janela de Contenção

O processo de *backoff* exige que após uma transmissão com sucesso, escolha de forma aleatória em intervalo que varia de acordo com a janela de *backoff* máxima *CWmax*. Esse procedimento é feito para garantir que o acesso ao meio seja feito de forma justa. No entanto, um nó pode burlar esta janela máxima de forma a obter pequenos intervalos de *backoff*. Isso faz com que o nó malicioso ganhe acesso ao meio mais vezes que outros membros da rede. Na prática, esse método reduz drasticamente a capacidade de transmissão - *throughput*, o que pode ser entendido pela camada de aplicação como ataque de *DoS* [14]. No final deste capítulo, serão apresentadas algumas soluções propostas para resolver este problema.

3.3.1.2 Colisões Intencionais

Com objetivo de impedir a transmissão de outros membros da rede, um nó pode inserir pacotes de forma aleatória, causando colisões intencionais. Essas colisões fazem com que os nós dobrem suas janelas de contenção no procedimento de *backoff*, aumentando o seu tempo de decaimento conseqüentemente diminuindo a taxa de transmissão. Nesses casos, o objetivo do nó causador não é obter vantagens com acesso à rede, mas apenas causar distúrbios intencionais. Esse tipo de ataque também ocorre com o conluio entre dois nós maliciosos, injetando fluxos contínuos de pacotes com a intenção de parar as transmissões na área de atuação dos mesmos [14].

3.3.1.3 NAV Estendido

Ao iniciar o processo de transmissão, um nó informa em seu RTS um tempo maior que o necessário para a transmissão. Isso irá forçar os demais nós a aguardar mais tempo para voltar a decair o *backoff*. Monitorando o meio, esse ataque pode ser facilmente detectado, uma vez que o NAV estabelecido maliciosamente deixará os nós parados enquanto o meio estará ocioso.

3.3.1.4 *Timeout* Intensional

Alterações no tempo SIFS podem causar distúrbios no estabelecimento de rotas, a fim de deixar setores da rede com pouco tráfego e, assim, aumentar a taxa de transmissão do nó mal comportado [13].

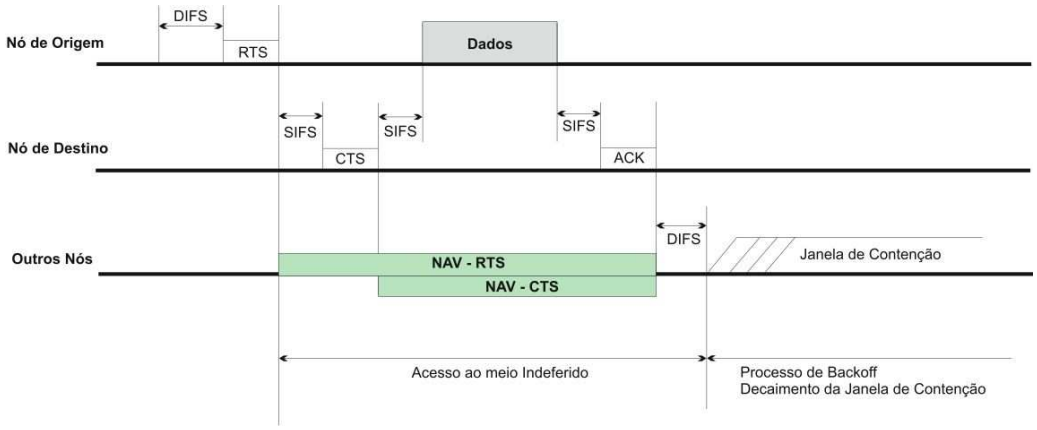

Figura 3.4: Procedimento de Acesso ao Meio [15]

A figura 3.4 mostra o procedimento que ocorre quando um nó necessita trocar informações com outro. Quando um nó começa a transmissão, inicialmente verifica se o meio está ocioso; se positivo ele envia mensagem RTS e aguarda o retorno do CTS durante determinado tempo TO_{CTS}, depois do qual o nó de origem entende que houve perda do pacote ou suposta colisão e tenta novamente outra transmissão do RTS. Supondo que houve o envio do CTS por parte do nó destino, o mesmo irá aguardar determinado tempo TO_{DATA}. Se os pacotes de dados não chegarem, o nó de destino entenderá que o nó origem não recebeu a mensagem de CTS. Analogamente, se os pacotes de dados forem transmitidos, o nó de origem aguarda o tempo TO_{ACK} para o retorno do ACK, se o mesmo não for retornado o nó de origem entende que o pacote de dados não foi recebido e tenta outro procedimento de *backoff*, para o envio do dado. Os tempos limites - *timeout* - são calculados de acordo com os seguintes parâmetros [13]:

$$TO_{CTS} = T_{RTS} + 2\delta + SIFS + T_{CTS} \quad (3.1)$$

$$TO_{DATA} = T_{CTS} + 2\delta - 2 \cdot SIFS + rf + T_{ACK} \quad (3.2)$$

$$TO_{ACK} = T_{DATA} + 2\delta + SIFS + T_{ACK} \quad (3.3)$$

Os valores T_{RTS}, T_{CTS} e T_{DATA} são os tempos necessários para a transmissão dos *frames* RTS,CTS e DATA, respectivamente. O valor δ é o atraso máximo de propagação do sinal no meio. O parâmetro rf corresponde ao tempo informado pelo cabeçalho do RTS, e pode ser calculado com a seguinte equação:

$$rf = 3.SIFS + T_{CTS} + T_{DATA} + T_{ACK}$$

Se o nó de destino aumentar o tempo SIFS, o que seria aproximadamente mais que 10% de um *time slot*, isso será suficiente para que o nó de origem esgote o tempo TO_{CTS} - *timeout*. O nó destino, no entanto, responde a requisição do CTS, mas com o tempo estourado. Depois de período de tempo sem sucesso, é repassado para a camada de rede que o *link* está quebrado, forçando a busca por outra rota. O objetivo do nó mal comportado é garantir que o tráfego da rede passe por outro caminho, garantido assim mais acesso ao meio, sem com isso alterar a sua janela de contenção no processo de *backoff*. Essa técnica pode parecer complicada, mas a idéia é parecer que o nó está sofrendo de algum tipo de interferência, dificultando a sua detecção por parte dos sistemas de monitoramento do meio, o que torna este tipo de ataque de difícil detecção. O artigo que descreve este ataque propõe sistema de monitoramento que leva em consideração essas alterações [13].

Existem outros distúrbios que podem ser vistos em [14, 13, 12, 25], que também tratam em mais detalhes os problemas relatados nas subseções anteriores.

Na próxima seção, serão vistas algumas soluções para camada de rede, onde são propostos alguns protocolos que atacam principalmente problemas de falta de cooperação no estabelecimento de rotas. Na camada MAC veremos solução para o problema de corrupção do processo de *backoff* denominado DOMINO[28].

Várias pesquisas foram desenvolvidas buscando o aumento de performance das redes sem fio. Os estudos que veremos a seguir são focados mais precisamente nas redes *ad hoc*, onde os métodos de estabelecimento de comunicação são mais difíceis se comparados com redes mais estáticas como as cabeadas.

3.4 Roteamento Baseado em Confiança

A cooperação dos nós é o que os protocolos de roteamento baseado em confiança procuram estabelecer. Para isso, usam análise de reputação ou método de recompensa, como é o caso do *Nuglets Nuglets*. A maioria dos protocolos a seguir foram evoluções de protocolos já existentes como DSR e o AODV.

3.4.1 CORE

Em [23], os autores Michiardi *et al.* propuseram o CORE, que utiliza técnica de monitorização cooperativa, em que as estações utilizam mecanismos de reputação para medir o nível de contribuição. A descoberta de rotas pelo protocolo DSR, serve de exemplo da utilização do CORE. Neste caso, uma estação, ao enviar um RREQ, monitora o comportamento do seu vizinho, se o mesmo irá responder com RREQ, RREP ou simplesmente não responder. Se a resposta do vizinho for a esperada, um valor positivo é atribuído, caso contrário receberá valor negativo. Utilizando métodos de avaliação, cada estação é capaz de avaliar, mediante as observações do comportamento de seus vizinhos, o nível de reputação de cada um.

3.4.2 *Nuglets*

Baseados na idéia de retribuição pela cooperação na rede, L. Buttyán *et al.* propuseram, em [19], espécie de moeda virtual chamada de *nuglets*, onde cada estação tem de pagar ou cobrar para utilizar a rede. Sendo assim, cada estação possui *nuglets counter* responsáveis por contar quantos *nuglets* cada uma tem. Um *hardware* confiável e resistente à falsificação definido como *security module* fica responsável por controlar os *nuglets*, os quais ficam protegidos por encriptação. Neste artigo, dois modelos são definidos o *Package Purse Model* e o *Package Trade Model*. No primeiro, a estação origem precisa saber previamente quanto *nuglets* serão necessários para que o pacote seja roteado, já que cada estação no meio do caminho irá cobrar pelo repasse do pacote. No segundo modelo, a estação origem não precisa saber quantos *nuglets* precisa ter para enviar o pacote, uma vez que as estações intermediárias irão negociar o repasse do pacote, cabendo a estação destino pagar por ele. O privilégio do *Package Purse Model* é que não é vantajoso o envio desnecessário de pacotes.

3.4.3 CONFIDANT

O CONFIDANT, proposto por S. Buchegger *et al.* [31], é baseado no protocolo DSR. O CONFIDANT trabalha como extensão para um protocolo de roteamento e é composto por quatro módulos: *Monitor, Reputation System, Path Manager e Trust Manager*. O monitor escuta o tráfego da estação vizinha e observa o comportamento. Ele mantém cópia do pacote enquanto

escuta a transmissão pelo próximo nó, isto é, se houver alguma mudança no pacote será percebido. O *Reputation System* é responsável por manter tabela com o *ranking* de reputação das estações da rede. Ao ser constada a existência de estação maliciosa, o *Path Manager* é responsável por remover tais rotas da tabela de roteamento, além de controlar e ajustar a tabela de *ranking* de reputação entre as estações, de acordo com métricas de segurança estabelecida. Já quando estação é definida como maliciosa é enviada mensagem ALARM para o *Trust Manager*. Cada estação que achar que um nó é malicioso pode enviar um ALARM às outras, no entanto, ao receber um ALARM uma estação pode atribuir isso ao *ranking* e passa a verificar se este ALARM não é alarme falso. A dificuldade de se verificar se uma mensagem de ALARM é errada ou não é tratado usando-se estatísticas Baysianas. Cada estação, em uma rede com CONFIDANT, possui todos os quatro módulos acima referenciados.

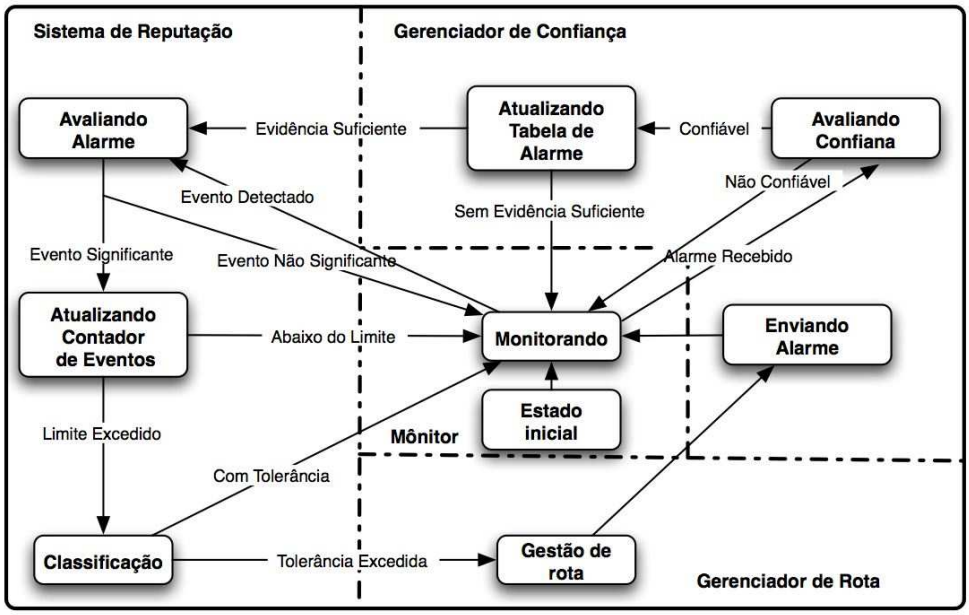

Figura 3.5: Arquitetura do Sistema de Confiança do CONFIDANT

3.5 Roteamento Baseados em Chaves Criptográficas

Os protocolos que utilizam chaves criptográficas tem por objetivo não apenas verificar a identidade dos nós, mas a integridade das informações transmitidas. Estes protocolos, por sua vez, não tratam a forma como as chaves são distribuídas, o que pode ir contra à idéia de redes *ad hoc* que não adotam mecanismos centralizados.

3.5.0.1 *Secure Routing Protocol* - SRP

Existem outras soluções para a questão da confiança entre estações em redes *ad hoc* que levam em conta o uso de criptografia. Proposto por Papadimitratos *et al.*, o SRP [24] tem por objetivo principal garantir a autenticidade das partes envolvidas no processo de definição das rotas. Compartilhando chave secreta, as estações são capazes de estabelecer confiança entre si. Para isso, quando é necessário descobrir nova rota, a estação origem envia o pacote à estação destino juntamente com o *Menssage Authenticator Code* (MAC) do pacote. As estações intermediárias repassam o pacote até o destino que calcula novamente o MAC com o uso da chave compartilhada e verifica a autenticidade do pacote. Outros detalhes mais complexos serão omitidos, mas podem ser vistos detalhadamente em [24].

3.5.0.2 ARIADNE

O ARIADNE [40], proposto por Y. Hu *et al.*, é baseado na idéia de autenticação usando chaves, que no caso do ARIADNE é simétrica, o que o torna vantajoso em relação a chaves assimétricas que exigem alto poder computacional em relação às chaves simétricas. A idéia principal dessa solução é a autenticação em cada estação da rede por onde o pacote pode passar. O protocolo de autenticação *broadcast* TESLA [40] é utilizado como solução de autenticação adotada pelo ARIADNE.

3.5.0.3 ARAN

Com o objetivo de estabelecer solução que fosse composta de autenticação, mecanismo de integridade de mensagem e não repudiação, o ARAN [33] foi proposto por K. Sanzgiri *et al.* O funcionamento do ARAN depende da existência de servidor que deve prover certificados a todas as estações que compõem a rede. Esses certificados devem possuir o IP da estação assim como chave pública e selo de tempo. Este é que irá determinar a expiração do certificado. Todos os nós deverão manter seus certificados atualizados com o servidor. Assim, é possível obter rota checando a autenticidade dos pacotes enviados e estabelecer confiança entre as estações.

Muitas das tecnologias mostradas neste capítulo existem apenas no campo científico e são objeto de pesquisas ainda não implementadas. Isso nos dá breve visão do quanto as redes sem fio podem evoluir.

3.6 Soluções para a Subcamada MAC

Vamos detalhar duas soluções diferentes para o problema de alteração da janela de contenção, com objetivos de ganhar largura de banda.

3.6.1 DOMINO

O autor [28] propõe modelo de provedor com AP confiável e simples. Considera que os usuários são mal comportados para obter benefícios da rede e não causar falha [28].

3.6.1.1 Componentes do Domino

Os dados são monitorados em curtos intervalos de tempo chamados de *monitoring periods*. Todas as vezes que uma estação é considerada suspeita, um contador é incrementado e depois de um limite de vez pré-definido é ultrapassado, esta estação é considerada mal comportada. Toda vez que o meio é monitorado e uma estação não definida como suspeita, o seu contador é decrementado até chegar a zero. Isso é útil para que uma estação se torne confiável novamente.

- O primeiro teste que o DOMINO [28] usa é o *Scrambled Frames*, esse teste se baseia em quantas vezes uma estação transmite. Ele monitora a quantidade de CTS/ACK/DATA, para determinar quantas vezes uma estação transmitiu. Como o uso da banda é compartilhada, uma estação que muito transmite impede que as outras façam a mesma coisa. Um parâmetro de tolerância é adicionado a fim de dar margem de segurança e evitar falsos positivos. Esta variável de tolerância é de 0 a 1.

- O segundo teste do DOMINO leva em consideração as possíveis alterações nos parâmetros do protocolo. Analisa a manipulação do período DIFS, escuta a última vez que o ACk foi enviado e verifica o seu tempo de duração. Se este tempo for menor que o DIFS normal, ele computa como suspeito.

- O terceiro teste que ele monitora é o tamanho do NAV. Para isso, a informação transmitida é verificada com valor do NAV estabelecido com o valor RTS e DATA e detecta se a estação está utilizando uma NAV maior ou não. Neste teste, há variável de tolerância "A"que tem valor maior que 1.

- O quarto teste envolve parâmetros do *backoff* e verifica qual é o maior *backoff* que uma estação usa. Para isso, divide o valor máximo do *backoff* e divide por dois. Se a estação monitorada tiver no período observado, o seu maior *backoff* menor que o parâmetro estipulado $\frac{(CW-1)}{2}$, a estação é computada como suspeita.

- O quinto teste do DOMINO verifica o valor do *backoff* "atual". Para monitorar este período de *backoff*, é observado o tempo gasto desde o último período DIFS até o próximo período DIFS de início de outra transmissão da estação monitorada. É claro que neste espaço de tempo devem ser computados os DIFS, SIFS ACK e os dados transmitidos por outras estações para determinar quando a janela de *backoff* decaiu.

- No entanto, esse teste não é eficiente quando uma estação usa interframes *delay*. Este tipo de transmissão é muito comum, principalmente quando o protocolo TCP faz controle de congestionamento. De acordo com outros artigos (6, 19), esse tipo de problema representa 91% do tráfego real de uma rede. Em se tratando de sistema de *hotspot* onde o tráfego é elevado, isso deve ser levado em consideração.

- O sexto e último teste visa a sanar a dificuldade de detectar fraudes no processo de *backoff* quando se tem interframes *delay* na transmissão de dados entre estações.

Imagine situação em que a estação monitora precisa transmitir e existe tráfego considerável no meio. Quando a estação começar a transmitir, em função do tráfego, ela começa a enfileirar pacotes na camada MAC. Nesta situação, a estação pode se beneficiar com o processo de *backoff*, transmitindo vários pacotes sem contudo deixar o meio ocioso para as outras estações. O processo de monitoramento é feito de forma que se houver dois *frames* não intercalados consecutivos, o DOMINO considera que houve período um ocioso e somente um período de *backoff*.

3.6.1.2 Simulação

As simulações apresentadas pelo autor basearam-se nas análises dos testes 5 e 6, ou seja, os que tratam basicamente do *backoff*, e foram feitas usando o simulador ns-2 com a extensão Monarch (13). Para montar o cenário, foi utilizado *Access Point* com oito estações. O tráfego UDP foi configurado para o tipo CBR a uma taxa de 500KB por pacote e 200 pacotes por segundo. Para o tráfego TCP, a aplicação escolhida foi o FTP, por ser aplicação mais fácil de monitorar. O processo de simulação foi feito com média de 10 testes com a duração de 110 segundos cada, sendo que o período de monitoramento foi de 10s.

A primeira análise que o autor faz com o cenário diz respeito ao impacto que estação mal comportada pode causar com a banda da rede. Inicialmente, foram colocados dois gráficos que mostram que a diferença entre o tráfego com e sem estação mal comportada, e pelos gráficos obtidos fica claro a perda com a manipulação do *backoff*. Outra conclusão que pode ser obtida pela análise das simulações é que dependendo do teste, *backoff* atual e *backoff* consecutivo os resultados são significativos ou não.

Para o teste do *backoff* atual, o autor chegou a conclusão pelos testes de simulação que pelo tráfego UDP o ganho de performance pela estação maliciosa é bem acentuado, o que foi constatado o contrário com o tráfego TCP, pelo fato que este tipo de tráfego conta com controle de congestionamento. Isso afeta severamente o uso do TCP para este tipo de ataque. Para o teste do *backoff* consecutivo, ao contrário do *backoff* atual, o tráfego UDP apresentou resultados muito semelhantes, com e sem estação maliciosa. Por este motivo, o teste envolvendo UDP foi desprezado pelo autor. Já o tráfego com TCP apresentou ganho considerável.

Para efeito de implementação, foi adotado o uso da análise dos dois testes, isto é, do *backoff* atual e consecutivo. Outro ponto constatado nas análises feitas, foi que com pouco tráfego não é possível estabelecer níveis confiáveis de conclusão. Considerando que com pouco tráfego não seria interessante para estação maliciosa atacar a rede, os testes serão desprezados quando constatado pouco tráfego.

3.6.1.3 A Implementação

Usando três *laptops*, dois como estação e um como monitor, o autor fez a implementação do DOMINO apenas alterando os *drivers* das placas *wireless*. A vantagem neste tipo de implementação é pelo fato de funcionar de forma passiva. No entanto, o artigo mostra que o DOMINO pode ser implementado no próprio *Access Point*. De acordo com o autor do artigo, o impacto de processamento é da ordem de 0,022%, ou seja, o processamento dos dados do DOMINO não afeta o desempenho normal do *access point*. Porém, a desvantagem da implementação no *access point*, seria a dificuldade de atualização do *firmware*.

Apesar de simples, há maneiras de burlar ou dificultar as técnicas que o DOMINO usa para fazer sua análise de dados. Porém, para que isso seja possível, o atacante precisa saber quais parâmetros os testes que envolvem o sistema de análise se baseia. Como isso, seria difícil de ser deduzido pelo atacante, o DOMINO se apóia neste ponto. Muitos dos cálculos disponíveis no Apêndice podem ser usados para propor melhorias nas idéias propostas pelo autor.

3.6.2 *Selfish MAC layer misbehavior in wireless networks*

Para solucionar o problema da janela de contenção do processo de *backoff*, o autor propõe alteração do IEEE 802.11 [18]. A idéia principal seria definir qual será o próximo *backoff* que um nó deverá praticar em transmissões subseqüentes. O funcionamento teoricamente é simples. Na primeira conexão, o nó transmissor define o seu *backoff* de acordo com o padrão sem alterações do IEEE 802.11; quando o nó receptor receber o RTS, imediatamente calcula nova janela de contenção e envia este novo valor na mensagem CTS. O valor também será enviado na mensagem ACK. Depois o nó transmissor é monitorado para verificar se irá ou não obedecer ao *backoff*

definido. Se houver transmissão antes do período de *backoff* definido, haverá grande probabilidade de se tratar de nó mal comportado.

O sistema é dividido em três esquemas distintos responsáveis pela detecção, diagnóstico e penalização:

- no primeiro esquema, o receptor decide, ao finalizar transmissão, se o transmissor cometeu algum desvio de conduta frente ao protocolo, monitorando seu *backoff*;

- no segundo esquema, se o transmissor foi identificado como mal comportado, ele será penalizado de acordo com o grau do seu desvio;

- no terceiro esquema, baseado no grau de maliciosidade cometida pelo transmissor, depois de várias transmissões.

No primeiro esquema, o nó receptor, depois de definir o *backoff* e enviá-lo pelo RTS ou CTS, monitora o meio para verificar se o transmissor de fato obedeceu ao *backoff* estipulado, mas podem haver interferências que podem gerar falso positivo em relação à maliciosidade do nó monitorado. Isso ocorre porque o nó que monitora pode sofrer interferências e, neste caso, pára de decair o *backoff*, porém o nó monitorado pode ser considerado malicioso injustamente. Para isso, o autor define um α que será a tolerância. Para exemplificar como a tolerância funciona, temos B_{act} como o *backoff* monitorado, e B_{exp} é o *backoff* esperado.

$$B_{act} < \alpha * B_{exp} \qquad 0 < \alpha \leq 1$$

Na solução, o autor simulou utilizando alpha=0,8 por encontar valores mais coerentes [22].

O autor também trata a situação quando o nó não responde a continuação de uma transmissão. Um contador é implementado no RTS, de forma que quando um RTS é transmitido o valor deste contador é 1, se houver colisões o transmissor incrementa este contador. Assim, quando um receptor receber o RTS do transmissor monitorado, ele pode inferir se o nó em questão estava sofrendo colisões intencionais, ou se ele estava simplesmente ignorando as transmissões para o nó que monitora. Como o nó receptor monitora o meio, escuta se o nó em questão está transmitindo com outros nós da rede. Se isso estiver ocorrendo, o nó é considerado malicioso.

Outro problema que pode ocorrer é quando há colisões. Neste caso, o contador deve ser incrementado, mas como o processo de *backoff* é um valor probabilístico, fica difícil estimar qual foi o real *backoff*, pois neste caso o nó monitorado deve dobrar a janela de contenção [22].

Neste capítulo, verificou-se que vários problemas podem ocorrer quando as regras definidas pelos padrões são quebradas. Analisamos as principais

pesquisas que exploram alguns desses problemas, onde procuramos dar visão simples de como as soluções são propostas. No próximo capítulo, vamos discutir o objetivo deste trabalho que visa a propor abordagem de sistema de Reputação baseado em *Cross-Layer*.

Capítulo 4

Abordagem Proposta

Os estudos mostrados no capítulo 3 trazem idéia das pesquisas desenvolvidas na busca por soluções aos diversos problemas das camadas de rede e enlace. As maiores contribuições que visam a soluções que consideram avaliação de comportamento, entretanto, são destinadas a camada de rede tendo poucos trabalhos destinados aos problemas existentes na camada de enlace.

Neste capítulo, trataremos do objetivo deste trabalho que avalia as fraquezas exploradas nos protocolos do padrão IEEE 802.11, resultando na proposição de abordagem para detecção de maliciosidade, baseado em métodos estatísticos e demonstrações matemáticas, que serão tratados nas seções a seguir.

4.1 Descrição dos Objetivos

Os problemas tratados na seção 3.3 mostram os principais distúrbios que podem ser causados no padrão IEEE 802.11, mais precisamente na camada de enlace.

Quando um nó transmissor inicia o processo de transmissão, é enviada ao nó receptor mensagem RTS que tem em seu cabeçalho campo informando quanto tempo irá durar a transmissão em questão. Baseado na informação desse campo do cabeçalho RTS, os outros nós sob o raio de atuação do sinal de transmissão devem estabelecer o *NAV* e, conseqüentemente, para o decaimento do *backoff* até que o NAV termine. Um nó malicioso com a intenção de diminuir a taxa de transmissão dos seus vizinhos pode informar pelo RTS ou CTS tempo maior que o necessário para transmissão em questão. Esse tipo de ataque pode ser monitorado facilmente observando o meio, uma vez que o NAV continuará ativo e o meio ocioso.

O processo de *backoff* definido pelo IEEE 802.11 tenta resolver a questão do acesso ao meio utilizando métodos probabilísticos na definição das janelas de contenção *CW*, ou seja, para que determinado nó tenha acesso ao meio ele deve escolher aleatoriamente valor no intervalo [0,CW_{min}]. Na subseção 2.3.2.6, vimos o processo de estabelecimento da janela de contenção, que neste trabalho terá a primeira janela no intervalo [0,15] até

[0,1023]. Agora, sob a ótica do compartilhamento de acesso ao meio, para que o mesmo seja distribuído de forma justa a probabilidade de acesso de cada nó deve ser $P = \frac{100}{n}$ onde n é a quantidade de nós compartilhando o mesmo meio. Se um nó intencionalmente alterar o seu intervalo de *backoff* para valores menores que CW_{min} da primeira janela de contenção, o mesmo passa a ter mais chances de obter acesso ao meio e, conseqüentemente, passa a transmitir mais que os outros nós. É notório que se um nó transmite mais, conseqüentemente outros ficarão sem transmitir ocasionando perda de desempenho da rede.

Para detectar esse problema, precisamos monitorar os intervalos de *backoff* praticado por um suspeito, no entanto, o monitoramento do processo de *backoff* não é procedimento simples. O artigo [28] estabelece meios para inferir o valor do *backoff* atual por meio de cálculos baseados em *Cadeias de Markov*, além de inferir *backoffs* passados por mensuração dos *TimeSlots* entra duas transmissões, conforme figura 4.1.

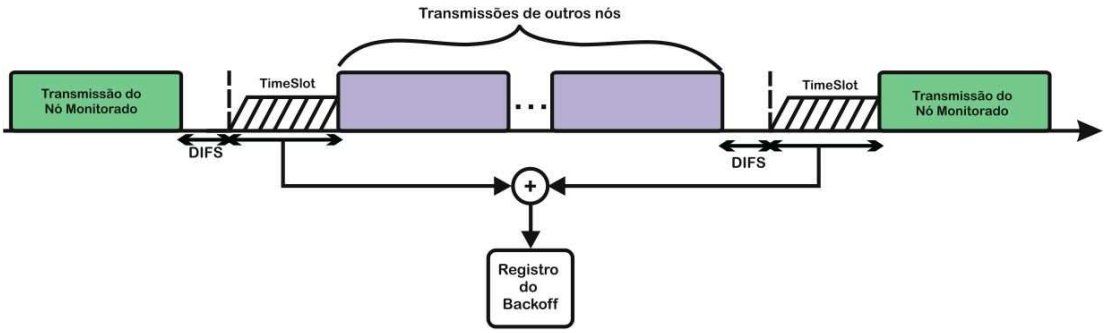

Figura 4.1: Mensuração do *Backoff*[28]

Suponha que um nó monitorado esteja sob suspeita de ser mal comportado, e o mesmo se localize na área de atuação de dois nós distintos, isto é, em ambiente de saltos múltiplos. Quando esse nó for monitorado pelo seu vizinho, o último pode estar em área onde não sofra inferência do sinal que atua sobre o nó monitorado. Logo, se houver parada do processo de decaimento da janela de contenção, isso não será percebido pelo vizinho que está monitorando. Nesse caso, o nó monitor registrará valor de *backoff* maior do que realmente foi executado. Em outra mão, se o nó que monitora sofrer interferência do meio vai paralisar o seu processo de decaimento da janela de contenção e não vai poder inferir se o nó monitorado também está sob influência do sinal em questão, logo vai registrar *backoff* maior que o executado pelo dispositivo monitorado, o que pode ocasionar falsos alertas de mal comportamento. Nesse caso, o método proposto por [28] não funcionaria corretamente uma vez que o sistema da figura 4.1 foi desenvolvido para redes *single-hops*.

Esses nós ou dispositivos possuem antenas ominidirecionais que têm área de atuação circular, logo uma avaliação das áreas de atuação do sinal de um nó pode mostrar a probabilidade de o mesmo estar ou não em área de atuação do sinal de vizinhos distintos. Com isso, o mecanismo proposto na figura 4.1 evolui para contribuir em modelo de detecção, que iremos pro-

por levando em consideração a possível localização do nó monitorado sem a utilização de GPS.

Nas próximas seções, veremos conceitos de definições matemáticas que servirão de base para mostrar os estudos estatísticos, assim como os geométricos que contribuíram para a construção do modelo que será proposto.

4.2 Conceitos de Definições

A seguir, veremos breve revisão de alguns conceitos estatísticos, que servirão de base para a fundamentação das teorias tratadas nos capítulos à frente.

4.2.1 Conceitos Estatísticos

A pesquisa na área de estatísticas teve como foco inicial buscar ferramentas que pudessem comprovar idéias em torno de processos aleatórios. Como o objetivo principal do trabalho envolve processos aleatórios de *backoff*, veio a necessidade de revisão teórica.

Conceitos iniciais, como espaço amostral, evento, evento elementar, evento certo, espaço amostral finito, entre outras definições são mostradas no trabalho do Bussab (2004) [37] no livro *Estatística Básica*.

Para que nossa pesquisa obtivesse êxito, o estudo da inferência estatística mostrou-se de fundamental importância. Por meio desse conceito, é possível realizar inferências a respeito da população (intervalos de *backoff*) partindo das amostras (monitoramento das janelas de *backoff*). Porém, a pesquisa em tal situação precisa ser aprofundada uma vez que não se sabe a quantidade exata de quantas observações serão necessárias para que tenhamos menor erro possível. A teoria do limite central foi outro assunto abordado no trabalho de Bussab. Outro autor que teve seu trabalho pesquisado foi o Grinstead (1997)[9], com a obra *Introduction to Probability*, que traz de forma mais profunda as principais provas dos conceitos que giram em torno das variáveis discretas aleatórias, o que contribui em muito na compreensão de algumas idéias.

Definição 1. *Espaço amostral é o conjunto de todos os resultados possíveis.*

$$\Omega = \{1, 2, 3, ..., 31, 32\}$$

Definição 2. *Quando temos um subconjunto do espaço amostral, chamamos de evento.*

Exemplo 1. *Em Ω temos o seguinte evento A*
Onde,

$$A = \{1, 3, 5, 7\}$$

Definição 3. *Quando temos um único elemento do espaço amostral, chamamos de Evento Elementar.*

$$\text{Evento Elementar } = \{16\}$$

Definição 4. *Quando um evento coincide com um espaço amostral, temos um evento certo.*

$$\text{Evento certo } \implies \Omega = A$$

Definição 5. *Quando temos um evento vazio, chamamos de Evento Impossível.*

$$A = \emptyset$$

Definição 6. *Dado um espaço amostral finito, temos $P(A)$ como a probabilidade de ocorrer o evento A.*

Onde,

$$P(A) = \frac{A}{\Omega} \quad , \ 0 \leq P(A) \leq 1$$

4.2.1.1 Valor Médio de uma Variável Aleatória

Definição 7. *Dado uma V.a.X discreta assumindo os valores $x_1, ..., x_n$ chamamos* VALOR MÉDIO *ou* ESPERANÇA MATEMÁTICA *de x ao valor.*

$$E(X) = \sum_{i=1}^{n} x_i P(X = x_i) = \sum_{i=1}^{n} x_i P_i$$

Definição 8. *Chamamos de* VARIÂNCIA DA V.A.X *o seguinte valor:*

$$Var(X) = \sum_{i=1}^{n} [x_i - E(X)]^2 \cdot P_i$$

A raiz quadrada positiva da variância é definida como sendo o *Desvio-Padrão* de $X, DP(X)$.

4.2.1.2 Distribuição Uniforme Discreta

Definição 9. *A v.a.X, assumindo os valores $x_1, ...x_n$, tem distribuição uniforme, se e somente se:*

$$P(X = x_i) = p(x_i) = p = \frac{1}{k}$$

Para todo i=1,2,3,...,k

É fácil verificar que,

$$E(X) = \frac{1}{k}\sum_{i=1}^{k} x_i \quad , \quad Varx(X) = \frac{1}{k}\{\sum x_i^2 - \frac{(\sum x_i)^2}{k}\}$$

A função de distribuição acumulada é dada por:

$$f(x) = \sum_{x_i \leq x} \frac{1}{k} = \frac{n(x)}{k}$$

Onde n(x) é o número de $x_i \leq x$.

Exemplo 2. *Suponha que um dado honesto seja lançado. Seja* X *a v.a que indica o número de pontos marcados na face do dado.*

Assim,

$$\begin{aligned} E(X) &= \frac{1}{6} \cdot (1+2+3+4+5+6) = \frac{21}{6} = 3,5 \\ Var(X) &= \frac{1}{6} \cdot ([1+2+3+4+5+6] - [\frac{[21]^2}{6}) = \frac{35}{12} = 2,9 \end{aligned}$$

Tabela 4.1: Distribuição das Probabilidades

x	1	2	3	4	5	6	TOTAL
$p(x)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1

4.3 Descrição da Abordagem Proposta

4.3.1 Estatística e Posicionamento

Como o acesso ao meio é método probabilístico, é natural que alguns valores oscilem. Nesse caso, quanto mais intervalos de *backoffs* analisados, mais precisa fica a decisão sobre a maliciosidade do nó monitorado. Logo, é fundamental saber quantos intervalos devem ser monitorados para que seja tomada uma decisão. Sabendo quantos intervalos devemos monitorar, temos, no entanto, que inferir a possível localização do nó monitorado uma vez que ele pode estar sob influência de outro sinal, causando monitoramentos equivocados. Para amenizar esses equívocos, podemos inferir esta

posição por meio das trocas de tabelas de roteamento de dois saltos além de usar avaliações geométricas.

A seguir, veremos como podemos encontrar quantos intervalos devemos monitorar e como inferir a localização de um nó realizando análise geométrica.

4.3.1.1 Análise Estatística

O processo de *backoff* implementado pelo IEEE 802.11 foi desenvolvido para que os nós que compartilham o mesmo canal pudessem compartilhar o meio de forma justa. Mas até que ponto podemos dizer que o processo de *backoff* está atuando de forma justa? Apesar do processo de *backoff* utilizar processo de sorteio aleatório, isso pode ser facilmente burlado dependendo da implementação do padrão.

Uma análise das probabilidades envolvidas pode mostrar se determinado nó pode ou não estar se comportando maliciosamente, ou seja, se o mecanismo de estabelecimento dos intervalos de *backoff* estão, de fato, de acordo com o padrão IEEE 802.11.

Inicialmente, vamos considerar que todos os nós estão operando de forma justa, isto é, estão no primeiro intervalo de *backoff* que podem variar dependendo da modulação adotada. Nesse caso, vamos adotar o intervalo de 0 a 15 que será então a população a ser considerada, onde temos 7,5 como média.

Os valores sorteados são variáveis aleatórias discretas, ou seja, a cada seleção a probabilidade de escolha das outras amostras não sofrem influências. Valores como variância e desvio-padrão podem ser encontrados usando as seguintes equações:

$$Var(X) = \sum_{x=0}^{15} \frac{(x - 7,5)^2}{16} = 21,25 \tag{4.1}$$

$$\sqrt{Var(X)} = \sqrt{\sum_{x=0}^{15} \frac{(x - 7,5)^2}{16}} \cong 4,61 \tag{4.2}$$

Para simular o cálculo acima, foram considerados a média, variância e o desvio-padrão de 100 baterias de testes de 1000 amostras de intervalos de *backoff*. Os resultados obtidos foram muito próximos do esperado.

A variância estabelecida na tabela 4.2 seria usada se todos os elementos do intervalo fossem escolhidos de forma igual. Mas, quando se pensa em médias, temos de nos aproximar ao máximo da média da população em questão, tornando necessário que variância seja diminuída. Mas qual o valor ideal para a variância? Para exemplificar a pergunta, qual seria a variância para que a probabilidade da média de uma amostra aleatória seja menor que 6,5? Usando a tabela Z , encontrada no anexo deste trabalho 5,

Tabela 4.2: 5 Simulações com Média de 100 baterias de 1000 amostras

Simulação	Variância	Desvio
1ª	21,36	4,62
2ª	21,10	4,59
3ª	21,34	4,62
4ª	21,48	4,63
5ª	21,24	4,61

podemos chegar a este valor.

De acordo com a tabela-Z, temos:

$0,01 \cong 0,5 - 0,49010 \rightarrow 2,33$

$$Var(X) = \frac{6,5 - 7,5}{-2,33} \cong 0,43$$

Logo, para que tenhamos a probabilidade de 1% que uma média aleatória seja menor que 6,5, teremos desvio-padrão de 0,65 e variância de 0,43. Veja que esses valores são bem diferentes dos encontrados acima. Aqui, fica fácil ver que podemos mensurar o grau de probabilidade de determinado dado esteja dentro do intervalo desejado.

Para sabermos se determinado nó está de fato malicioso ou não, temos de saber qual é o seu *backoff* médio. Para fazer melhor análise, vamos construir distribuição das médias dos intervalos de *backoffs*.
Sendo assim, com nossa população $\{0, 1, 2, ..., 15\}$, onde temos média $\mu = 7,5$ e variância $\sigma^2 = 21,25$, podemos inferir:

Teorema: Seja X uma variável aleatória com média μ e variância σ^2, e seja $(X_1, ..., X_n)$ uma AAS (Amostra Aleatória Simples) de X, então,

$$X(\overline{X}) = \mu \qquad e \qquad Var(\overline{X}) = \frac{\sigma^2}{n}$$

Observe que à medida que a quantidade da amostra vai aumentando a variância vai diminuindo, ou seja, podemos encontrar médias dentro de intervalo de confiança estabelecido.

Agora, que sabemos como fazer a distribuição das médias de *backoffs*, vamos analisar como se comporta essa distribuição de acordo com alguns parâmetros estabelecidos. Inicialmente, vamos tomar seqüência de 10, 50, 100 amostras aleatórias de 60, 125, 250 e 500 elementos do intervalo de *backoff* para comparar o comportamento da variância das médias e o desvio-padrão além das probabilidades envolvidas.

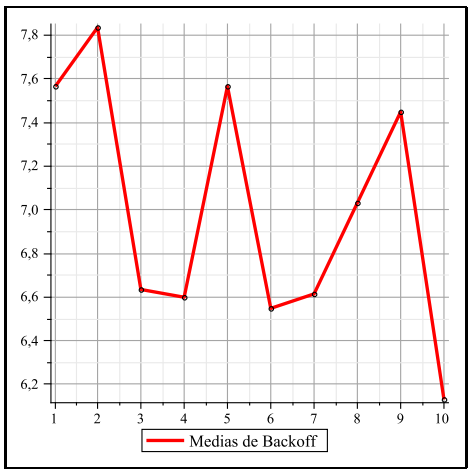

Figura 4.2: *Médias de 10 Amostras de 60 Elementos*

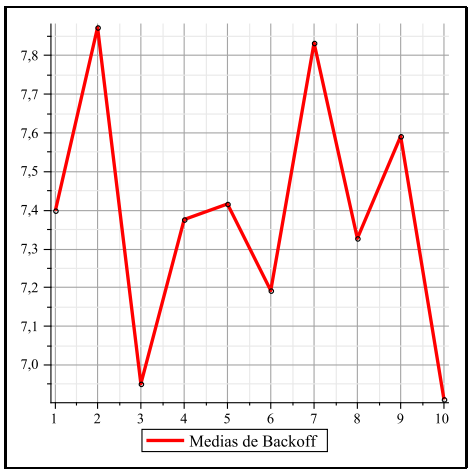

Figura 4.3: *Médias de 10 Amostras de 125 Elementos*

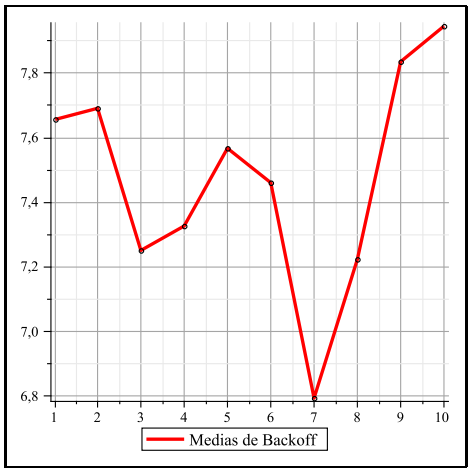

Figura 4.4: *Médias de 10 Amostras de 250 Elementos*

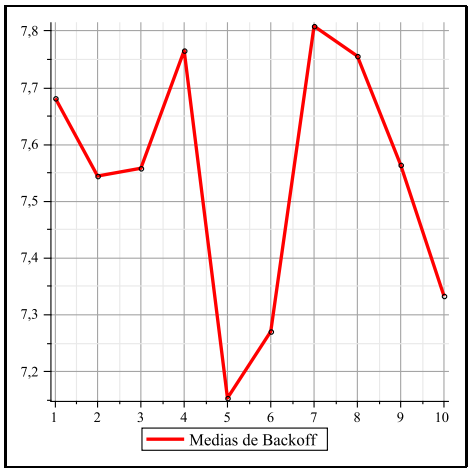

Figura 4.5: *Médias de 10 Amostras de 500 Elementos*

.

Tabela 4.3: Estatísticas da Distribuição das Médias de 10 Amostras

X_n	x_n	$\overline{X}$	σ^2	σ
10	60	6,998	0,3262	0,5711
10	125	7,387	0,1036	0,3219
10	250	7,476	0,1149	0,3390
10	500	7,544	0,05054	0,2248

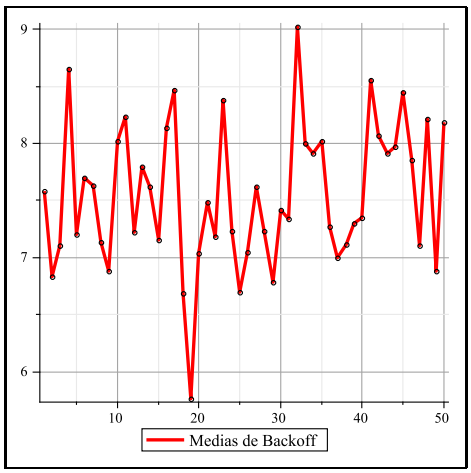

Figura 4.6: *Médias de 50 Amostras de 60 Elementos*

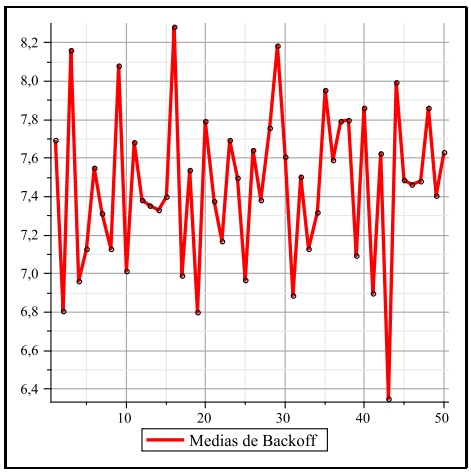

Figura 4.7: *Médias de 50 Amostras de 125 Elementos*

.

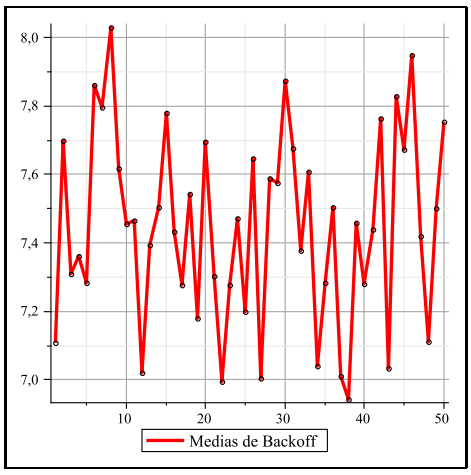

Figura 4.8: *Médias de 50 Amostras de 250 Elementos*

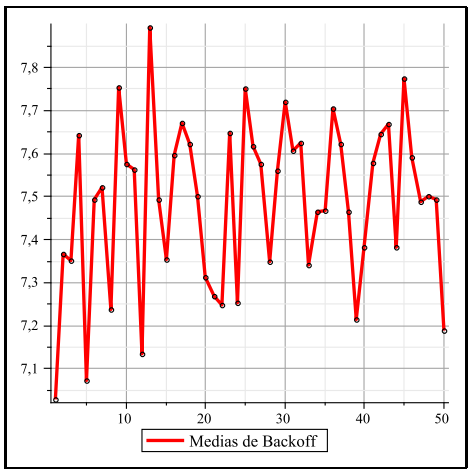

Figura 4.9: *Médias de 50 Amostras de 500 Elementos*

Tabela 4.4: Estatísticas da Distribuição das Médias de 50 Amostras

X_n	x_n	$\overline{X}$	σ^2	σ
50	60	7,548	0,3900	0,6245
50	125	7,456	0,1620	0,4025
50	250	7,448	0,07773	0,2788
50	500	7,488	0,03694	0,1922

Analisando os gráficos obtidos, pode-se notar que o teorema é válido, pois à medida que a amostra aumenta a variância tende a diminuir, pois temos que a variância da distribuição amostral da média é $Var(\overline{X}) = \frac{\sigma^2}{n}$. Essa é a teoria fundamental do Teorema do Limite Central.

Teorema 1. *Para amostras aleatórias simples* $(X_1, ..., X_n)$ *retiradas de população com média* μ *e com variância* σ^2 *finita, a distribuição amostral da média* $\overline{X}$ *aproxima-se para* n *grande, de distribuição normal, com média* μ *e variância* σ^2/n *[37].*

De acordo com o teorema, a distribuição amostral se aproxima da normal, nesse caso podendo expressar de outra maneira o *Teorema do Limite Central*, onde podemos chegar à definição da estatística Z, que representa a probabilidade de dado intervalo sob a curva da normal.

Corolário 1. *Seja* $(X_1, ..., X_n)$ *uma amostra aleatória simples da população* X*, com média* μ *e variância* σ^2 *finita, e* $\overline{X} = \frac{(X_1+...+X_n)}{n}$*, então*

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \sim N(0,1) \Rightarrow Z = \frac{\sqrt{n}(\overline{X} - \mu)}{\sigma} \tag{4.3}$$

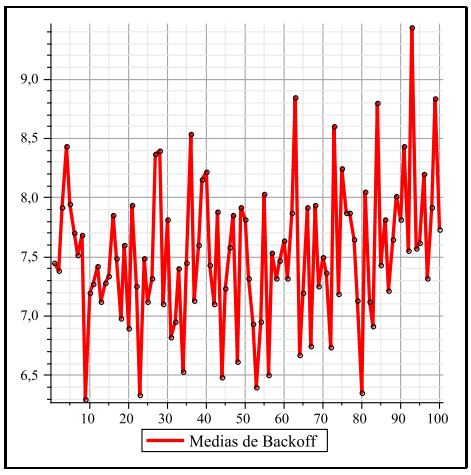

Figura 4.10: *Médias de 100 Amostras de 60 Elementos*

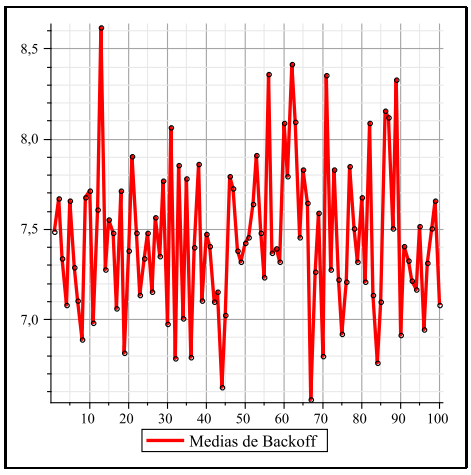

Figura 4.11: *Médias de 100 Amostras de 125 Elementos*

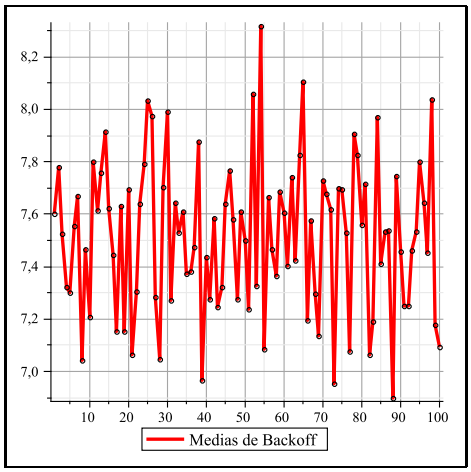

Figura 4.12: *Médias de 100 Amostras de 250 Elementos*

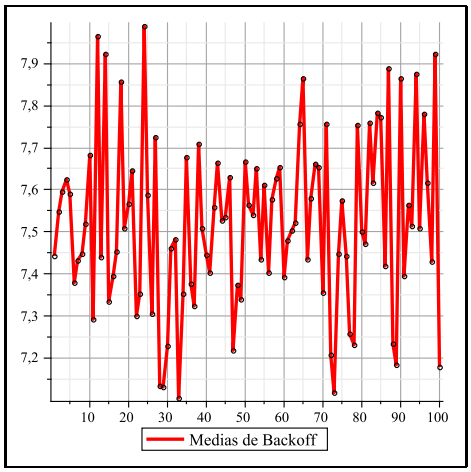

Figura 4.13: *Médias de 100 Amostras de 500 Elementos*

Quando se trata da análise do comportamento de determinado nó, a quantidade de transmissões devem ser analisadas dentro de determinado intervalo de tempo. Sendo assim, suponha que estejamos estimando a média populacional $\mu \cong \frac{backoff}{2}$. No entanto, para obtermos isso, usaremos a média amostral $\overline{X} = \frac{[X_1, X_2, ..., Xn]}{n}$ baseada na quantidade n de elementos da amostra, onde podemos chegar à seguinte expressão:

$$P(|\overline{X} - \mu| \leqslant \varepsilon) \geqslant \gamma$$

Da expressão anterior, o intervalo, $0 < \gamma < 1$ é a probabilidade de que a variação das diferenças das médias que deve ser menor que o erro amostral

Tabela 4.5: Estatísticas da Distribuição das Médias de 100 Amostras

X_n	x_n	$\overline{X}$	σ^2	σ
100	60	7.524	0.3570	0.5975
100	125	7.450	0.1722	0.4150
100	250	7.515	0.08335	0.2887
100	500	7.521	0.04105	0.2026

máximo ε , esteja dentro de determinado intervalo. Pelo teorema do limite central, quando aumentamos o tamanho da amostra, a distribuição das médias se aproxima de uma distribuição normal. Então, podemos afirmar que $\overline{X} \sim N(\mu, \sigma^2/n)$, o que implica que $\overline{X} - \mu \sim N(0, \sigma^2/n)$. Assim, podemos transformar a equação $P(|\overline{X} - \mu| \leqslant \varepsilon) \geqslant \gamma$. Da expressão 4.3, sabemos que $Z = \frac{\sqrt{n}(\overline{X}-\mu)}{\sigma}$, então

$$P(-\varepsilon \leqslant \overline{X} - \mu \leqslant \varepsilon) = P(\frac{-\sqrt{n}\varepsilon}{\sigma} \leqslant Z \leqslant \frac{-\sqrt{n}\varepsilon}{\sigma}) \approx \gamma$$

Da idéia do teorema do limite central, podemos de um dado γ chegarmos nos intervalos de confiança z_γ tal que $P(-z_\gamma < Z < z_\gamma) = \gamma$.

Logo, das expressões acima, podemos chegar à seguinte equação:

$$\frac{\sqrt{n}\varepsilon}{\sigma} = z_\gamma, \quad \textit{onde podemos obter} \quad n = \frac{\sigma^2 z_\gamma^2}{\varepsilon^2}$$

De z_γ, ε e a variância σ^2 podemos chegar ao tamanho da amostra que desejamos. Antes, contudo, temos de saber como obter o valor de z_γ . Esse valor vai depender da probabilidade desejada. Para encontrar z_γ, temos de estabelecer alguns parâmetros além de consultar a tabela Z anexa. Vamos supor que desejamos estabelecer $\gamma = 0,95$, ou seja, queremos probabilidade de 5%. Como a tabela usada é bilateral, então para cada lado da curva da normal, temos 2,5% de probabilidade. Observe que a figura 5.1 do anexo representa a área usada no cálculo da tabela Z, ou seja, para obtermos o valor referente a 2,5% temos que subtrair $0,025$ de $0,5$, onde encontramos 0,47500. O valor z_γ é obtido observando a linha e coluna que se encontra 0,47500. A linha é identificada por números que vão de 0,0 até 3,9 e as colunas vão de 0,00 a 0,09. Logo, o valor de z_γ = 1,96 que são as coordenadas referentes a 0,47500 na tabela Z.

Exemplo 3. *Seja a variância igual a18,75 (variância de uma população de distribuição uniforme), fixando um erro amostral de* $\varepsilon = 0,5$ *e de acordo com*

um dado $\gamma = 0,95$*, o que implica* z_γ *=1.96, podemos encontrar* n*:*

$$n = \frac{18,75 \cdot (1,96)^2}{(0,5)^2} \simeq 288$$

Assim, a probabilidade de que uma das médias das amostras esteja fora do intervalo definido é de apenas 2,5%, isto é, em suposta análise de intervalos de *backoff*, se tivermos medidas fora da média calculada e queremos precisão de 2,5%, podemos considerar o nó malicioso. Para melhor exemplificar, a tabela 4.6 faz comparação de quantas amostras temos de ter para obtermos margem de erro desejada.

Tabela 4.6: Quantidade de Amostras com $\varepsilon = 0,3$ e $\sigma^2 = 18,75$

Margem	Tabela - Z	z_γ	n
3,00%	0,46995	1,88	736
2,50%	0,47500	1,96	800
2,01%	0,47982	2,05	876
1,50%	0,48500	2,17	981
1,01%	0,48983	2,32	1121

4.3.1.2 Análise de Posicionamento

Redes *ad hoc*, como toda rede sem fio, não possuem topologia rígida como as redes cabeadas, principalmente se a rede em questão tem grande mobilidade. Partimos da idéia que um nó monitorado pode gerar intervalos de *backoff* alterados, sob a alegação que não teve o seu meio ocupado. Por esse motivo, propomos cenário simples com o objetivo de explorar situações em que as áreas comuns entre vizinhos seja mínima. Para esse estudo, consideramos que para todos os nós, os rádios possuem a mesma características e com áreas de abrangências iguais onde o raio considerado é 1.

O cenário proposto foi em linha por apresentar a menor área de interseção.

Suponha que logo após uma transmissão de D, C comece a monitorar seus intervalos de *backoffs*. De acordo com as transmissões envolvidas na figura 4.14, podemos construir a tabela 4.7 considerando A $\rightarrow$ Tx e B $\rightarrow$ RX.

Como C está sob influência dos sinais tanto de B quanto de D, ao perceber o CTS transmitido, ele estabelece o NAV de acordo com o tempo definido no cabeçalho da mensagem. Nesse mesmo momento, D não recebe o sinal advindo das transmissões entre A e B, logo ele continuaria a decair o seu

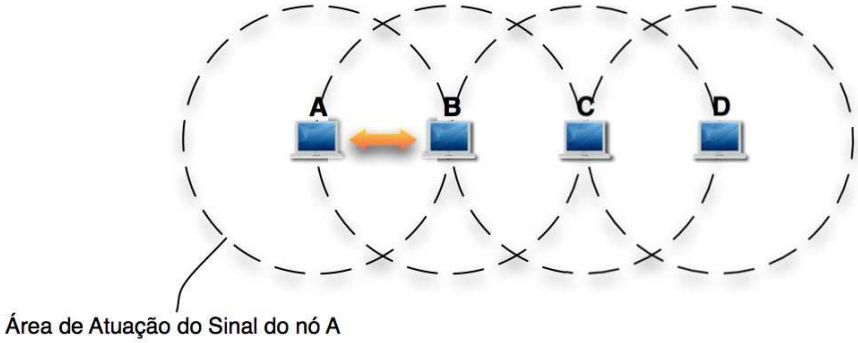

Figura 4.14: Cenário 1 - Transmissão entre os Nós A e B

Tabela 4.7: Sinais Percebidos por Cada Nó

Nós	RTS (A)	CTS (B)	DADOS (A)	ACK (B)
A	Transmite	Recebe	Transmite	Recebe
B	Recebe	Transmite	Recebe	Transmite
C	Decai *Backoff*	NAV (CTS)	NAV	NAV
D	Decai *Backoff*	Decai *Backoff*	Decai *Backoff*	Decai *Backoff*

backoff impossibilitando C de inferir o *backoff* executado por D, levando obviamente a avaliações incorretas. Em outra mão, se D estivesse sob a influência de outra fonte de sinal de forma que C não escutasse, C poderia inferir um *backoff* maior do que realmente foi executado.

Agora, vamos avaliar o cenário proposto pela figura 4.16, supondo que A comece uma transmissão com B. Analisando os sinais envolvidos, temos a seguinte tabela:

De acordo com a tabela 4.9, o nó D ouvirá o CTS de B e deverá estabelecer seu NAV paralisando o seu decaimento, juntamente com C. Com isso, C terá como fazer um registro mais coerente dos históricos de *backoffs* praticados por D.

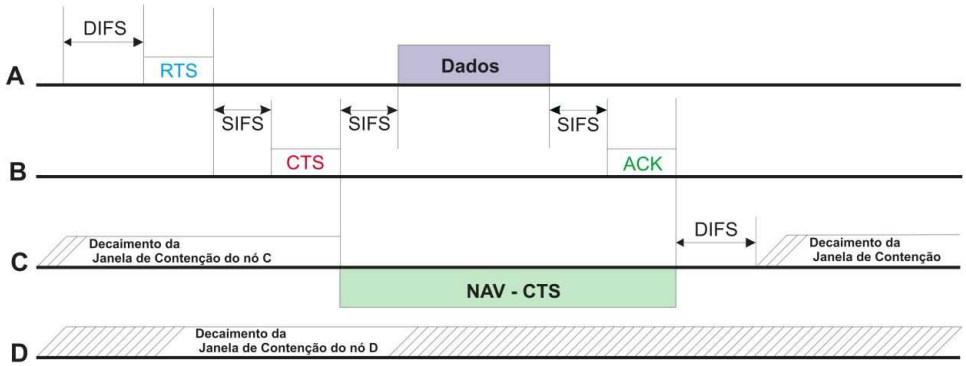

Figura 4.15: Decaimento do *Backoff* - B e C com Elo de Vizinhança

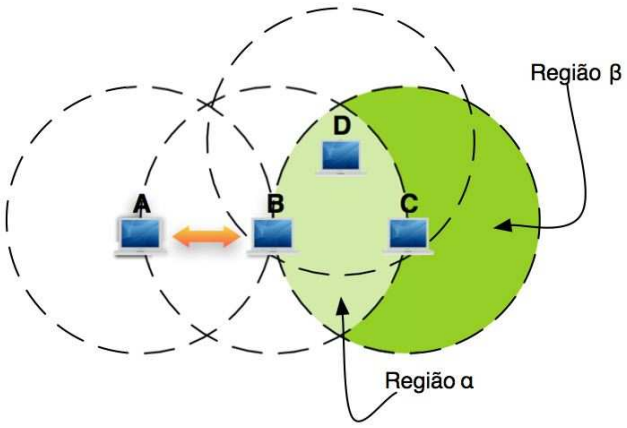

Figura 4.16: Cenário 2 - Transmissão entre os Nós A e B

Tabela 4.8: Sinais Percebidos por Cada Nó

Nós	RTS (A)	CTS (B)	DADOS (A)	ACK (B)
A	Transmite	Recebe	Transmite	Recebe
B	Recebe	Transmite	Recebe	Transmite
C	Decai *Backoff*	NAV (CTS)	NAV	NAV
D	Decai *Backoff*	NAV (CTS)	NAV	NAV

Observando os dois cenários, temos duas possíveis áreas onde um nó pode ser vizinho de C e B (região α) ou ser somente vizinho de C (região β). Saber onde um nó monitorado possa estar, é de fundamental importância para obtermos um leitura correta dos backoffs. Propomos solução simples baseada na troca de tabelas de rotas entre nós distantes dois saltos. Com isso, de acordo com os cenários propostos, C verifica nas tabelas de seus vizinhos (B) se os mesmos têm elo de vizinhança com o nó monitorado (D), se houver C saberá que ele se enquadra no cenário 2 proposto, caso contrário, estará no cenário 1, ou seja, o nó monitorado sofrerá as influências verificadas na tabela 4.7. Se C estiver no limite de alcance da vizinhança de B, podemos encontrar a área onde um nó é vizinho de B e C utilizando a seguinte equação:

$$2 \cdot \int_{0}^{\frac{\sqrt{3}}{2}} 2 \cdot \sqrt{1 - x^2} - 1 \ \ dx \tag{4.4}$$

De acordo com a equação 4.4, a região onde um nó é vizinho de B e C é de aproximadamente $\frac{4\pi - 3\sqrt{3}}{6}$, ou seja, no pior caso a probabilidade que um vizinho de C seja vizinho de B simultaneamente (região α) é de aproximadamente 39,1% e ser vizinho de apenas de C (região β) é de 60,8%.

É intuitivo que essas áreas variem de tamanho dependendo da distân-

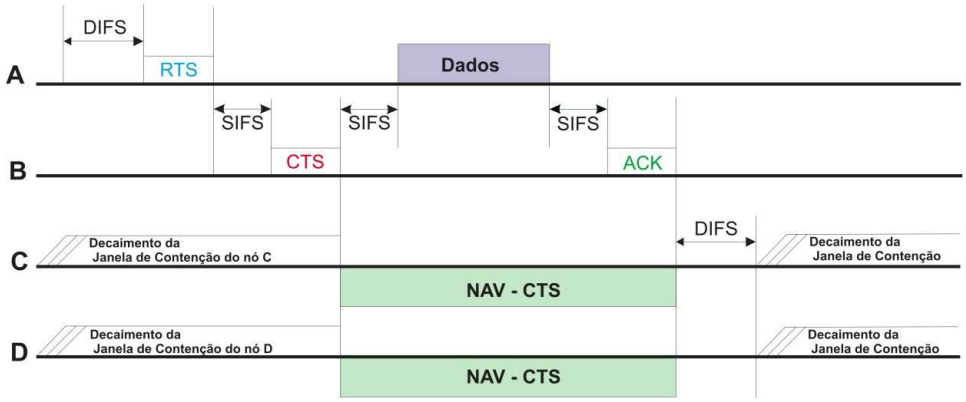

Figura 4.17: Decaimento do *Backoff* - C, D e B com Elo de Vizinhança

cia que C esteja de B. Como todos os nós possuem as mesmas características, podemos utilizar o nível do sinal para definir a distância utilizando as técnicas descritas no artigo [39], que propõem solução de baixo custo e assíncrona.

Da equação 4.4, podemos deduzir outra equação que nos fornece a área β em função da distância entre B e C. Pelo gráfico da figura 4.18, podemos ver que C se torna vizinho de B quando a distância é menor que 1 e varia até $r \approx 0$.

Seja $x \in \mathbb{R} \mid x \in [0, 1]$

$$f(x) = \frac{-x\sqrt{4-x^2}}{2} + 2arcsen(\frac{\sqrt{4-x^2}}{2}) \tag{4.5}$$

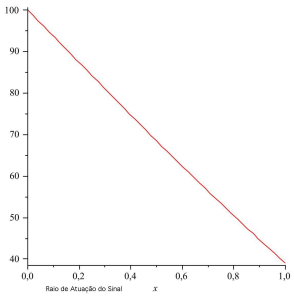

Figura 4.18: Porcentagens da Área β em Função da Distância entre B e C

Existem soluções que não consideram a influência dos sinais estudados por serem redes baseadas em *single-hops* [28]. Nosso modelo baseia-se em rede *ad hoc* de saltos múltiplos, logo tivemos de levar em consideração esses sinais. A troca de tabela de roteamento é uma solução simples é dá condições de inferir de forma mais coerente os *backoffs* executados.

Dependendo da sensibilidade do rádio, um nó pode ouvir ruídos advindos de outras transmissões, considerando o meio como ocupado. Diferentemente dos sinais recebidos claramente, não é possível identificar a origem

Tabela 4.9: Probabilidade de um Vizinho de C ser Vizinho de B

raio	Porcentagem da área β
0,000	100 %
0,125	92,04 %
0,250	84,12 %
0,375	76,26 %
0,500	68,50 %
0,625	60,86%
0,750	53,39%
0,875	46,12%
1,000	39,10%
1,001	0,00%

dos ruídos, mas é possível inferir sua possível fonte. Conforme vimos no cenário proposto na figura 4.14, se A transmite para B, C ouve CTS/ACK e os ruídos advindos das transmissões de A. Se C mensurar o intervalo de tempo entre os ruídos e as mensagens de CTS e ACK, ele pode inferir a possível localização da origem do ruído que neste caso tem como sua fonte o nó A. Esta é apenas uma abordagem das inúmeras que podem ser feitas a respeito da influência dos ruídos gerados por transmissões. Um estudo mais complexo seria necessário para explorar mais os conceitos que envolvem os ruídos o que não será abordado por este trabalho.

4.4 Proposta: Sistema de Reputação

Estudamos nas seções anteriores que podemos inferir, com um margem de erro, a confiabilidade de um determinado nó levando em consideração a sua posição em relação aos seus vizinhos. De acordo com simulações feitas na subseção 4.3.1.1, concluímos, também, que dependendo da margem de erro desejada podemos estabelecer a quantidade mínima para que seja possível fazer inferências sob a maliciosidade do nó em questão. Associando essas soluções, propomos um modelo de detecção de maliciosidade voltado para uma solução em multicamadas ou *cross-layer*. A idéia do modelo é monitorar e fornecer informações às diversas camadas do modelo TCP, de maneira a estabelecer níveis de confiança baseados não apenas em um único parâmetro mais em um conjunto deles. Nosso modelo pode ser visto na figura 4.19, onde vemos em detalhes cada um dos módulos do sistema.

4.4.1 Módulo Monitor

Este módulo é o responsável em monitorar informações das camada de enlace e rede e repassar as informações ao sistema de reputação que será o

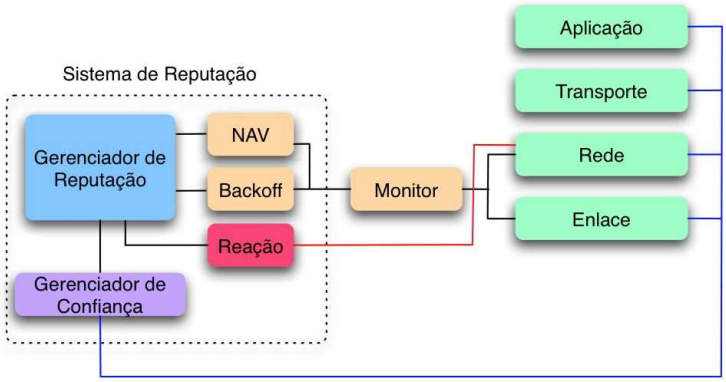

Figura 4.19: Sistema de Reputação Baseado em *Cross-Layer*

responsável por estabelecer os níveis de confiança em relação ao nó monitorado. Os dados são coletados de forma passiva, de maneira a não influenciar no funcionamento estabelecido pelo padrão IEEE 802.11.

4.4.2 Módulo de Reputação

O módulo de reputação é o núcleo do modelo proposto; nele, temos seis módulos que são responsáveis por receber os dados do módulo monitor e fornecer níveis de confiança às diversas camadas.

4.4.2.1 Módulo de Reputação do NAV

A cada mensagem de RTS e CTS, o valor do campo *Duration* é registrado e comparado com o tempo que o meio esteve ocupado. Se depois do envio da mensagem de ACK o meio ficar ocioso por mais de dois DIFS *time*, um contador temporário $C_{Temp(nav)}$ é decrementado. Quando o contador temporário chegar no limite -3, o contador que estabelece o nível de reputação $C_{nav(t)_p}$ onde t é o instante do evento, é decrementado de 1, e o contador temporário é incrementado de 1. O contador $C_{nav(t)_p}$ tem seu valor inicial igual a 0 e pode varia de -3 a 3, e toda vez que seu valor muda o mesmo é enviado para o gerenciador de reputação por meio do cálculo do índice de reputação do NAV $In_{(t)_p}$, onde p é a identificação do nó monitorado.

$$In_{(t)_p} = \begin{cases} 1 & \text{se} \quad 0 < C_{nav(t)} \leq 3 \\ 0 & se \quad C_{nav(t)} = 0 \\ -1 & se \quad -3 \leq C_{nav(t)} < 0 \end{cases} \tag{4.6}$$

Para que o nó possa se redimir, sempre que o NAV for obedecido, o contador temporário $C_{Temp(nav)}$ é incrementado de 1 até chegar a 0 onde o contador $C_{nav(t)}$ também tem o seu valor incrementado de 1. Note que o incremento do nível de reputação é mais custoso, isso busca evitar que o nó

monitorado fique oscilando sua maliciosidade para obter vantagem.

Devido às colisões que podem ocorrer durante a troca de pacotes, esse módulo só altera o contador temporário se houver o recebimento da mensagem de ACK.

4.4.2.2 Módulo de Reputação do *Backoff*

Avaliando os intervalos de transmissões de um nó, podemos inferir os eventuais intervalos de *backoffs* estabelecidos pelo nó monitorado.

Pelos estudos das seções anteriores, sabemos que, dependendo da localização do nó, outras transmissões podem influenciar no decaimento do intervalo de *backoff*. Por isso, o módulo, ao receber as informações de monitoramento, verifica nas tabelas de roteamento, também enviadas pelo módulo monitor, se o nó monitorado possui elo de vizinhança com outro vizinho do nó observador. Se tiver, o nó observador saberá que os mesmos sinais que ele ouve o nó monitorado também ouve; logo, se o observador parar de decair o *backoff* do nó monitorado também precisa parar de decair. A expressão 4.7 representa o que ocorre nesse caso.

$$\text{Backoff}_t = \text{Backoff}_{(monitorado)} \tag{4.7}$$

A equação 4.8 é responsável por estabelecer a média justa do *backoff* $B_{mj(t)}$, que será comparado com a média dos *backoff* monitorados. Um nó malicioso tem o objetivo de ter acesso ao meio a maior parte do tempo, logo visa a ter os menores intervalos de *backoffs* possíveis, nesse caso o cálculo de $B_{mj(t)}$ usa a média considerando a primeira janela de contenção. Para que não haja falsos positivos pelas oscilações a valores próximos do considerado justo, o cálculo do desvio-padrão D_p é considerado em conjunto com uma constante α, usada para estabelecer um tolerância a essas eventuais oscilações.

$$B_{mj(t)} = \frac{CW_{min}}{2} - D_p \cdot \alpha \tag{4.8}$$

A equação 4.9 faz a média aritmética dos *backoffs* monitorados, onde n é a quantidade de observações necessárias, p é a identificação do nó monitorado.

$$B_{m(t)_p} = \sum_{i=1}^{n} \frac{\text{Backoff}_i}{n} \tag{4.9}$$

Pelo estudo estatístico visto nas seções anteriores, temos quantidade n de observações que devemos fazer para realizar as inferências de reputação. Após as n observações, o módulo calcula a média aritmética e verifica o resultado obtido, se o valor obtido for menor que a média justa do *backoff*, o índice de reputação $Ib_{(t)_p}$ onde t é o tempo em que o índice foi calculado e

p é a identificação do nó tem seu valor alterado.

O índice de reputação do *backoff* tem seu valor inicial igual a zero e pode variar entre -1 e 1, onde $0 < Ib_{(t)_p} \leq 1$ significa um nó com comportamento normal e $-1 \leq Ib_{(t)_p} < 0$ um nó com comportamento suspeito ou malicioso.

$$Ib_{(t)_p} = \begin{cases} 1 & \textit{se} \quad B_{mj(t)} \leq B_{m(t)_p} \\ -1 & \textit{se} \quad B_{mj(t)} \geq B_{m(t)_p} \\ 0 & \textit{Caso Contrário} \end{cases} \tag{4.10}$$

Após o cálculo do índice de reputação, o mesmo é enviado ao gerenciador de reputação que veremos a seguir.

4.4.2.3 Gerenciador de Reputação

A função deste módulo é receber os índices de reputação calculados pelos módulos de reputação mais o índice enviado pelos vizinhos e estabelecer um índice geral de reputação IR_p que será disponibilizado para as demais camadas de aplicação, transporte, rede e enlace.

Assim que os dados são recebidos, estes são armazenados em uma tabela indexando o tempo em que foram obtidos. Isso é necessário para que seja possível estabelecer pesos na obtenção do índice geral. A equação 4.11 utiliza esses tempos para obter média ponderada dos índices utilizados nos cálculos. A média ponderada é utilizada para que seja possível um nó considerado malicioso se redimir com o passar do tempo.

$$IR_p = \frac{(Ib_{(t_{ant})_p} \cdot t_{ant}) + (Ib_{(t)_p} \cdot t)}{t_{ant} + t} + \frac{(In_{(t_{ant})_p} \cdot t_{ant}) + (In_{(t)_p} \cdot t)}{t_{ant} + t} + (Ivz_{p' \to p}) \cdot \beta \tag{4.11}$$

$$onde \quad \beta = \begin{cases} 0 & \text{se} \quad IR_{p'} < 0 \\ 0,1 & \text{se} \quad IR_{p'} > 0 \end{cases}$$

O nó observador pode utilizar informações dos seus vizinhos (p') para a construção do índice geral. No entanto, isso vai depender da confiança que o nó observador tem em relação ao nó que forneceu tal informação, além de ter apenas 10% do seu valor considerado. Na equação 4.11, o índice fornecido pelo vizinho p' é identificado como $Ivz_{p' \to p}$.

O índice geral de reputação tem seu valor inicial igual a 0 e varia de -1 a 1, considerando como nós malicioso os que recebem índices menores que zero. Depois de obtido o índice geral de reputação, o gerenciador registra-o em uma tabela juntamente com o tempo, onde serão utilizados pelo gerenciador de confiança.

4.4.2.4 Gerenciador de Confiança

O gerenciador de confiança é o responsável por tratar e fornecer às diversas camadas informações do nível de confiança calculado pelo gerenciador de reputação. As camadas podem utilizar informações de confiança para tomadas de decisão. Podemos tomar como exemplo um aplicação cliente que deseja trocar informações importantes com um aplicação servidor em outro nó.

Por meio do gerenciador de confiança, a aplicação pode determinar quais dados podem ser enviados dependendo do nível de confiança com o nó servidor, sem que seja preciso repassar pacotes para as camadas abaixo.

A tabela 4.10 traz comparação dos parâmetros considerados na construção dos índices de confiança.

Tabela 4.10: Comparação dos Parâmetros que Estabelecem Níveis de Confiança

Backoff	*NAV*	Reputação	Confiança
0	0	0	Confiável
0	1	1	Confiável
0	-1	-1	Não Confiável
1	0	1	Confiável
1	1	1	Confiável
1	-1	-1	Não Confiável
-1	0	-1	Não Confiável
-1	1	-1	Não Confiável
-1	-1	-1	Não Confiável

4.4.2.5 Módulo de Reação

Quando um nó é considerado malicioso, uma série de medidas precisam ser tomadas para minimizar os efeitos provocados por ele. Mas, quando estamos em uma rede sem fio, onde os nós são entes independentes e de processamento local, pouco podemos fazer, uma vez que não podemos desligar o nó monitorado. A medida mais adequada neste caso é isolar o nó considerado malicioso não repassando pacotes ou até mesmo removendo-o das tabelas de roteamento.

O módulo de reação fica em constante sincronia com a tabela de reputação e, dependendo do índice verificado, recebe alerta que é repassado para a camada de rede, que pode tomar diversas providências que vão da simples utilização de rotas alternativas até a remoção completa da rota para o nó malicioso.

No capítulo 2, vimos que o padrão IEEE 802.11 estabelece uma série de parâmetros que podem ser burlados para obtenção de vantagens. Das

análises realizadas no capítulo 3, concluímos que podemos fazer um sistema mais eficiente observando eventos da camada de enlace, uma vez que é nessa camada que ocorre um dos principais problemas no que diz respeito ao processo de *backoff* e NAV. O sistema de reputação proposto busca resolver tentativas de burla do intervalo de *backoff*, com quantidade mínima de observações, tornando possível avaliações mais rápidas e análises mais precisas.

Poucos simuladores implementam de forma fiel o padrão IEEE 802.11. Foram utilizados o Glomosim e o NS-2 na tentativa de obter valores coerentes, mas ao tentar fazer inferências ao processo de *backoff*, verificou-se que os simuladores não escolhem um intervalo de *backoff* se antes de transmitir o canal esteja ocioso [6], ou seja, eles transmitem sem estabelecer um intervalo de *backoff*. Esse é exatamente o ponto que precisamos avaliar para validar o sistema proposto. Adaptar os simuladores a nossa necessidade pede grau maior de complexidade, logo um trabalho futuro, faremos a validação do sistema proposto descrito.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho, procuramos mostrar os padrões e protocolos envolvidas no processo de organização de uma rede *ad hoc*, bem como as fraquezas exploradas, a fim de propor sistema de detecção de maliciosidade que estabeleça confiança baseada em dados da camada de enlace.

Durante o processo de busca do estado da arte sobre as eventuais situações que podem ser exploradas no padrão estabelecido pelo IEEE 802.11, percebemos que muitas pesquisas preocupavam-se em estabelecer mecanismos de confiança visando a características de apenas uma camada, sem levar em conta informações advindas de outras. Como vimos na seção 2.2.3, a idéia de termos implementações em multicamadas é algo ainda na teoria que não está implementado comercialmente. Desse pensamento, podemos ter a liberdade de inferir que o motivo das poucas pesquisas propostas na literatura não se limita apenas a fatores que envolvem o grau de complexidade. No entanto, há vários trabalhos publicados que se preocupam com a questão e que propõem boas soluções.

Avaliando todas as soluções, observamos que boa parte dessas eram eficientes em um ponto e falhas em outro. Na verdade, não existe solução ideal, o que podemos fazer é propor uma solução que possa abranger a maior quantidade de problemas possíveis e propor soluções baratas. A idéia de utilizar um sistema multicamadas vem ao encontro com essa vertente. As redes *ad hoc* são sistemas que geram grande dificuldades na proposição de sistema de reputação eficiente. Sabemos que em redes desse tipo, os nós envolvidos não são capazes de saber com exatidão o que ocorre nos seus vizinhos, o que pode ser agravado se forem consideradas redes em *multi-hops*. Frente à essa dificuldade, a maioria dos sistemas propostos acabam tendo um ponto fraco. O DOMINO [28], por exemplo, tenta inferir o intervalo de *backoff* praticado, ouvindo o intervalo de transmissão somado a cálculos desenvolvidos por meio da cadeia de *markov*. No entanto, o DOMINO não foi desenvolvido para redes com multisaltos, logo se torna ineficiente quando tentamos utilizá-lo em redes maiores onde conseqüentemente teremos multisaltos. O trabalho descrito no artigo [22] propõe alterações no padrão IEEE 802.11 para solucionar eventuais burlas no processo de *backoff*. No entanto, esta idéia não funciona se um nó vizinho que não seja também do nó monitorado começar a transmitir, neste caso teríamos lei-

turas equivocadas do nó monitorado o resultaria condenações falsas. Note que há grande diversidade de soluções que conseqüentemente apresentam ineficiência.

A abordagem proposta neste trabalho nasceu da necessidade de uma solução multicamadas que fosse eficiente em redes *ad hoc*. A continuidade deste trabalho passa pelo desenvolvimento de um processo de validação uma vez que não podemos realizá-lo a contento.

Vimos que há vários métodos que tentam inferir os intervalos de *backoffs*, no entanto, esse processo não é fácil se não levarmos em conta os sinais envolvidos, principalmente os ruídos. Em trabalhos futuros, iremos realizar estudo mais complexo das causas e efeitos que os ruídos gerados por outras transmissões podem causar ao modelo proposto, além de estabelecer meios de identificar sua origem. Estudos e simulações em situações mais severas, como em redes de grande mobilidade, serão o próximo passo, além de estabelecer a validação do modelo proposto.

Referências Bibliográficas

[1] IEEE Standard 802.2. Logical link control standart. *IEEE Standard*, pages 1–245, May 1998.

[2] IEEE Standard 802.2. Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications. *IEEE Standard*, pages 1–95, Oct. 2001.

[3] Josang A., Ismail R., and Boyd C. A survey of trust and reputation systems for online service provision, 2006. Distributed Systems Technology Centre, University of Queensland, Australia, 2006.

[4] ANATEL. Agência Nacional de Telecomunicações, 2006.

[5] Tanenbaum Andrew S. *Computer Networks*. Prentice Hall, third edition, 1996.

[6] Ryad Ben-El-Kezadri and Farouk Kamoun. Yavista: A graphical tool for comparing 802.11 simulators. *JOURNAL OF COMPUTERS*, 3(2):10–20, Feb. 2008.

[7] Perkins C. Ad hoc on demand distance vector (aodv) routing, Nov. 2000.

[8] Perkins C.E., E.M. Royer, S.R. Das, and M.K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 8(1):16–28, Feb. 2001.

[9] Grinstead C.M. and Snell J. L. *Introduction to Probability*, volume 1. American Mathematical Society, 1997.

[10] Lought D. L., Blankenship T. K., and Krizman K. J. Tutorial on wireless LAN's and ieee 802.11. Jun. 1995. http://computer.org/students/looking/summer97/ieee802.htm.

[11] Gambetta Diego. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*. Basil Blackwell, 1988. Reprinted in electronic edition from Department of Sociology, University of Oxford, chapter 13, pp. 213-237".

[12] Lei Guang and C. Assi. Vulnerabilities of ad hoc network routing protocols to mac misbehavior. *Wireless and Mobile Computing, Networking and Communications, 2005. (WiMob'2005), IEEE International Conference on*, 3:146–153 Vol. 3, Aug. 2005.

[13] Lei Guang and C. Assi. A self-adaptive detection system for mac misbehavior in ad hoc networks. *Communications, 2006. ICC '06. IEEE International Conference on*, 8:3682–3687, Jun. 2006.

[14] Lei Guang and Chadi Assi. Cross-layer cooperation to handle mac misbehavior in ad hoc networks. *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, pages 219–222, May 2006.

[15] IEEE. Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless LAN medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, Dec. 2007.

[16] Liu J. and Singh S. Atcp: Tcp for mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 19(7):1300–1315, Jul. 2001.

[17] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[18] Pradeep Kyasanur. Selfish mac layer misbehavior in wireless networks. *IEEE Transactions on Mobile Computing*, 4(5):502–516, 2005. Senior Member-Nitin H. Vaidya.

[19] Buttyán L. and Hubaux J. Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks. Technical Report DSC/2001, 2001.

[20] Goldberg L. Wireless LAN's: Mobile computing's second wave. *Eletronic Design*, Jun. 1995.

[21] D.H. McKnight and N.L Chervany. The meanings of trust, 1996. Working paper Series 96-04, University of Minnesota, Management Information Systems Reseach Center, URL: http://misrc.umn.edu/wpaper/.

[22] Kyasanur P. and Vaidya N.H. Selfish mac layer misbehavior in wireless networks. *Mobile Computing, IEEE Transactions on*, 4(5):502–516, Sept.-Oct. 2005.

[23] Michiardi P. and Molva R. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks, 2001.

[24] Papadimitratos P. and Haas Z. Secure routing for mobile ad hoc networks, 2002.

[25] Svetlana Radosavac, John S. Baras, and Iordanis Koutsopoulos. A framework for mac protocol misbehavior detection in wireless networks. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, pages 33–42, New York, NY, USA, 2005. ACM.

[26] Vijay T. Raisinghani and Sridhar Iyer. Cross-layer design optimizations in wireless protocol stacks, Mar. 2003.

[27] Lars Rasmusson and Sverker Jansson. Simulated social control for secure Internet commerce. In Catherine Meadows, editor, *Proceedings of the 1996 New Security Paradigms Workshop*. ACM, 1996.

[28] M. Raya, I. Aad, J.-P. Hubaux, and A. El Fawal. Domino: Detecting mac layer greedy behavior in ieee 802.11 hotspots. *Mobile Computing, IEEE Transactions on*, 5(12):1691–1705, Dec. 2006.

[29] Tang Rui, Zhang Jinbao, Chen Xia, Tan Zhenhui, and Jin Xiaojun. A novel generic cross-layer architecture for next generation mobile communication system. *Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2007 International Symposium on*, pages 43–46, Aug. 2007.

[30] Basagni S., Conti M., Giordano S., and Stojmenovic I. *Mobile Ad Hoc Networking*, volume 1. IEEE Press, Wiley Interscience, 2004.

[31] Buchegger S. and Le Boudec J. Performance analysis of the CONFIDANT protocol: Cooperation of nodes — fairness in dynamic ad-hoc networks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, CH*, Jun. 2002.

[32] Buchegger S. and Le Boudec J. A robust reputation system for mobile ad hoc networks, 2003. S. Buchegger and J.-Y. Le Boudec. A Robust Reputation System for Mobile ad hoc Networks. Technical Report IC/2003.

[33] Kimaya Sanzgiri, Brian Neil Levine, Clay Shields, Bridget Dahill, and Elizabeth M. Belding-Royer. Aran: A secure routing protocol for ad hoc networks. *10th IEEE International Conference*, pages 78 – 87, Nov. 2002.

[34] The Official Bluetooth Web Site. site: http://www.bluetooth.com.

[35] M. Srivastava, V.; Motani. Cross-layer design: a survey and the road ahead. *Communications Magazine, IEEE*, 43(12):112–119, Dec. 2005.

[36] Kawadia V. and Kumar P.R. A cautionary perspective on cross-layer design. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 12(1):3–11, Feb. 2005.

[37] Bussab W. O. and Morettin P. A. *Estatística Básica*, volume 1. Saraiva, 2004.

[38] Qi Wang and Abu-Rgheff M.A. Cross-layer signalling for next-generation wireless systems. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 2:1084–1089 vol.2, Mar. 2003.

[39] C.-Y. Wen, R. D. Morris, and W. A. Sethares. Distance estimation using bidirectional communications without synchronous clocking. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 55(5):1927–1939, May 2007.

[40] Hu Y., Perrig A., and Johnson D. Ariadne: A secure on-demand routing protocol for ad hoc networks, 2002.

[41] Hu Y., Johnson D., and Perrig A. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), IEEE, Calicoon, NY*, Jun. 2002.

Anexo

Tabela Z

A tabela Z é utilizada para obtenção das probabilidades sob a curva da normal, conforme a figura 5.1

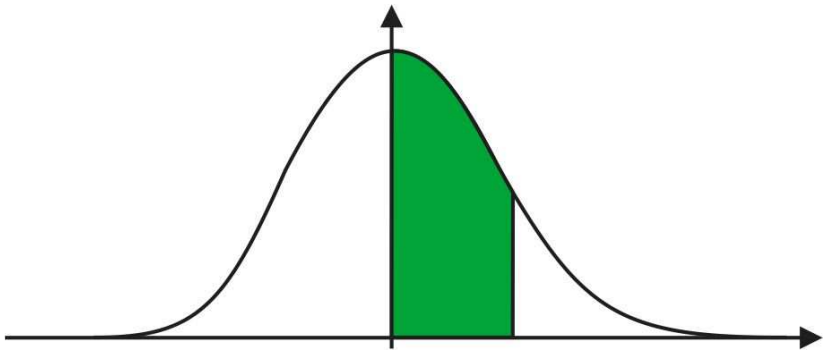

Figura 5.1: Área de Cálculo da Tabela Z

Tabela 5.1: Tabela Z - Primeira Parte

z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,00000	0,00399	0,00798	0,01197	0,01595	0,01994	0,02392	0,02790	0,03188	0,03586
0,1	0,03983	0,04380	0,04776	0,05172	0,05567	0,05962	0,06356	0,06749	0,07142	0,07535
0,2	0,07926	0,08317	0,08706	0,09095	0,09483	0,09871	0,10257	0,10642	0,11026	0,11409
0,3	0,11791	0,12172	0,12552	0,12930	0,13307	0,13683	0,14058	0,14431	0,14803	0,15173
0,4	0,15542	0,15910	0,16276	0,16640	0,17003	0,17364	0,17724	0,18082	0,18439	0,18793
0,5	0,19146	0,19497	0,19847	0,20194	0,20540	0,20884	0,21226	0,21566	0,21904	0,22240
0,6	0,22575	0,22907	0,23237	0,23565	0,23891	0,24215	0,24537	0,24857	0,25175	0,25490
0,7	0,25804	0,26115	0,26424	0,26730	0,27035	0,27337	0,27637	0,27935	0,28230	0,28524
0,8	0,28814	0,29103	0,29389	0,29673	0,29955	0,30234	0,30511	0,30785	0,31057	0,31327
0,9	0,31594	0,31859	0,32121	0,32381	0,32639	0,32894	0,33147	0,33398	0,33646	0,33891
1,0	0,34134	0,34375	0,34614	0,34849	0,35083	0,35314	0,35543	0,35769	0,35993	0,36214
1,1	0,36433	0,36650	0,36864	0,37076	0,37286	0,37493	0,37698	0,37900	0,38100	0,38298
1,2	0,38493	0,38686	0,38877	0,39065	0,39251	0,39435	0,39617	0,39796	0,39973	0,40147
1,3	0,40320	0,40490	0,40658	0,40824	0,40988	0,41149	0,41308	0,41466	0,41621	0,41774
1,4	0,41924	0,42073	0,42220	0,42364	0,42507	0,42647	0,42785	0,42922	0,43056	0,43189
1,5	0,43319	0,43448	0,43574	0,43699	0,43822	0,43943	0,44062	0,44179	0,44295	0,44408
1,6	0,44520	0,44630	0,44738	0,44845	0,44950	0,45053	0,45154	0,45254	0,45352	0,45449
1,7	0,45543	0,45637	0,45728	0,45818	0,45907	0,45994	0,46080	0,46164	0,46246	0,46327
1,8	0,46407	0,46485	0,46562	0,46638	0,46712	0,46784	0,46856	0,46926	0,46995	0,47062
1,9	0,47128	0,47193	0,47257	0,47320	0,47381	0,47441	0,47500	0,47558	0,47615	0,47670

Tabela 5.2: Tabela Z - Segunda Parte

z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
2,0	0,47725	0,47778	0,47831	0,47882	0,47932	0,47982	0,48030	0,48077	0,48124	0,48169
2,1	0,48214	0,48257	0,48300	0,48341	0,48382	0,48422	0,48461	0,48500	0,48537	0,48574
2,2	0,48610	0,48645	0,48679	0,48713	0,48745	0,48778	0,48809	0,48840	0,48870	0,48899
2,3	0,48928	0,48956	0,48983	0,49010	0,49036	0,49061	0,49086	0,49111	0,49134	0,49158
2,4	0,49180	0,49202	0,49224	0,49245	0,49266	0,49286	0,49305	0,49324	0,49343	0,49361
2,5	0,49379	0,49396	0,49413	0,49430	0,49446	0,49461	0,49477	0,49492	0,49506	0,49520
2,6	0,49534	0,49547	0,49560	0,49573	0,49585	0,49598	0,49609	0,49621	0,49632	0,49643
2,7	0,49653	0,49664	0,49674	0,49683	0,49693	0,49702	0,49711	0,49720	0,49728	0,49736
2,8	0,49744	0,49752	0,49760	0,49767	0,49774	0,49781	0,49788	0,49795	0,49801	0,49807
2,9	0,49813	0,49819	0,49825	0,49831	0,49836	0,49841	0,49846	0,49851	0,49856	0,49861
3,0	0,49865	0,49869	0,49874	0,49878	0,49882	0,49886	0,49889	0,49893	0,49896	0,49900
3,1	0,49903	0,49906	0,49910	0,49913	0,49916	0,49918	0,49921	0,49924	0,49926	0,49929
3,2	0,49931	0,49934	0,49936	0,49938	0,49940	0,49942	0,49944	0,49946	0,49948	0,49950
3,3	0,49952	0,49953	0,49955	0,49957	0,49958	0,49960	0,49961	0,49962	0,49964	0,49965
3,4	0,49966	0,49968	0,49969	0,49970	0,49971	0,49972	0,49973	0,49974	0,49975	0,49976
3,5	0,49977	0,49978	0,49978	0,49979	0,49980	0,49981	0,49981	0,49982	0,49983	0,49983
3,6	0,49984	0,49985	0,49985	0,49986	0,49986	0,49987	0,49987	0,49988	0,49988	0,49989
3,7	0,49989	0,49990	0,49990	0,49990	0,49991	0,49991	0,49992	0,49992	0,49992	0,49992
3,8	0,49993	0,49993	0,49993	0,49994	0,49994	0,49994	0,49994	0,49995	0,49995	0,49995
3,9	>0,49995	etc ...								
z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09