Pós-Graduação em Ciência da Computação

“Análise do Problema do Protocolo MAC IEEE 802.11 em Redes Ad Hoc Multihop”

Por

Adalton de Sena Almeida

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, FEVEREIRO / 2003

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ADALTON DE SENA ALMEIDA

"Análise do Problema do Protocolo MAC IEEE 802.11 Em Redes Ad Hoc Multihop"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: DJAMEL F. H. SADOK

RECIFE, FEVEREIRO / 2003

Almeida, Adalton de Sena
Análise do problema do Protocolo MAC IEEE 802.11 em redes Ad Hoc Multihop / Adalton de Sena Almeida. – Recife : O Autor, 2003.
iv, 117 folhas : il., fig., tab., graf.

Dissertação (mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2003.

Inclui bibliografia e anexo.

1. Computadores – Redes locais sem fio. 2. Protocolo de acesso (DFWMAC) – Redes locais sem fio (IEEE 802.11) – Análise de problemas. 3. Redes sem infraestrutura (Wireless). I. Título.

004.732	**CDU (2.ed.)**	**UFPE**
004.65	**CDD (21.ed.)**	**BC2003-074**

Agradecimentos

A Deus

A quem dedico um agradecimento especial pois, em todos os momentos de minha vida, sinto sua presença, orientando-me e iluminando-me sempre pelos caminhos da Vitória e da Paz.

Meus Pais

Ao meu pai ***Artur*** *e minha mãe* ***Francisca*** *que foram os grandes responsáveis pela minha existência e boa educação sempre voltada para o meu crescimento profissional e humano.*

Esposa e Filhos

À minha esposa ***Aurilene****, pelo apoio e compreensão recebidos nos momentos em que precisei em função dessa conquista. Aos meus filhos* ***Ana Beatriz*** *e* ***Adalton filho****. Vocês são a fonte de inspiração diária na minha vida.*

A todos vocês dedico este título e esta vitória.

Aos Irmãos

Aos meus irmãos ***Antonio Francisco, Ana Célia, Airton e Ana Cléia****, que sempre acreditaram e depositaram em mim toda confiança em meu esforço e dedicação.*

Prof. Djamel Sadok

Ao meu orientador Prof ***Djamel****, por ter acreditado em mim e pelo papel fundamental que desempenhou em todos os momentos. Pelo jeito alegre e brincalhão que serviram para descontrair e diminuir a tensão nos momentos difíceis.*

Profa. Judith Kelner

Pela postura firme, objetiva e criteriosa com que costuma agir. A senhora foi muito importante e essencial colaboradora do meu trabalho.

Ney Paranaguá

Amigo e grande colaborador nessa empreitada. Obrigado pela compreensão, apoio e por ter entendido o significado deste ideal na minha vida.

Demais Agradecimentos

- Ao Centro Federal de Educação Tecnológica do Piauí – CEFET-PI, na pessoa da Diretora Geral, **Profa. Rita Martins de Cássia**, por ter viabilizado este mestrado a mim e aos professores de informática daquela instituição de ensino.

- Aos colegas da turma do mestrado: **Constantino, Eduilson e Erivan** pela ajuda que me prestaram colaborando e ajudando no desenvolvimento dessa dissertação.

- Amigos da Infoway: **Peta, Chico, Marcos, Maxwel, Franz e Luciani** pelo esforço que fizeram me substituindo durante o período que estive ausente procurando sempre fazer o melhor de si.

- Amigo e compadre **Cícero Vilson,** pelo apoio e incentivo creditados em mim e pela sua amizade.

- Amigo **Hildomar,** pela confiança, sinceridade e incentivos permanentes.

- **Carlos Cordeiro,** doutorando da Universidade de Cicinnati (EUA), que muito colaborou para o desenvolvimento e algumas conclusões desta dissertação.

- **Prof. José Ferreira de Rezende,** da UFRJ importante colaborador dessa dissertação.

Resumo

O protocolo MAC IEEE 802.11 DFWMAC (*Distributed Foundation* ***Wireless*** *Access Control*) foi padronizado para uso em redes locais sem fio e tem sido utilizado para testar e simular redes locais sem fio ***ad hoc multihop***. Este protocolo tem apresentado problemas quando trabalhamos com redes *ad hoc multihop.* Este problema fica evidente quando submetemos tráfego **TCP** (*Transmission Control Protocol*) entre duas estações. Por tratar-se de um protocolo de controle de acesso ao meio distribuído, não possuindo um controle central, a decisão de transmissão é feita pelas próprias estações de acordo com o funcionamento do **DFWMAC**. Ainda pelas suas características de funcionamento distribuído, problemas de "terminal escondido" e "terminal exposto" podem ocorrer comprometendo de maneira significativa o tráfego de conexões TCP. Associado aos problemas de "terminal escondido" e "terminal exposto", o algoritmo de ***Backoff*** Exponencial Binário (BEB) contribui para que este protocolo não funcione bem em redes *ad hoc multihop*.

O resultado da ação de todos estes problemas é a degradação do ***throughput*** do TCP gerando **instabilidade** e **injustiça** no acesso ao meio compartilhado. A instabilidade fica evidente quando a variação do *throughput* é muito alta em intervalos de tempo muito curtos. Isto pode ser visto com apenas uma conexão TCP entre duas estações. Já o problema de injustiça aparece quando submetemos duas conexões TCP simultâneas, sendo que uma consegue transmitir pacotes de dados a uma taxa alta, utilizando toda a largura de banda, enquanto a outra conexão não consegue transmitir nenhum pacote permanecendo com o *throughput* zero durante o tempo em que as duas conexões estão ativas.

Este trabalho propõe uma solução para lidar com estes problemas.

Palavras chaves: DFWMAC, *Wireless***,** MAC, TCP, Instabilidade, Injustiça, *Backoff, Throughput*, *ad hoc, Multihop*

Abstract

The MAC IEEE 802.11 Protocol DFWMAC (Distributed Foundation Wireless Access Control) was standardized for use in LANs wireless and has been used to test and to simulate LANs wireless ad hoc multihop LANs. This protocol has presented problems when we work with ad hoc multihop networks. This problem is evident when we submit TCP (Transmission Control Protocol) traffic between two stations. Since it is a distributed control protocol, the transmission decision is made per the proper stations in accordance with the functioning of the DFWMAC. Furthermore it has other problem such as, the hidden and terminal exopsed terminal that can occur compromising in a significant wa y the traffic of TCP connections. Associated with the problems of the hidden terminal and exposed terminal, the Binary Exponential of Backoff (BEB) algorithm contributes so that this protocol does not function adequately in ad hoc multihop networks.

The combination of all these problems is the degradation of throughput of TCP generating instability and unfairness in the access to the shared medium. The instability is evident when the variation of throughput is very high in very short intervals of time. This can be seen with only one TCP connection between two stations. The unfairness problem aldo appears when submitting two simultaneous connections TCP, where one able to transmit packets of data to a high rate, using all the bandwidth, while the other TCP connection remains unable to transmit no packets avaliable during the time where the two connections are active.

A solution is proposed to deal with these problems.

Words keys: *DFWMAC, Wireless, MAC, TCP, Instability, Injustice, Backoff, Throughput, ad hoc, Multihop*

Sumário

Lista de Figuras

Lista de Gráficos

Lista de Tabelas

Capítulo 1

Introdução

1.1 - Apresentação

Para se construir uma rede, dois fatores básicos precisam ser levados em consideração: equipamentos transmissores / receptores e o meio de transmissão. Esses fatores sofreram ao longo do tempo várias evoluções tecnológicas, contribuindo decisivamente para o crescimento acentuado que chegamos hoje na área de redes e telecomunicações. O meio de transmissão evoluiu do fio metálico de cobre para o uso do ar passando pela fibra ótica, infra-vermelho, radiodifusão etc. A construção de uma rede *wired* (com fio) implica na ligação física entre todos os equipamentos que a compõem. Isto significa a passagem de uma quantidade razoável de cabos, que podem se estender por quilômetros de distância, sem contar a dificuldade de passagem de cabos por alguns locais onde a infra-estrutura física-geográfica não é das melhores ou às vezes inviável a sua utilização. As redes sem fio romperam com estas barreiras oferecendo como alternativa um outro meio de transmissão presente em todo o planeta, o ar, apesar de não apresentar confiabilidade em alguns casos.

As transmissões via satélite [51] já não são mais nenhuma novidade e se apresentam como uma alternativa viável para transmissões de longa distância onde não há infra-estrutura física pronta ou dificuldade em instalá-la seja por problema físico ou inviabilidade financeira. Os satélites estão sendo utilizados em transmissões de dados, TV, telefonia fixa e celular [7] além de outros serviços. A telefonia celular minimizou os limites geográficos, possibilitando a comunicação imediata entre duas ou mais pessoas em qualquer lugar do mundo.

As primeiras redes locais sem fio apresentavam baixa interoperabilidade, pois cada rede possuía um conjunto de características único, já que os fabricantes construíam suas redes conforme seus próprios critérios que fossem julgados como tecnicamente corretos ou que fossem de encontro a seus interesses pessoais. Por este motivo é possível encontrar redes com vazões e alcances diferentes, empregando diversos critérios de segurança particulares de cada organização. Esta falta de padronização emperrou o desenvolvimento e, sobretudo, a adoção de redes locais sem fio. Somente em maio de 1991 [4] foi submetido ao IEEE (*Institute of*

Electrical and Electronics Engineers), organização responsável pela elaboração dos padrões adotados em redes locais e metropolitanas, agrupadas dentro da família IEEE 802, um pedido de autorização para formar o Grupo de Trabalho 802.11, cujo objetivo era definir uma especificação para conectividade sem fio entre estações de uma área local. A medida que se elaborava o padrão, os fabricantes de rede passaram a formular planos de migração de seus produtos, de acordo com as exigências feitas pela norma 802.11. O atraso na elaboração do padrão aliado a um mercado aquecido, determinou que muitos produtos fossem lançados no mercado, mas com garantias de transição suave para as especificações do padrão IEEE 802.11 [1].

A partir dessa padronização, houve um considerado crescimento tecnológico de transmissão *wireless,* surgimento de vários fabricantes de equipamentos, tendo com conseqüência, aumento da velocidade de transmissão, barateamento e maior facilidade em adquirir produtos com essa tecnologia. Para se ter uma idéia desse crescimento, as velocidades de transmissão passaram da modesta faixa de Kbps para Mbps, podendo chegar até 54 Mbps no padrão IEEE 802.11a [2] que opera na freqüência aberta de 5 Ghz. Esses fatores juntos terminaram por viabilizar a implantação das redes locais sem fio em ambientes corporativos.

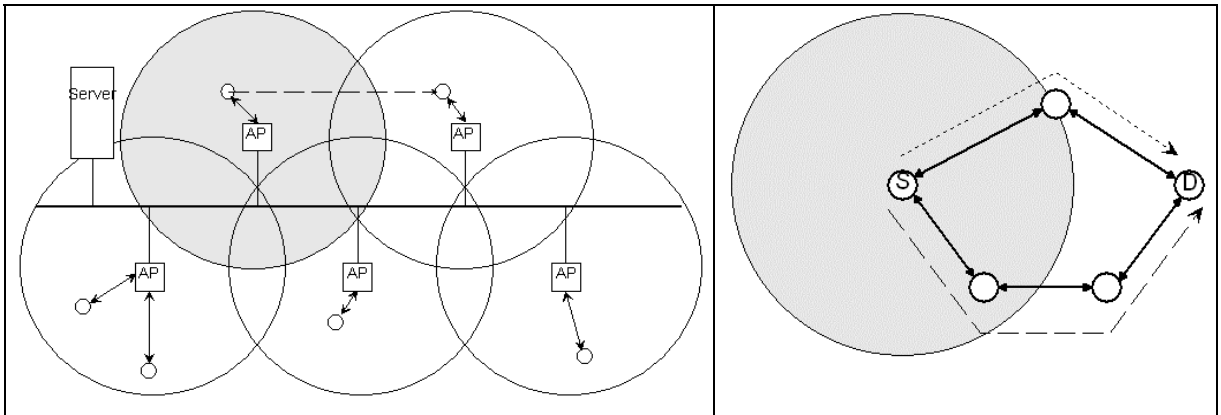
As redes locais já conquistaram seu espaço já faz algum tempo, sobretudo as redes locais com fio que vêm crescendo rapidamente desde 1971 com as redes *Ethernet* [52]. Na década de 90 começaram a aparecer as primeiras redes locais sem fio que trouxeram muito da experiência adquirida com as redes locais com fio. Dois fenômenos consolidados ao longo da década de 90, devolveram às redes locais sem fio grande interesse em pesquisa e desenvolvimento tecnológico, sendo eles miniaturização e comunicação pessoal sem fio [4].

A partir do lançamento do padrão IEEE 802.11, o uso das redes locais sem fio aumentou substancialmente seu crescimento na última década. Surgiram então duas configurações para redes locais sem fio, a primeira que tem como mecanismo de comunicação entre duas estações um ponto de acesso (*Access Point)* comum a todas as estações. Toda comunicação entre as estações é feita através do ponto de acesso (AP). Veja a Figura 1.1a [71]. A segunda é chamada de rede *ad hoc,* que é

desprovida de qualquer equipamento comutador de transmissão entre duas estações. Cada estação pode comunicar-se diretamente com outra estação sem a necessidade um ponto de acesso comum a elas. Veja a Figura 1.1b [72]. Até o final de 2002 já tínhamos mais de 2.000.000 (dois milhões) de placas de rede instaladas [4].

Uma rede *ad hoc* móvel é um conjunto de nós móveis formando redes dinâmicas, autônomas e independentes de qualquer infra-estrutura. Considerando a sua topologia, mobilidade e limite de potência de transmissão, uma determinada estação pode não alcançar diretamente outra estação durante uma transmissão, devido a distância que uma estação pode estar da outra. Assim sendo a transmissão entre estações deverá fazer uso de uma propriedade das redes *ad hoc* que é o *multihop*, ou seja, cada estação se comporta como estação e roteador fazendo a propagação do sinal até a estação destino. Este roteamento torna-se eventualmente complicado em razão da mobilidade das estações.

Estaremos estudando nesta dissertação as duas configurações de redes locais sem fio, contudo, o foco principal da nossa pesquisa serão as redes *ad hoc multihop* sem a presença da mobilidade. Onde veremos o funcionamento do subnível MAC *Medium Access Control*) e os problemas envolvendo seu principal protocolo de acesso ao meio, o DFWMAC.



a) Rede com infra-estrutura | **b)** Rede Ad Hoc Multihop

Figura 1.1 – Configurações de Redes Sem Fio

1.2 - Motivação

O meio físico além de ser caro ainda, é bastante escasso. A busca por alternativas que proporcionassem novos mecanismos de conexões de redes, levou à descoberta de um meio de transmissão de dados alternativo e que oferece facilidades de instalação física de redes, dispensando assim em quase a sua totalidade, a presença do cabo. Este meio é o ar e as redes formadas que a utilizam são as redes sem fio.

As redes locais sem fio com ponto de acesso já são usuais e podem ser vistas em funcionamento prático facilmente. Um exemplo pode ser encontrado no GPRT (Grupo de Pesquisa de Rede em Telecomunicações), grupo de pesquisa da UFPE (Universidade Federal de Pernambuco), onde parte das estações estão interligadas via um ponto de acesso que se interconecta ao resto da rede local *Ethernet*. Outro exemplo é a universidade de Taubaté no Vale do Paraíba (SP) que possui um complexo de escolas superiores, a Universidade de Taubaté e centros de pesquisa, todos localizados em prédios separados e fisicamente distantes. A opção pela tecnologia sem fio foi viável e funcional. Um dos problemas deste tipo de rede é decorrente da sua própria topologia que centraliza todas as comunicações entre as estações através do ponto de acesso. O escopo dessa pesquisa não é exatamente este tipo de rede, mas sim uma rede sem infra-estrutura para estabelecimento de comunicação entre dois pontos. Uma rede onde a comunicação pode ser feita entre estações sem a necessidade de um nó central da rede, **uma rede *ad hoc multihop.***

O protocolo MAC IEEE 802.11 usado pelas redes *multihop ad hoc* sem fio é o DFWMAC (*Distributed Foundation Wireless Access Control)*. Existem sérios problemas de performance de tráfego TCP (*Transmission Control Protocol*) quando usado neste tipo de rede. Os problemas ocorrem em algumas situações decorrentes da topologia da rede, em determinados momentos, conforme veremos nos capítulos seguintes. Tais problemas podem chegar a comprometer a utilização de redes locais sem fio *ad hoc multihop* em larga escala.

Os benefícios gerados por uma rede sem fio em relação às redes com fio são muitos, dentre eles podemos citar:

- **Mobilidade** – Sistemas de redes sem fio podem oferecer aos usuários, acesso a informação em tempo real em qualquer lugar dentro de suas organizações. O usuário poderá se locomover recebendo as informações em tempo real sem a necessidade de conectar o seu equipamento a nenhuma tomada;

- **Instalação Rápida e Simples** – Instalar uma rede sem fio pode ser uma tarefa rápida e fácil, além de eliminar a necessidade de passagem de cabos através de paredes e andares, o que em certas ocasiões torna-se financeiramente caro e inviável, além de trabalhoso;

- **Flexibilidade –** A informação pode chegar em locais onde a rede sem fio não consegue chegar ou não foi projetada para chegar até determinado local. Outra vantagem da flexibilidade é a facilidade em acessar informações independentemente da infra-estrutura física do local. Evidentemente respeitando os limites do padrão;

- **Custo Reduzido –** os custo envolvidos na instalação de uma rede sem fio são muito menores que os de uma rede com fio. Além disso, o ciclo de vida de uma rede sem fio é bem maior, que a de uma rede com fio, pois o cabo pode quebrar ou sofrer algum dano com o passar do tempo;

- **Escalabilidade –** Redes sem fio podem ser configuradas segundo diversas topologias de acordo com as necessidades. Configurações podem ser alteradas facilmente e a distância entre estações adaptadas desde poucos usuários até centenas.

Considerando todas as vantagens que uma rede local sem fio pode proporcionar, estudar, analisar e até mesmo propor soluções para os problemas que a impedem de crescer mais ainda, torna-se um grande e importante desafio. Acreditamos na

viabilidade das redes locais sem fio, não em substituição às com fio, mesmo porque elas podem completar uma a outra, mas porque ela pode oferecer todos os serviços que uma rede com fio oferece além de outros, como foi citado acima. Por isso este trabalho concentra-se em estudar este tipo de rede. Com certeza isso não irá encerrar todas as pesquisas, porque elas estão apenas começando. Este é o primeiro e fundamental passo para um estudo futuro e bem mais aprofundado, afinal de contas, as pesquisas não param.

O objetivo desse trabalho é fazer um estudo aprofundado sobre o problema existente em redes *ad hoc multihop*, verificando e avaliando soluções e trabalhos relacionados, no sentido de apresentar alternativas que possam amenizar ou contribuir para a resolução definitiva do problema. Espera-se, ao final, ter contribuído de alguma maneira para difundir o conhecimento a cerca das redes locais sem fio *ad hoc multihop.*

1.3 - Organização da Dissertação

Esta dissertação encontra-se assim organizada:

O capítulo 1 faz um breve histórico do surgimento e padronização das redes locais sem fio, definindo de maneira bem geral o seu funcionamento. Em seguida, é apresentada a motivação para o desenvolvimento desta dissertação, onde, antecipamos o problema a ser estudado. Por fim descrevemos a organização da dissertação.

O capítulo 2 faz o levantamento do estado da arte, onde será mostrado detalhadamente o padrão 802.11, sua estrutura, componentes e protocolos de nível MAC. Este capítulo apresenta o funcionamento do protocolo MAC em estudo.

O capítulo 3 apresenta o problema existente em redes sem fio *ad hoc multihop*, a ser estudado através de simulações, onde mostra que o protocolo TCP tem limitações de performance quando usado neste tipo de rede.

O capítulo 4 propõe prováveis e possíveis abordagens que podem ser feitas para a resolução do problema. As análises foram feitas também baseadas em simulações. O objetivo dessas propostas é apontar soluções que possam levar a soluções definitivas ou amenizar o problema descrito no capítulo 3.

O capítulo 5 conclui esta dissertação, mostrando as suas contribuições e apontando possíveis pesquisas para o futuro a fim de dar continuidade na busca por uma solução definitiva.

O capítulo 6 traz as referências bibliográficas utilizadas no desenvolvimento dessa dissertação e que foram fundamentais para o levantamento e embasamento das conclusões alcançadas.

Capítulo 2

O Estado da Arte em Redes 802.11

Neste capítulo apresentaremos os aspectos relacionados com o estado da arte do padrão IEEE 802.11. Serão tratados aspectos relativos ao padrão IEEE 802.11 envolvendo toda a sua arquitetura, serviços, interfaces e protocolos. Abordaremos os dois níveis definidos pelo padrão, que são os níveis MAC e físico (PHY). Este capítulo traz uma boa idéia do que é composto e como funciona este padrão. Para isso iremos conhecê-lo da arquitetura até as suas funcionalidades principais como o esquema de *backoff* exponencial adotado no padrão.

2.1 - O Padrão IEEE 802.11

Este é o padrão definido para as redes locais sem fio (*Wireless Local Area Network – WLAN*). Este padrão foi definido pela IEEE em 1990, mas seu projeto ficou praticamente parado por cerca de 7 (sete) anos em virtude das características técnicas do padrão que não ofereciam subsídios necessários para que o projeto saísse de fato do papel. Estas características giravam em torno da baixa taxa de transferência que era muito inferior a 1 Mbps. Com o aumento dessa taxa, hoje 11 Mbps no padrão 802.11b [73], essa tecnologia passou a ser vista com sendo promissora atraindo investimentos para construção de equipamentos na área de comunicação sem fio. Soluções proprietárias existiram antes da definição desse padrão, entretanto por ser um padrão aberto, ele foi bastante aceito já que oferecia vantagens como: interoperabilidade, baixo custo, demanda de mercado e confiabilidade de projeto.

2.1.1 - Componentes do Padrão IEEE 802.11

São vários s componentes de uma rede sem fio com infra-estrutura, que interagem entre si objetivando oferecer uma rede sem fio que suporte mobilidade entre as estações de maneira transparente para os níveis superiores. A seguir iremos conhecer cada um desses componentes e qual o seu papel dentro da arquitetura.

2.1.1.1 - Access Point (AP)

Um ponto de acesso é definido como sendo o ponto responsável pela comunicação entre as estações que compõem a rede. O ponto de acesso funciona com uma espécie de comutador, centralizando toda a comunicação. Isto significa que uma estação não transmite diretamente para outra estação e sim para o AP que retransmite a informação via *broadcast* até chegar a estação destino. O AP é utilizado apenas em redes com infra-estrutura, no caso de uma rede *ad hoc,* o AP não é necessário. Veja a Figura 2.1 [1].

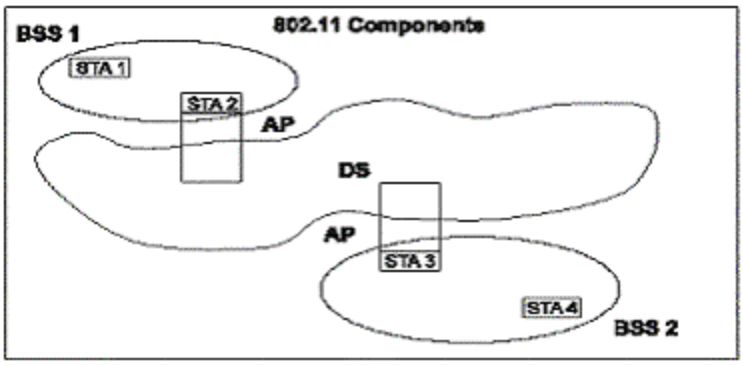


Figura 2.1 – Ilustração de AP

2.1.1.2 - Basic Service Area (BSA)

É uma área delimitada onde dentro da qual um conjunto de estações podem comunicar-se. Esta área é conhecida como célula e define a área de cobertura de uma rede. Cada estação possui um transmissor com potência definida. Quanto maior for a potência maior será a distância que uma estação pode alcançar. Evidentemente esta distância necessita estar dentro do permitido pelo padrão que é de 250m [1].

2.1.1.3 - Basic Service Set (BSS)

É definido como um conjunto de estações ou STAs que se comunicam através de um mecanismo de transmissão em um BSA. Em outras palavras, são todas as estações que estão dentro de uma célula e que podem comunicar-se mutuamente. Veja a Figura 2.2 [1].

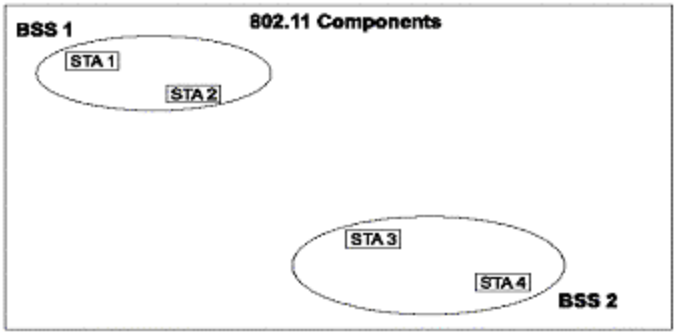


Figura 2.2 - Dois BSS's

2.1.1.4 - Distribution System (DS)

É o mecanismo utilizado para interligar dois BSSs e integrar LANs (*Local Area Network*) criando desta forma um outro componente da arquitetura, o, ESS, que será visto abaixo. Esta interligação é feita fisicamente através dos APs e pode ser feita utilizando vários meios de comunicação. Veja a Figura 2.1 [1].

2.1.1.5 - Extend Service Area (ESA)

A área coberta por várias BSAs forma uma ESA. Para que a área de cobertura de comunicação não se restrinja somente a sua BSA, foi criado a ESA que interliga várias BSAs através dos APs.

2.1.1.6 - Extend Service Set (ESS)

Representa um conjunto de um ou mais BSSs conectados através de um DS integrando desta forma LANs de diferentes BSS. Isto possibilita o *roaming* entre BSS distintos provocados pela mobilidade das estações em uma rede sem fio. Veja a Figura 2.3 [1].

A identificação da rede ocorre da seguinte forma: cada um dos ESSs recebe uma identificação chamada de ESS-ID; dentro de cada um desses ESSs, cada BSS recebe uma identificação chamada de BSS-ID. Então o conjunto formado por esses dois identificadores (ESS-ID e BSS-ID), formam o endereço da rede (*Network ID*) sem fio de padrão 802.11 [6].

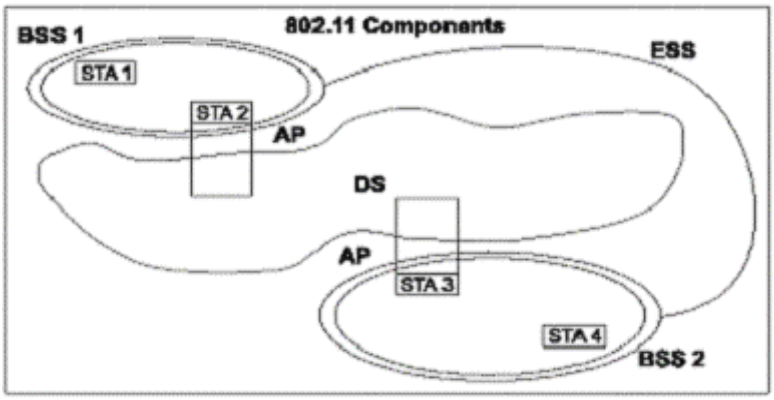


Figura 2.3 – ESS interligando dois BSSs

2.1.2 - Serviços do Padrão IEEE 802.11

O IEEE 802.11 não especifica detalhes de implementação do DS. O padrão especifica os serviços. Os serviços são associados a diferentes componentes da arquitetura. Eles são divididos em duas categorias e utilizadas no subnível MAC IEEE 802.11.

2.1.2.1 - Serviços de Sistema de Distribuição (D*istribution System Service* (DSS))

São oferecidos pelo DS, por este motivo recebem este nome. Eles são compostos dos seguintes serviços:

- Distribuição;
- Associação;
- Disassociação;
- Integração;
- Reassociação.

Veremos logo abaixo cada um deles.

A) Distribuição

Este é o principal serviço utilizado pelas STAs do IEEE 802.11. Ele é invocado por cada mensagem de dados ou de um STA operando dentro de um ESS, isto quando o quadro é enviado via DS. O serviço de distribuição acontece via DSS. É através deste serviço que estações localizadas dentro de BSS's diferentes troquem informações.

B) Associação

É um serviço necessário na entrega de mensagens dentro de um DS. Este serviço necessita conhecer qual deve ser o AP para acessar uma determinada STA IEEE 802.11. Esta informação é provida pelo DS. Antes da permissão de um STA para enviar mensagens de dados via AP, ele obrigatoriamente será associado a um AP. A qualquer instante, um STA pode ser associado a não mais que um AP. Desta forma asseguramos que o DS pode determinar uma única reposta para a pergunta: "*Qual AP está servindo o STA X?".*

Uma vez que uma associação for completada, um STA pode fazer uso total de um DS para se comunicar. Associação é sempre inicializada pela STA móvel, não pelo AP.

C) Disassociação

Este serviço é invocado sempre que uma associação está para ser terminada e é invocado por uma das duas partes de uma associação (STA AP ou não-AP). Uma disassociação é uma notificação não uma requisição. APs podem precisar disassociar STAs para permitir um AP ser removido da rede por serviços ou outras reações.

D) Integração

O serviço de integração é responsável pela realização de tudo que for necessário para entrega de mensagens do DSM (*Distribution System Medium*) para meio da rede local integrada, incluindo todas as requisições do meio e espaço de endereçamento. Mensagens recebidas por uma LAN integrada vinda de um DS para um STA IEEE 802.11, chamará este serviço antes que a mensagem seja distribuída pelo serviço de distribuição.

E) Reassociação

É um serviço invocado para mover uma associação atual de um AP para outro. Isto mantém o DS informado do mapeamento atual entre AP e STA assim como das movimentações de estações entre BSSs dentro de um ESS. Uma reassociação é sempre inicializada por um STA móvel.

2.1.2.2 - Serviços de Estações (Station Service (SS))

Estão presentes em cada estação IEEE 802.11, incluindo APs. Os SS são compostos dos seguintes serviços:

- Autenticação;
- Sem-Autenticação;
- Privacidade.

A seguir veremos cada um deles detalhadamente.

A) Autenticação

Em uma rede local cabeada, o primeiro nível de segurança é o físico. Ou seja, para que um usuário acesse uma rede local é necessário que ele esteja fisicamente conectado a mesma. Quando trabalhamos com redes sem fio, a segurança precisa ser redobrada uma vez que o meio físico não será mais o fio e sim o ar. Para isso foram criados mecanismos de autenticação que permitem somente usuários autorizados a utilizarem uma determinada rede local sem fio. O padrão IEEE 802.11 suporta vários desses mecanismos de autenticação. Se for o caso, uma rede sem fio pode trabalhar utilizando Sistemas Abertos para autenticação. Pode ser usada também autenticação por chave publica, neste caso a implementação requer uma privacidade equivalente a uma rede com fio (*WEP – Wired Equivalent Privacy)*. O processo de autenticação não é fim-a-fim ou usuário-a-usuário mas sim sempre entre estações da rede.

B) Sem-autenticação

Sem-autenticação não é uma requisição, é uma notificação. Ela é invocada sempre que uma autenticação está para ser concluída. Em um ESS, visto que a autenticação é pré-requisito para associação, o reconhecimento de uma notificação sem-autenticação obriga uma estação ser desligada (disassociada). O serviço de autenticação também pode ser invocado pela parte autenticada (STA AP ou não-AP). Quando um AP envia esta notificação para um STA associado, obriga a associação também ser terminada.

C) Privacidade

Em uma rede com fio, apenas as estações fisicamente conectadas na rede local podem ouvir o tráfego. Nas redes sem fio que possuem um meio compartilhado, o processo de privacidade precisa ser diferente e até mais eficiente. Uma estação que esteja dentro da faixa de transmissão de um transmissor/receptor pode ouvir a comunicação entre as estações. Objetivando alcançar as mesmas funcionalidades de uma rede com fio, o IEEE 802.11 oferece a capacidade de criptografar o conteúdo das mensagens que trafegam na rede.

O IEEE 802.11 especifica o uso opcional de um algoritmo de privacidade, WEP, que foi projetado para satisfazer as metas de privacidade equivalente a uma rede local com fio. A privacidade deve somente ser invocada para quadros de dados e alguns quadros de gerenciamento de autenticação. Todas as estações inicialmente iniciam seu estado de serviços como desligado em seguida são ligados os serviços de autenticação e privacidade. O algoritmo WEP será visto na seção 2.2.

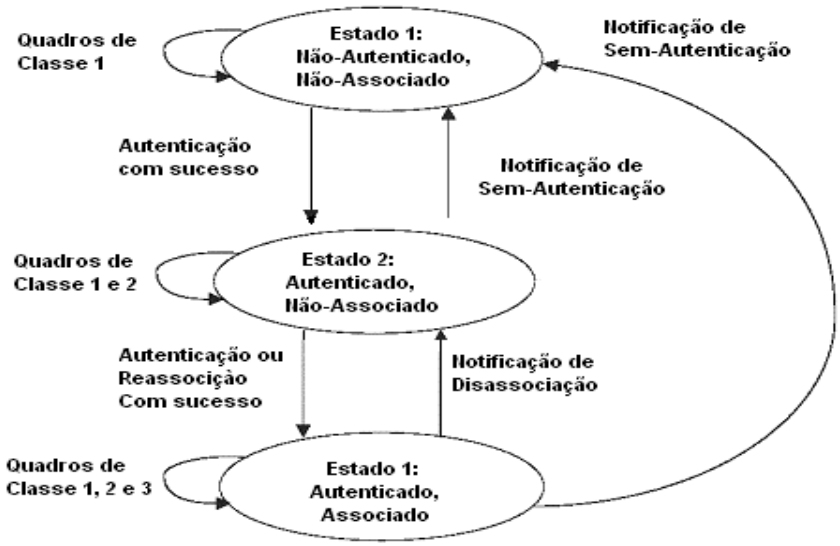


Figura 2.4 – Relacionamento entre os Serviços

Na Figura 2.4 [1], podemos visualizar o relacionamento entre os serviços descritos acima. No estado 1, a estação encontra-se sem autenticação e não associada a nenhum AP. No estado 2, a estação foi autenticada com sucesso, mas ainda não

associada a nenhum AP. Caso a autenticação não seja feita a estação receberá uma notificação de sem-autenticação. No estado 3, a estação completa o ciclo sendo associada a um AP. Caso isso não tenha sido feito com sucesso, ela volta ao estado 1, recebendo uma notificação de sem-autenticação, quando começa todo o ciclo novamente.

2.2 - O Algoritmo Wired Equivalente Privacy (WEP)

A WEP (*Wired Equivalent Privacy*) é um mecanismo de segurança do padrão IEEE 802.11 onde seu uso é opcional, utilizado para proporcionar segurança de dados equivalente à de uma rede com fios através do uso de técnicas de criptografia simples de privacidade, possibilitando desta forma que *links* de rede local sem fio tornem-se tão seguros quanto os *links* de redes com fios. A criptografia de dados WEP é utilizada para impedir:

1. Que intrusos acessem a rede utilizando equipamentos similares a rede local sem fio;
2. "Roubo" de informações que trafegam na rede. Através da WEP o administrador da rede define o conjunto das respectivas chaves de cada usuário da rede sem fio de acordo com uma seqüência de chaves definidas pela criptografia WEP. O que se vê na prática é que menos de 30% das redes WLANs são protegidas;

Quem não possuir a chave necessária, será negado o acesso. A WEP usa o algoritmo RC4 [1] com chave de 40 ou 128 bits. Quando WEP é ativado, cada estação e os pontos de acesso devem possuir uma chave. Esta chave será utilizada para criptografar os dados antes de serem transmitidos pelas estações emissoras. Quando uma estação receber um pacote não criptografado com a chave adequada, este pacote será descartado e não entregue ao *host*. Isso impedirá o acesso àrede por curiosos e pessoas não autorizadas [1].

O algoritmo WEP possui as seguintes propriedades:

- **razoavelmente forte:** A segurança oferecida pelo algoritmo conta com a dificuldade de descoberta da chave secreta de qualquer forma de ataque com força bruta;

- **auto-sincronizável:** Esta propriedade de auto-sincronização é importante pela perca constante de *link* já que na maioria dos casos o serviço oferecido é de melhor esforço (*best effort*) [54];

- **Eficiente:** além da sua eficiência, ele pode ser implementado tanto em *software* quanto em *hardware;*

- **Opcional:** a sua utilização em redes 802.11 não é obrigatória.

Não nos aprofundaremos neste tópico uma vez que o foco principal de nosso trabalho não é esse. Entretanto demais informações complementares poderão ser encontradas em [1].

2.3 - O Subnível MAC

O modelo de referência OSI (*Open System Interconection*) [6][7], formaliza a organização de uma rede em 7 (sete) camadas, sendo elas na ordem de baixo para cima: física, enlace, rede, transporte, seção, apresentação e aplicação. A camada de enlace por sua vez foi dividida para efeito de organização em duas subcamadas o MAC (*Medium Access Control*) e o LLC (*Logical Link Control*). A subcamada MAC relaciona-se com o nível físico e o LLC com o nível de rede. Em redes sem fio a subcamada MAC foi padronizada pela IEEE. Veja A Figura 2.5 [6].

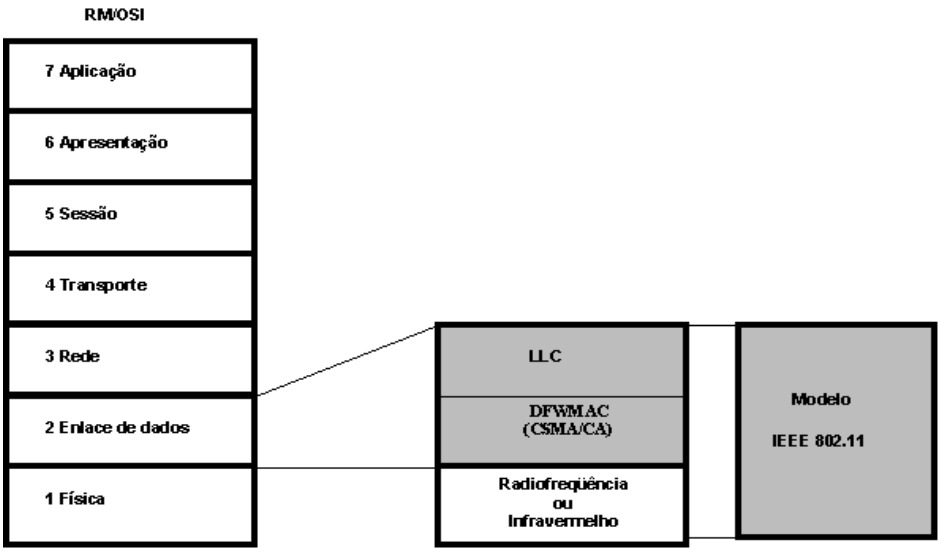


Figura 2.5 – Equivalência de Camadas do Modelo OSI e o Padrão 802.11

O subnível MAC suporta as seguintes primitivas de serviço:

A) MA-UNITDATA.request, utilizada para requisitar algum serviço do MAC;
B) MA-UNITDATA.indication, utilizada para indicar uma requisição;
C) MA-UNIDATA-STATUS.indication, informa a situação da requisição.

Veja que o MAC é não orientado a conexão. Esta seção tem o objetivo de conhecer esta subcamada de acordo com a sua especificação formal em [1]. Veremos a sua estrutura interna e como ela funciona.

2.3.1 - O Formato do Quadro MAC

Todo quadro que trafega no nível de enlace em redes sem fio segue o formato descrito na Figura 2.6. Os campos *Address 2, Address 3, Sequence Control, Address 4 e Frame Body,* são representados em apenas alguns tipos de quadros.

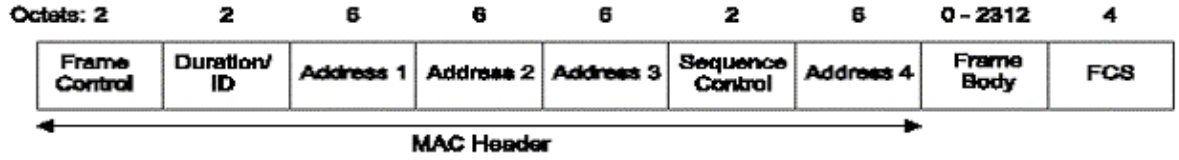


Figura 2.6 – O Formato do Quadro MAC

A seguir iremos descrever cada um dos campos que compõem este quadro.

A) Frame Control

Este campo possui vários outros subcampos descritos abaixo com o seguinte formato:

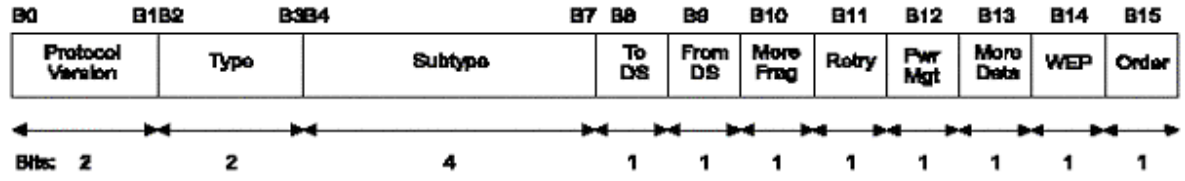


Figura 2.7 – O Formato do Campo Frame Control

- **Protocol Version**

É um campo composto de 2 (dois) bits que controla a versão do protocolo em uso pelo padrão. Para esta versão atual o valor desse campo é 0 (zero). Cada vez que um quadro for recebido este campo será checado. Se o valor desse campo for maior que 0 (zero) o quadro será descartado.

- **Type e Subtype**

Estes campos servem para identificar o tipo de quadro. Existem três tipos de quadro definidos pelo MAC, sendo eles: controle, dados e gerenciamento. Para cada um desses tipos existem vários subtipos definidos a combinação desses bits indica o tipo em uso. Veja a Tabela 2.1

Valor do Tipo (b3 b4)	Descrição do Tipo	Valor do subtipo b7 b6 b5 b4	Descrição do Subtipo
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110-0111	Reserved
00	Management	1000	Beacon
00	Management	1001	Announcemente traffic indication message(ATIM)
00	Management	1010	Disassociation
00	Management	1011	Autentication
00	Management	1100	Deauthentication
00	Management	1101-1111	Reserved
01	Control	0000-1001	Reserved
01	Control	1010	Power Save (PS)-poll
01	Control	1011	Request to Send (RTS)
01	Control	1100	Clear to Send (CTS)
01	Control	1101	Acknowledgment (ACK)
01	Control	1110	Contention-Free (CF) – End
01	Control	1111	CF-End + CF-ACK
10	Data	0000	Data
10	Data	0001	Data + CF-ACK
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-ACK + CF-Poll
10	Data	0100	Null Function (No Data)
10	Data	0101	CF-ACK (No Data)
10	Data	0110	CF-Poll (No Data)
10	Data	0111	CF-ACK + CF-Poll (No Data)
10	Data	1000-1111	Reserved
11	Reserved	0000-1111	Reserved

Tabela 2.1 – Combinações Válidas de Tipos e Subtipos

- **To DS**

É um campo de 1 bit que é inicializado com valor 1 para os tipos de quadros de dados destinados ao DS. Isto inclui todos os tipos de quadros enviados pelos STAs associados a um AP. Nos outros quadros este campo possui o valor 0.

- **From DS**

Este campo também é de apenas 1 bit e possui um valor 1 para quadros existentes dentro de um DS. Para os outros tipos de quadros ele possui o valor 0.

- **More Fragments**

É um campo de tamanho 1 que é iniciado com 1 para todos os quadros de dados ou gerenciamento que tenham outros fragmentos do atual MSDU (*MAC Service Data Unit*) ou MMPDU (*MAC Management Protocol Data Unit*) seguinte. Nos outros quadros tem o valor 0.

- **Retry**

É um campo de 1 bit que possui valor 1 para qualquer quadro de dados ou gerenciamento que tenha sido retransmitido antes do quadro anterior. Ele possuirá o valor 0 nos outros casos.

- **Power Management**

É o campo que indica se um STA utilizará o modo de gerenciamento de energia. O valor 1 indica que o STA estará neste modo. O valor 0 indicar que o STA irá ativar este modo.

- **More Data**

É um campo de 1 bit usado para indicar o tipo de quadro transmitido, sendo dados ou gerenciamento transmitido de um AP para um STA. Quando possui o valor 1 indica que os quadros foram transmitidos por um STA *contention-free (CF)-Pollable* para o ponto de coordenação (PC).

- **WEP**

É um campo que possui o valor 1 se o campo *Frame Body* contém informações que podem ser processadas pelo algoritmo WEP. O campo *Frame Body* será tratado mais àfrente.

- **Order**

É um campo que possuirá o valor 1 em qualquer quadro que contenha MSDU ou fragmentos. Nos outros casos ele possuirá o valor 0.

B) Duration / ID

Este campo possui um tamanho de 16 bits. Estes bits estão divididos da seguinte forma:

- Identifica a portadora de associação de identidade (AID) de uma estação que transmite um quadro nos 14 bits menos significantes, com os 2 bits mais significantes ambos com o valor 1. O valor do AID está na faixa de 1-2007.

- Em todos os outros quadros, este campo contém o valor da duração definido para cada tipo de quadro. Para os quadros transmitidos durante o período livre de disputa(contenção), o valor desse campo é de 32.768.

Quando o valor desse campo for menor que 32.768, ele será utilizado para atualizar o NAV (*Network Allocation Vector*).

C) Campos Address1, Address2, Address3, Address4

Estes campos são usados para indicar o BSSID, endereço origem, endereço destino, endereço da estação que está transmitindo e o endereço da estação que está recebendo, respectivamente.

D) Sequence Control

É um campo composto de 16 bits consistindo de dois sub-campos, sendo eles: *Sequence Number*, que compreende os 4 primeiros bits e o *Fragment Number,* formado pelos 12 bits restantes. Este campo é usado pelo receptor para determinar como ele iria receber a informação do nível 3.

E) Frame Body

É um campo de tamanho variável que contém informações especificas para um tipo e subtipo de quadro individual. No mínimo ele possuirá o tamanho zero.

F) FCS (Frame Check Sequence)

É um campo formado por 32 bits CRC (*Check Redundancy Cycle*). Ele é calculado sobre todos os campos do cabeçalho MAC e do campo *Frame Body.* A forma de calcular o FCS é baseada no seguinte padrão de geração polinomial de ordem 32:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

Neste campo é colocado o resto da divisão polinomial

2.3.2 - O Formato dos Quadros de Controle do MAC

Os quadros de controle são utilizados durante a transmissão de dados e têm a função de efetuar alguns controles, como por exemplo autorização para transmitir ou reconhecimento de recebimento de quadros por um receptor. Veremos a seguir alguns desses quadros, dentre eles podemos destacar:

- **RTS** (*Request do send*)
- **CTS** (*Clear to Send*)
- **ACK** (*Acknowledgment*)

2.3.2.1 - Request to Send (RTS)

Este quadro é utilizado sempre que o transmissor desejar utilizar o meio por um determinado intervalo de tempo para transmissão. Para isso o transmissor envia este pacote ao receptor para verificar se o meio está livre. Se o meio estiver livre, o transmissor começará a transmitir. O formato desse quadro segue na Figura 2.8 [1]

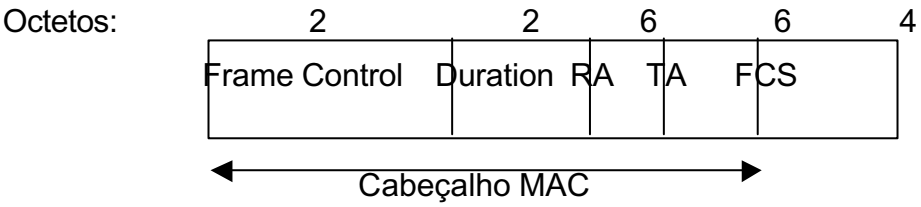


Figura 2.8 – O Formato do Quadro RTS

Onde,

RA ➔ é o endereço do receptor

TA ➔ é o endereço do transmissor

Duration ➔ é a duração de tempo solicitada para transmissão em microsegundos.

2.3.2.2 – Clear to Send (CTS)

Em resposta a um RTS enviado por um transmissor, o receptor responde com este quadro de controle, indicando que o meio estará disponível pelo intervalo solicitado. O formato desse quadro é visto na Figura 2.9.

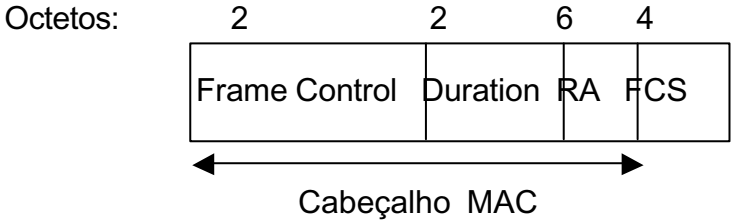


Figura 2.9 – O Formato do Quadro CTS

O RA do quadro CTS é copiado do campo TA para o quadro RTS imediatamente anterior ao CTS respondido. O valor da duração é obtido do último campo da duração do frame RTS menos o tempo requerido para transmitir o quadro CTS.

2.3.2.3 – Acknowledgment (ACK)

Este quadro de controle informa para um transmissor que o quadro enviado chegou ao receptor. Em outras palavras, este tipo de quadro é usado para confirmação de outros quadros, especialmente os quadros de dados. O formato desse quadro está ilustrado na Figura 2.10 [1].

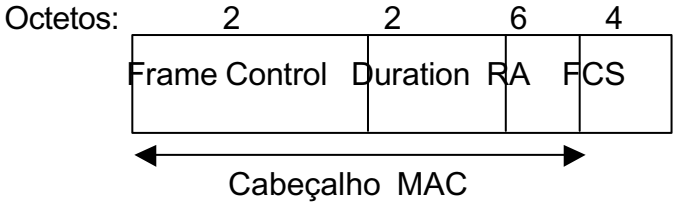


Figura 2.10 – O Formato do Quadro ACK

2.4 - A Arquitetura do MAC IEEE 802.11

O protocolo padrão para arquitetura de redes sem fio é o DFWMAC (*Distributed Foundation Wireless Access Control*). Ele suporta dois métodos de acesso, um distribuído, o DCF (*Distribution Coordination Function*) e outro centralizado, o PCF (*Point Coordenation Function*). O método distribuído é básico e obrigatório para todas as estações e pontos de acesso, nas configurações *ad hoc* e com infra-estrutura. O método centralizado e opcional, não sendo obrigatório o seu uso. Veja a Figura 2.11 [1]. Os dois métodos de acesso podem coexistir. Na verdade o método de acesso distribuído forma a base sobre a qual é construído o método centralizado [6]. A seguir veremos como funcionam os dois métodos.

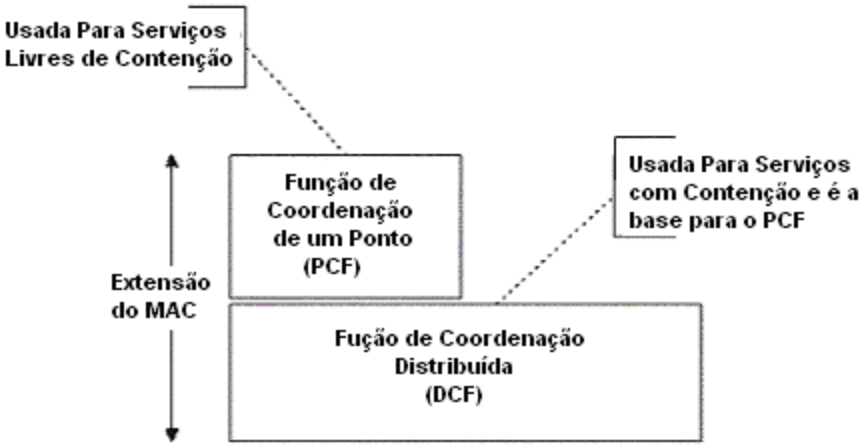


Figura 2.11 – A Arquitetura do MAC IEEE 802.11

2.4.1 – A Função de Coordenação Distribuída (DCF)

É o principal método de acesso ao meio do MAC IEEE 802.11. É um mecanismo de acesso múltiplo com detecção de portadora evitando colisão chamado CSMA/CA (*carrier sense multiple access with collision avoidance)*. O CSMA/CA é utilizado em razão da detecção de colisão ser muito difícil por assumir que todas as estações ouvem as outras, por requerer um rádio *full-duplex* de custo elevado e porque a

taxa de erro de bit na camada MAC do IEEE 802.11 é na faixa de 10^{-5} [3] [4].
Existem dois tipos de DCF definidos no padrão 802.11, o primeiro é obrigatório e está baseado no CSMA/CA, o outro é opcional e baseado em quadros de controle para transmissão RTS e CTS. A seguir veremos o funcionamento de cada um deles.

2.4.1.1 - DCF Baseado em CSMA/CA

O acesso básico ao meio utilizando o CSMA/CA funciona da seguinte maneira: sempre que uma estação deseja transmitir algum quadro, ela precisa antes ouvir (sentir) o meio, ou seja, detectar a portadora. Caso o meio esteja livre após um determinado intervalo de tempo denominado DIFS (*Distributed Interframe Space*), a estação está autorizada a iniciar a transmissão. Caso contrário, a transmissão será adiada por um tempo iniciando-se então o processo de *backoff*. O esquema de *backoff* será detalhado na seção 2.5. O tempo em que a estação irá esperar será aleatório e uniformemente distribuído variando de zero ao tamanho máximo da janela de contenção (CW). Passado este intervalo de tempo, se o meio ainda estiver ocupado, esta estação perdeu o ciclo e precisa esperar pela próxima chance, ou seja, até o meio ficar livre novamente por um período de pelo menos DIFS. Se passado o intervalo de tempo aleatório, o meio ainda estiver inativo, esta estação estará autorizada a acessar o meio.

Este tempo de espera aleatório é escolhido como sendo um múltiplo de um *slot* de tempo da janela de contenção. O *slot* é derivado do atraso de propagação do meio, atraso de transmissão e outros parâmetros dependentes do meio físico.

A estação receptora por sua vez utiliza o método de verificação cíclica (CRC) para detectar erros. Caso o quadro pareça sem erros, a estação receptora envia um quadro de reconhecimento ACK. Esse quadro é enviado durante um tempo SIFS (*Short Interframe Space*) após a recepção do quadro anterior. Veja a Figura 2.12 [3]

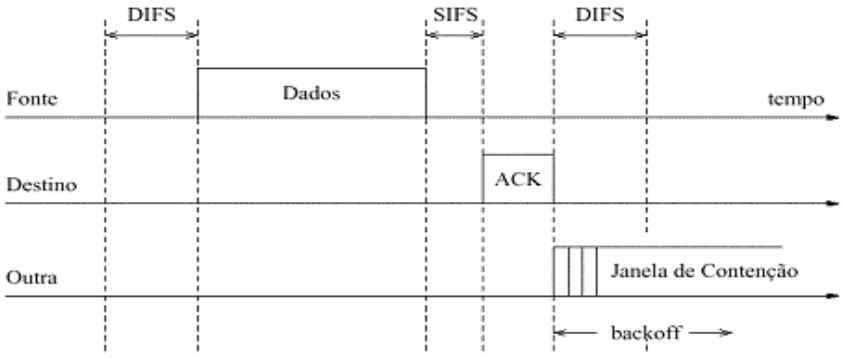


Figura 2.12 – Esquema Básico de Acesso ao DCF

Veja que este mecanismo básico não é justo, pois cada estação tem as mesmas chances de transmitir no próximo ciclo. Para criar o mecanismo mais justo, o IEEE 802.11 acrescentou o contador de *backoff.* Da mesma forma que antes, cada estação escolhe um tempo aleatório. Se uma determinada estação não conseguir acessar o meio no primeiro ciclo, ela pára seu contador de *backoff*, espera o canal ficar livre por um período DIFS enquanto o seu contador começa a diminuir até expirar. Quando após isso, essa estação acessa o meio. Desta forma esta estação não vai ter que escolher um tempo aleatório novamente. Portanto as estações que estão tentando acessar o meio a alguns ciclos levam vantagem em relação às estações que venham querer acessá-lo.

Mesmo com esse refinamento, este esquema ainda apresenta problemas. Dependendo do tamanho da janela de contenção, os valores gerados randomicamente podem ser próximos com pequena diferença, causando assim muitas colisões ou podem ser distantes, causando atraso. Por isso foi adicionado um outro mecanismo que tenta adaptar o tamanho da janela de contenção com o número de estações que estão tentando acessar o meio. Cada vez que ocorrer uma colisão, a janela de contenção será dobrada não ultrapassando o valor máximo. Quanto maior for a janela, menor será a probabilidade de ocorrer colisão, pois existirá um maior número de opções para se escolher um tempo de *backoff*. Além disso, para um tráfego pequeno no meio, o uso de uma janela menor minimiza o atraso. Esse algoritmo é chamado de *Backoff E*xponencial Binário (BEB).

O intervalo de tempo entre quadros é chamado de IFS (*Interframe Space*). Para determinar se o meio está ocioso por um intervalo de tempo especificado, uma estação fará uso da função de detecção de portadora. Foram definidos 4 (quatro) IFS's que oferecem níveis de prioridade diferentes para acessar o MAC. Abaixo será definido cada um deles e na Figura 2.13 [1] pode ser visto o relacionamento entre estes.

A) SIFS (Short Interframe Space)

SIFS é o menor dos espaços entre quadros. Ele será usado quando STAs monopolizarem o meio e necessitarem mantê-lo assim por toda a duração da seqüência de troca de quadros a ser executada. SIFS deve ser usado pelos quadros de controle ACK e CTS, quadros fragmentados MPDU (*MAC Protocol Data Unit*) e por STAs que estão respondendo a algum *polling* de um PCF.

Por definição, SIFS é menor que DIFS, ou seja, a estação receptora ouve o meio por SIFS para enviar o ACK. Caso a estação transmissora não receba o ACK, deduzirá que houve uma colisão, escalonará uma retransmissão e entrará no processo de *backoff.* Veja a Figura 2.13 [1].

B) DIFS (DCF Interframe Space)

O DIFS deve ser usado por STAs operando sobre o modo DCF para transmitir quadros de dados (MPDUs) e quadros de gerenciamento (*Management MPDU - MMPDU*). Deve ser permitido a um STA usando o DCF, transmitir se o mecanismo de detecção de portadora determinar que o meio esteja livre em um *slot* limite específico chamado TxDIFS e o tempo de *backoff* não tenha expirado. Veja a Figura 2.13 [1].

C) EIFS (Extended Interframe Space)

O EIFS deve ser usado pelo DCF sempre que o nível físico indique para o MAC que a transmissão do quadro iniciada não resultou na recepção correta de um frame completo do MAC com um valor correto do FCS (*Frame Check Sequence*). EIFS é definido para oferecer tempo suficiente para outro STA reconhecer qual foi, o STA onde o quadro foi recebido incorretamente antes que esse STA comece a transmissão. Veja a Figura 2.13 [1].

D) PIFS (PCF Interframe Space)

Este tempo é utilizado somente por STAs operando sobre o modo PCF para ganhar prioridade de acesso ao meio no inicio do período livre de contenção (CFP – *Contention-Freee Period*). Deverá ser permitido a um STA usando PCF, transmitir tráfego livre de contenção depois do mecanismo de detecção de portadora determinar que o meio esteja livre no *slot* limite chamado TxPIFS. Veja a Figura 2.13 [1].

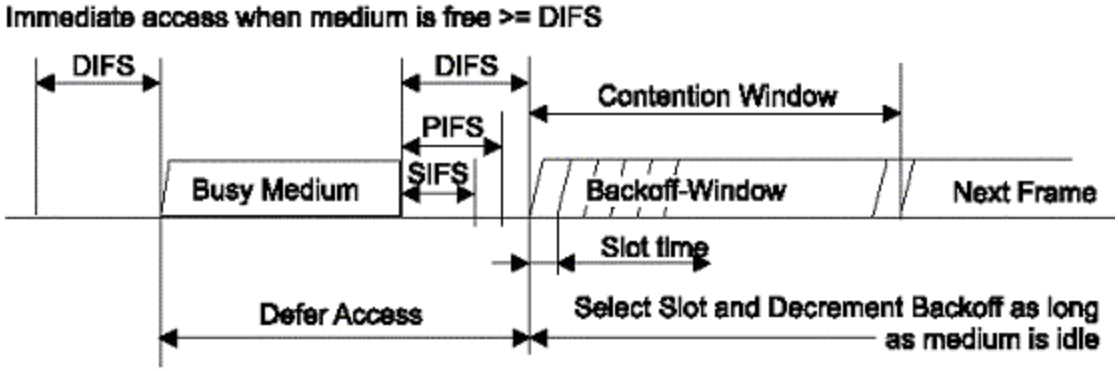


Figura 2.13 – Relacionamento entre IFS

2.4.1.2 - DCF Baseado em RTS / CTS

Este tipo inclui a utilização de quadros de controle RTS e CTS para evitar problemas gerados por terminais escondidos. O problema de terminais escondidos será detalhado no próximo capítulo. A detecção de portadora pode ser feita através de mecanismos físicos e virtuais. O mecanismo virtual usa uma distribuição de informação de reserva do meio através da troca de quadros RTS e CTS antes do envio dos dados. Os quadros de controle RTS e CTS contêm informações a respeito do nó destino e de um tempo relativo ao envio do quadro de dados e de seu respectivo quadro de reconhecimento, ACK. O uso de RTS e CTS é controlado por cada estação através de liminar de RTS, através da qual uma estação poderá não usar o RTS e o CTS, poderá sempre utilizá-los ou ainda usá-los somente na transmissão de um quadro maior que o tamanho predeterminado. Uma estação deve enviar um RTS ao receptor, após sentir o meio livre pelo por um intervalo de tempo DIFS, antes da transmissão de um quadro para reservar o meio. A probabilidade e conseqüência de colisão de um quadro RTS é bem menor e menos grave que a colisão de um quadro de dados que tem um tamanho maior que um quadro de controle como o RTS. O receptor deverá responder com um quadro CTS, após o meio estar livre por SIFS segundos, caso esteja pronto para receber dados.

Todas as estações que ouvirem o RTS, o CTS, ou ambos, irão utilizar a informação da duração relativa ao quadro de dados para atualizar o NAV, que é usado para uma detecção virtual da portadora. O NAV especifica o primeiro ponto no tempo onde a estação pode tentar acessar o meio novamente. Essa informação indica o período de tempo pelo qual uma transmissão não é iniciada pela estação. Desse modo, qualquer terminal escondido adiará sua transmissão para evitar colisões. Ao receber o quadro CTS e esperar o meio ficar livre por um tempo SIFS, o transmissor inicia o envio do quadro, assim como no modo DCF básico. Caso não receba o CTS, o transmissor entra na fase de *backoff* e retransmite o RTS. Veja a ilustração na Figura 2.14 [4].

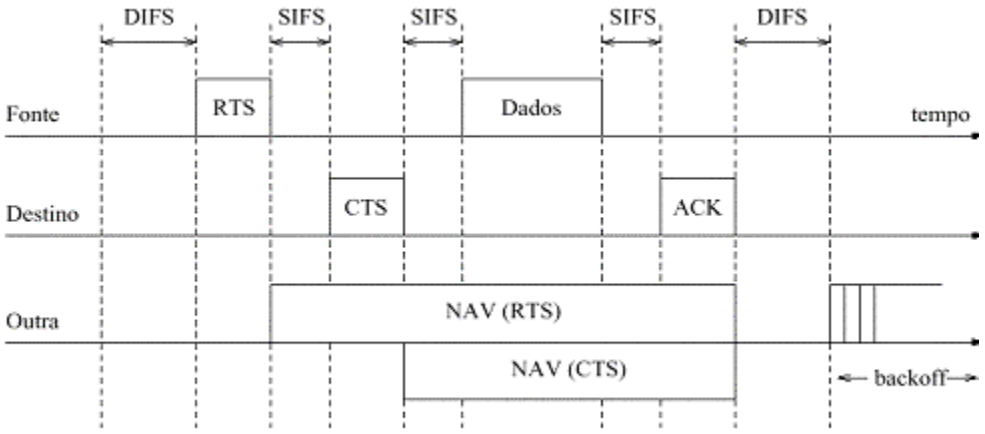


Figura 2.14 – Esquema de DCF utilizando RTS e CTS

Através desse mecanismo, colisões só podem acontecer no início do envio do quadro RTS. Eventualmente, duas ou mais estações podem iniciar a transmissão ao mesmo tempo, seja com o envio de um RTS, ou envio de dados. A utilização de RTS/CTS pode resultar em um alto *overhead*, gerando diminuição de transmissão, perda de banda passante e um *atraso* elevado. Entretanto este mecanismo é fundamental na prevenção de problemas de terminais escondidos, que geram maiores transtornos na rede.

2.4.2 – A Função de Coordenação de um Ponto (PCF)

Este é o segundo método de acesso ao MAC IEEE 802.11. Contrariamente a DCF, a sua utilização não se torna obrigatória no padrão. A grande diferença da PCF para a DCF, é que na PCF o acesso ao meio é controlado por um único ponto de acesso enquanto que no segundo cada estação é responsável pelo acesso ao meio. Assim sendo, toda vez que uma estação desejar utilizar o meio, o ponto de controle de acesso deverá verificar a utilização do meio através de consultas às estações. Caso nenhuma estação esteja transmitindo, a estação então está autorizada a transmitir sem a possibilidade de disputa(contenção) pelo meio [4].

O processo de funcionamento desse método baseia-se na utilização de um coordenador de ponto que está localizado no ponto de acesso. O coordenador divide o tempo de acesso em períodos de tempos chamados superquadros. Cada superquadro compreende um período livre de contenção chamado modo PCF e um período com contenção chamado modo DCF. O coordenador de ponto realiza consultas a cada estação para verificar a existência de transmissão. Esta consulta é feita apenas nos períodos em que as estações estão no modo PCF.

O coordenador de ponto inicia e controla o tempo livre de contenção. Ele escuta o meio por PIFS segundos e então começa um período livre de contenção através da difusão de um sinal de *beacon* (intervalo de tempo entre transmissores) [1]. Como, por definição, PIFS é menor que DIFS, nenhuma estação pode começar a enviar dados no modo PCF antes do coordenador de ponto. Todas as estações adicionam a duração máxima do período de contenção ($CFP_{maxduration}$) aos seus respectivos NAVs. O período livre de contenção pode terminar a qualquer momento através do envio de um pacote CFP_{end} pelo coordenador de ponto. Quando chega a vez de uma estação transmitir, o coordenador de ponto envia um pacote de dados caso exista algum a ser enviado dentro de um pacote de consulta (*piggback*). O receptor envia de volta um ACK, também com dados se for o caso, depois de SIFS segundos. Após encerrar a transmissão a todas as estações contidas em uma lista de consultas, o coordenador de ponto reinicia o processo de consulta após PIFS segundos. Os usuários que estão sem transmitir por alguns ciclos são retirados da lista de consultas e são consultados de novo no inicio do próximo período livre de contenção. A Figura 2.15 [1] ilustra este procedimento.

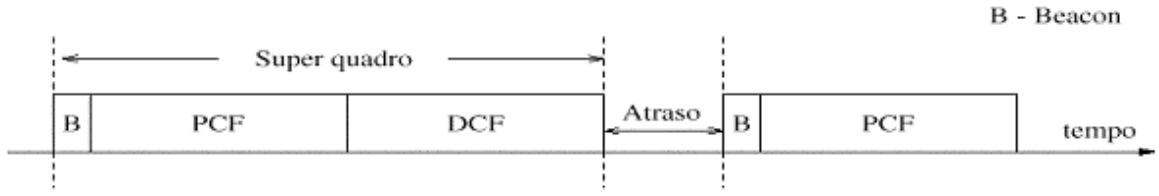


Figura 2.15 – Coexistência nos modos PCF e DCF

2.5 – O Esquema de Backoff do MAC IEEE 802.11

O subnível MAC do IEEE 802.11 implementa uma política de controle de acesso ao meio denominada Esquema de *Backoff*. Isto se traduz em um procedimento implementado nesse sub-nível que será invocado sempre que um STA transmitir um quadro e encontrar o meio ocupado indicado pelo meio físico ou por um mecanismo virtual de detecção de portadora.

Os mecanismos de detecção de portadora (*carrier-sense mechanism*) são utilizados para determinar o estado do meio podendo ser físico, neste caso provido pelo PHY, ou virtual (MAC), neste caso sendo provido pelo subnível MAC. Quando for provido pelo MAC, este mecanismo é referenciado pelo NAV. O NAV mantém uma previsão do futuro do tráfego no meio baseado na informação de duração obtida nos quadros RTS/CTS anteriores aos dados atuais permutados. O mecanismo de detecção de portadora combina o estado do NAV e o *status* dos STA´s transmitidos com a detecção de portadora física para determinar o estado do meio, se livre ou ocupado. O NAV pode ser imaginado como um contador, que conta de zero a uma taxa uniforme. Quando o contador é zero, é indicado pela detecção de portadora virtual que o meio está ocupado. Quando não for zero, a indicação é que o meio está livre. Sempre que um STA tiver transmitindo, o meio estará ocupado.

Sempre que um STA deseja transmitir quadros de dados ou de controle, ele antes chama o mecanismo de detecção de portadora para determinar o estado do meio. Se o meio estiver ocupado, o STA adia a transmissão até que o meio fique livre sem interrupção por um período de tempo igual a DIFS ou EIFS. DIFS, quando o último quadro detectado no meio foi recebido corretamente. EIFS quando o último quadro detectado no meio não foi recebido corretamente. Após um tempo DIFS ou EIFS com o meio livre, o STA então gera um período de *backoff* aleatório por um adiamento incremental de tempo antes de transmitir, a menos que *backoff timer* atual contenha um valor diferente de zero, neste caso a escolha do número aleatório não é necessária e não executada. O resultado esperado desse processo

é a minimização das colisões durante contenções (disputas) entre múltiplas STA que tenham sido adiadas no mesmo evento. A Equação 2.1, indica como é gerado esse tempo aleatório.

$$\text{Backoff Time} = \text{Random}() \times \text{aSlotTime}$$

Equação 2.1

Onde:

***Random*()** = é um número inteiro aleatório que varia no intervalo [0,CW].

CW (*Congestion Window)* é um número inteiro que varia de acordo com as características PHY *aCWmin* e *aCWmax*.

ASlotTime **=** Fatia de tempo necessária para transmitir.

Sempre que uma transmissão sem sucesso ocorrer o valor do parâmetro CW sofre um incremento exponencial. O incremento é feito em potencia de 2, e subtraindo 1 do resultado da exponenciação. Assim sendo, inicia-se com o valor *aCWmin* e continua incrementando até alcançar o valor *aCWmax.* A Figura 2.16 ilustra um exemplo de como isso funciona.

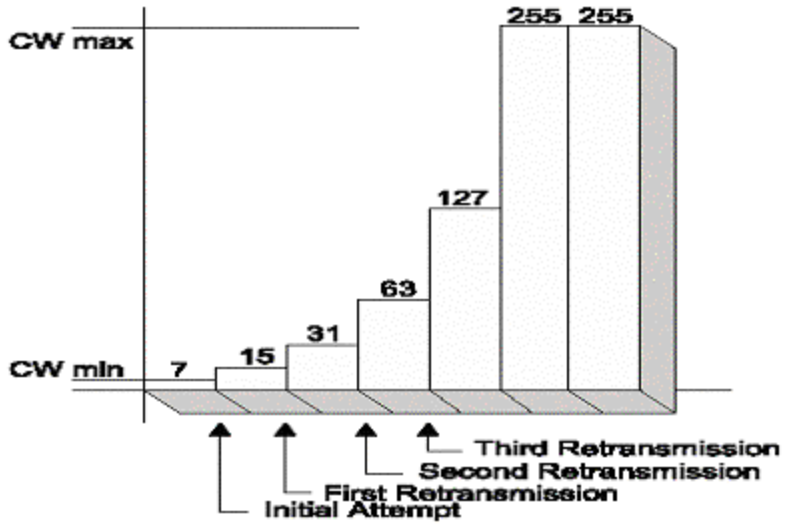


Figura 2.16 – Exemplo de Incremento de CW

Um STA executando o procedimento de *backoff* usa o mecanismo de detecção de portadora para determinar a atividade do meio durante cada *slot* do *backoff*. O procedimento de *backoff* irá decrementar o tempo de *backoff* por um *aSloTime* de tempo se nenhuma atividade for indicada pela duração de um *slot* de *backoff.* Se o meio estiver ocupado em qualquer tempo durante o *slot* de *backoff* então o procedimento de *backoff* é suspenso, ou seja o contador de *backoff* não decrementa para aquele *slot*. O meio deverá estar livre por um intervalo DIFS ou EIFS antes que seja permitido ao procedimento de *backoff* continuar. As transmissões começam sempre que o contador de *backoff* atingir zero.

O procedimento de *backoff* pode iniciar após diversas situações. No caso de transmissões realizadas com sucesso, o procedimento de *backoff* inicia após o recebimento do quadro ACK. Caso a transmissão não tenha ocorrido com sucesso, ele inicia no final do intervalo de *timeout* do quadro ACK. Se a transmissão ocorrer com sucesso, o valor do parâmetro CW volta a valer *aCWmin.* Isto assegura que os quadros transmitidos de um STA são sempre separados pelo último intervalo de *backoff* .

O efeito deste procedimento é que quando múltiplos STAs estão adiando transmissões e chegando a um *backoff* aleatório, então o STA seleciona o menor tempo de *backoff* da função aleatória, Equação 2.1, da janela de contenção. A Figura 2.17 [1] ilustra o funcionamento desta política.

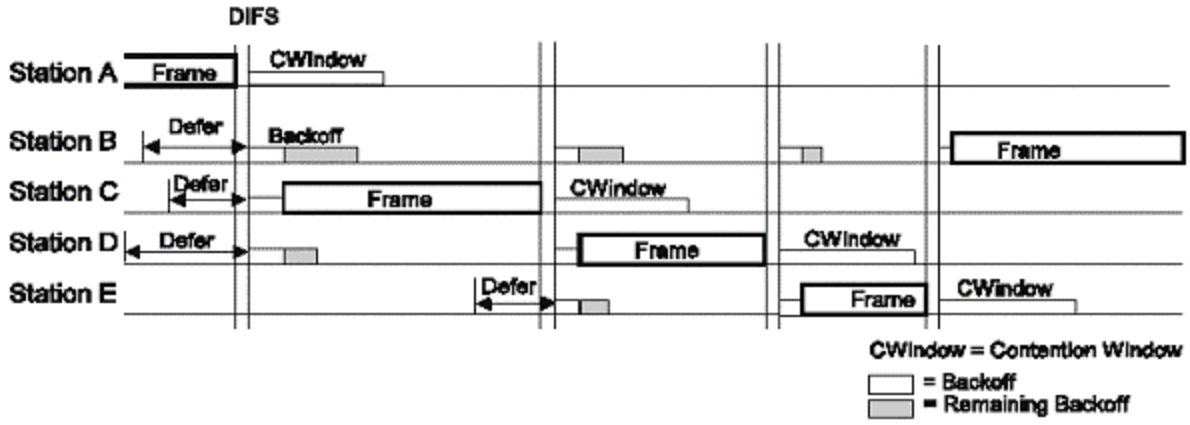


Figura 2.17 – Procedimento de Backoff

2.6 - O Nível Físico (PHY)

O nível físico do padrão IEEE 802.11 pode sem implementado através de radio-freqüência ou infra-vermelho, sendo elas [10][31] :

- **FHSS** (*Frequency Hopping Spread Spectrum*)
- **DSSS** (*Direct Sequence Spread Spectrum*)
- **IR** (*Infrared*).

As especificações FHSS e DSSS operam na freqüência de 2.4 Ghz da ISM (*Industrial Scientific and Medical*) [32]. O uso dessa freqüência é liberado não havendo nenhuma necessidade de licenciamento prévio de acordo com o FCC (*Federal Communications Comission*) [8]. A seguir iremos descrever cada uma delas [33].

2.6.1 - Frequency Hopping Spread Spectrum (FHSS)

Esta técnica de transmissão baseia-se em *spread spectrum* [55] [57]. Ela divide a banda do canal em subcanais, onde ocorrerá a transmissão em intervalos de tempos curtos, em outras palavras, o transmissor envia seus dados ciclicamente em vários subcanais de acordo com uma seqüência definida. O receptor por sua vez, deve percorrer os subcanais na mesma ordem em que o transmissor os utiliza. Cada canal deve ser utilizado por um breve espaço de tempo e, em média, todos os subcanais devem ser igualmente utilizados. Veja a Figura 2.18 [8] [11].

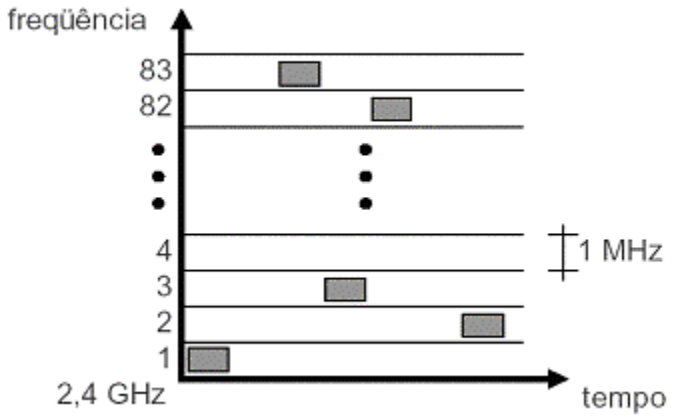


Figura 2.18 – Exemplo de Seqüência de Salto FHSS

O padrão IEEE 802.11 define a banda de 2,4 a 2,4835 GHz para o FHSS. Esta banda com largura de 83,5 MHz, foi dividida em 83 canais de 1MHz devendo ser utilizados pelo menos 75 destes subcanais. A cada intervalo de tempo de 30s um subcanal poderá ser ocupado por um intervalo durante 400 ms [9]. De acordo com o FCC o padrão utiliza 79 canais de 1MHz cada um.

A seqüência de saltos deve observar alguns critérios, sendo eles:

- Garantir a distância mínima de saltos para evitar propagação multipercurso;
- Minimizar saltos simultâneos de seqüências diferentes no mesmo canal ou em canais adjacentes;
- Minimizar saltos consecutivos em um mesmo canal de sistemas FHSS diferentes.

Os saltos deverão ter uma distância mínima de 6 (seis) canais. Prever-se também a utilização simultânea de até 26 (vinte e seis) sistemas FHSS 802.11 em uma mesma área, já que o padrão criou 3 (três) conjuntos de 26 (vinte e seis) seqüências de saltos, onde no pior caso, 5 (cinco) colisões de seqüências de um mesmo conjunto podem acontecer, incluindo saltos para freqüências adjacentes [11].

Na transmissão o padrão define a utilização de modulação GFSK (*Gaussian*

Frequency Shift Keying) [56], tecnologia simples e de baixo custo, para fornecer uma vazão mínima de 1 Mbps. A modulação GFSK utiliza um esquema do tipo multinível para possibilitar transmissões a taxas de 1 Mbps, 2 Mbps ou 11Mbps no caso do 802.11b [73]. A vazão de 1 Mbps é obrigatória e a de 2 Mbps opcional. Esta exigência permite a interoperabilidade de equipamentos de baixo custo, baixa vazão com os de alto custo e alta vazão.

O formato do quadro FHSS está ilustrado na Figura 2.19 [8] em seguida a definição de cada um dos seus campos.

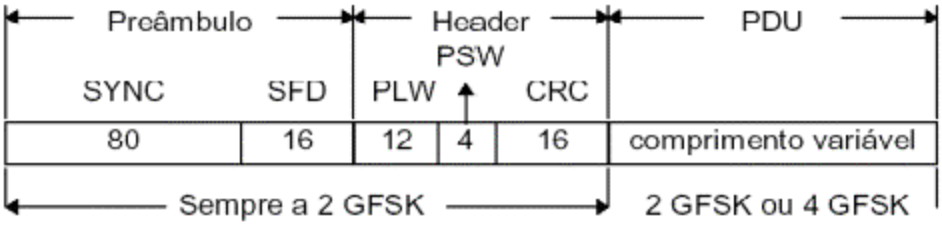


Figura 2.19 – O formato do Quadro FHSS

I) SYNC é uma seqüência de 80 bits no padrão 0101 e tem como objetivo adquirir o sincronismo, detectar a presença de sinal e resolver a diversidade da antena;

II) SFD (Start Frame Delimiter) define 16 bits, sendo ele: 0000, 1100, 1011, 1101, que oferecem sincronização de símbolos. Este padrão foi projetado para otimizar as propriedades de autocorrelação em conjunto com o padrão 0101 em frente dele;

III) PLW (Plcp_PDU Lenght Word) é um campo de 12 bits que indica o tamanho do PDU (P*hysical Data Unit*) em octetos, incluindo os 32 bits de CRC ao final do PDU;

IV) PSF (PLCP Signaling Field) é um campo de quarto bits, com três reservados e um para indicar a vazão do PDU (1 ou 2 Mbps);

V) CRC do cabeçalho gerado pelo polinômio ITU –T $P(x) = x^{16} + x^{12} + x^5 + 1$;

VI) PDU campo de dados das camadas superiores.

Preâmbulo e *header* são transmitidos a uma vazão de 1 Mbps, dados (PDU) a 1 ou 2 Mbps. Antes da transmissão os dados são embaralhados seguindo o polinômio de *feedback* $G(x) = x^7 + x^4 + 1$, e convertidos para símbolos, acrescenta-se um para reduzir a componente DC do sinal transmitido [8].

2.6.2 - Direct Sequence Spread Spectrum (DSSS)

Assim como no padrão FHSS, esta técnica também transmite na banda ISM de 2,4 GHz, a taxas de 1 Mbps ou 2 Mbps. Quando se utiliza a técnica de 1 Mbps, emprega-se a modulação DBPSK (*Differential Binary Phase Shift Keying)* [57]. Já na transmissão com velocidade de 2 Mbps, a modulação DQPSK (*Differential Quadrature Phase Shift Keying*) [58].

Na técnica de transmissão *spread spectrum* [55][57], cada tempo é dividido em *n* subintervalos denominados *chips*. Para transmitir 1 bit, uma estação envia uma seqüência de *chips,* isto é, cada bit é representado segundo uma seqüência pseudo-randômica de símbolos binários. Para enviar o bit 0, utiliza-se o complemento desta seqüência. O espalhamento do espectro do sinal ocorre de fato, pois, na transmissão de 1 Mbps tem-se o envio de n M*chips*/s. Veja a Figura 2.20a [8].

O padrão DSSS 802.11 adota para todas as máquinas que utilizam DSSS, a seqüência de *Barker,* que consiste em 11 símbolos, definida como sendo: +1, -1, +1,+1,-1, +1,+1,+1,-1,-1,-1, sinalizando uma taxa de *chip* de 11 M*chip*/s quando se transmite a taxa de 1 Mbps. Figura 2.20b ilustra como isso ocorre [8].

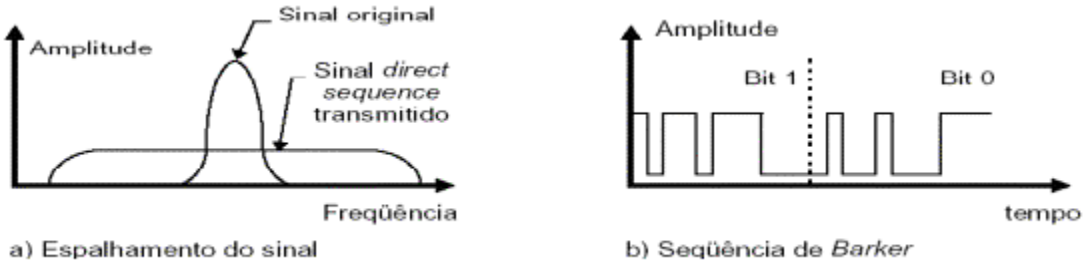


Figura 2.20 – Transmissão DSSS

O formato do quadro DSSS está ilustrado na Figura 2.21. Em seguida iremos descrever cada um dos seus campos [8].

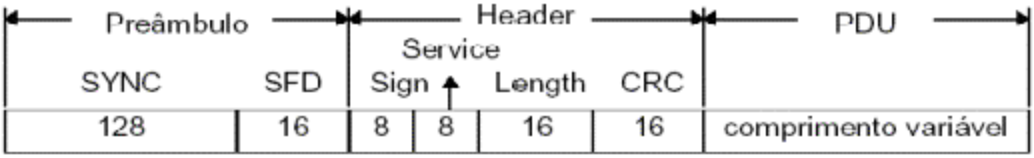


Figura 2.21 – O Formato do Quadro DSSS

I) SYNC são 128 bits embaralhados em 1, usado para o sincronismo do receptor, utilizando-se para isso a seqüência de *Barker* [59];

II) SFT (*Start Frame Delimiter*) oferece a sincronização de quadro e de octeto para o receptor, consistindo de 16 bits com o seguinte conteúdo: 1111 0011 1010 0000. Transmite-se a partir do bit menos significativo;

III) *Signal (* representado por *sign* na Figura 2.21**)** indica a vazão da transmissão dos dados do quadro. A velocidade é calculada pelo valor deste campo, multiplicada por 100 kbits. O padrão define dois valores obrigatórios para este campo, sendo 10 para 1 Mbps e 20 para 2 Mbps;

IV) *Service* é reservado para uso futuro, não sendo usado ainda;

V) *Length* é um número inteiro de 16 bits sem sinal, indica o número de microsegundos para a transmissão do PDU;

VI) CRC é gerado pelo polinômio ITU-T $x^{16} + x^{12} + x^5 + 1$.

Assim como na técnica FHSS todos os bits são embaralhados utilizando-se para isto o polinômio de *feedback* $G(x) = x^7 + x^4 + 1$.

O padrão define ainda 11 canais, dentro da banda de 2.4 GHz, para operação de redes locais DSSS, e mais uma faixa para operação no Japão. A Tabela 2.2 mostra os índices identificadores e as freqüências centrais destes canais. Nos Estados Unidos, estão disponíveis os canais de 1 a 11. Na Europa, os canais disponíveis são de 3 a 11. Redes vizinhas podem operar simultaneamente se escolherem canais diferentes com uma distância mínima de 30 MHz entre suas freqüências centrais.

Identificação do canal	Freqüências do FCC	Freqüências do ETSI	Freqüências do Japão
1	2412 MHz	N/D	N/D
2	2417 MHz	N/D	N/D
3	2422 MHz	2422 MHz	N/D
4	2427 MHz	2427 MHz	N/D
5	2432 MHz	2432 MHz	N/D
6	2437 MHz	2437 MHz	N/D
7	2442 MHz	2442 MHz	N/D
8	2447 MHz	2447 MHz	N/D
9	2452 MHz	2452 MHz	N/D
10	2457 MHz	2457 MHz	N/D
11	2462 MHz	2462 MHz	N/D
12	N/D	N/D	2484 MHz

Legenda:
N/D – não disponível
ETSI – *European Telecommunications Standards Institute*

Tabela 2.2 – Relação das Freqüências centrais de Canais para DSSS

2.6.3 - Infrared (IR)

A transmissão através da utilização de raios infravermelhos, é outra alternativa para as redes sem fio padrão 802.11. O comprimento da onda de raios infravermelhos varia de 0,75 a 1000 microns, que é maior do que as cores espectrais mas muito menor do que ondas de rádio. O padrão define a utilização infravermelha com comprimento de onda entre 750 e 850 nanômetros [13].

Neste tipo de rede, um transmissor e um ou mais receptores comunicam-se através de um plano de reflexão, que normalmente é o teto, paredes etc. Não deve haver qualquer tipo de obstáculo, que seja opaco a raios infravermelhos, em relação a qualquer nó móvel. O plano de reflexão deve ser monitorado por todos. Entretanto, não é necessário que nós móveis estejam alinhados entre si para se comunicarem. Todos devem comunicar-se através do plano de reflexão. A maior distância entre nós móveis e o plano de reflexão deve ser de, no máximo, 10 m. Veja a Figura 2.22 [13].

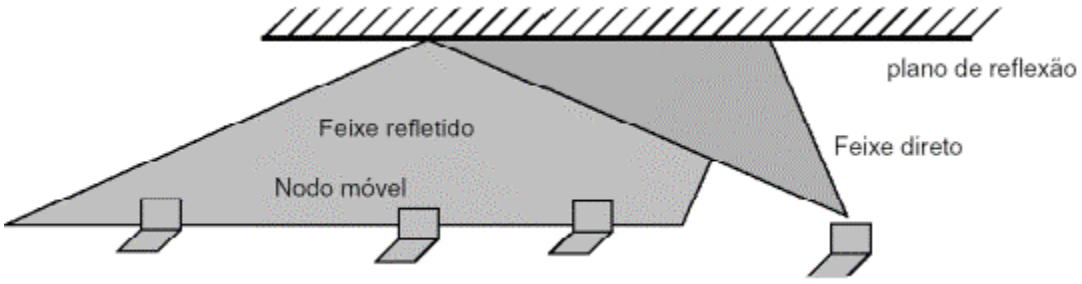


Figura 2.22 – Exemplo de Transmissão Infravermelha Difusa

Assim como nas outras duas técnicas de transmissão, é possível transmitir a 1 ou 2 Mbps. A modulação utilizada neste caso é de 16 PPM (*Pulse Position Modulation*) para 1 Mbps e 4 PPM para 2 Mpbs. O quadro deste tipo de transmissão também apresenta preâmbulo e *header,* que são transmitidos sempre a 1 Mbps, e dados, que podem ser transmitidos também a 2 Mbps [13].

2.7 - Protocolos do Subnível MAC

O nível de enlace está dividido em dois subníveis: o MAC e o LLC. O primeiro relaciona-se com o nível físico e o segundo com o nível de rede [7][8]. No nível físico não há nenhum tratamento ou organização da informação, ou seja, ou que existe são apenas bits brutos. O nível de enlace é responsável por organizar esses dados brutos em quadros, oferecendo assim, um primeiro mecanismo de tratamento da informação**.** O meio de transmissão em geral sempre foi caro e escasso. Em redes sem fio, o acesso ao mesmo é escasso e compartilhado entre todas as estações da rede. Controlar o aceso ao meio torna-se uma tarefa eventualmente complicada. Muitos esforços têm sido feitos nessa área e muitos protocolos foram propostos. Entretanto poucos são propostos para redes sem fio *ad hoc multihop.* Os protocolos de acesso ao meio sem fio para um simples canal podem ser tipicamente categorizados como sendo de múltiplo acesso. Em seguida iremos descrever o protocolo MACA (*Multiple Access Collision Avoidance*) e o MACAW (*Multiple Access Collision Avoidance For Wireless*) que surgiu como derivação do primeiro.

2.7.1 - Multiple Access Collision Avoidance (MACA)

O protocolo MACA foi proposto em [9] para uso em pacotes de radio como uma alternativa para o esquema tradicional do CSMA. O MACA é de certa forma similar ao protocolo proposto em [19].

O MACA utiliza pacotes RTS e CTS descritos na seção 2.3.2. De uma maneira geral funciona da seguinte forma: sempre que uma estação A deseja transmitir para uma estação **B**, ela envia um quadro RTS; este pacote RTS contém o tamanho dos dados a serem transmitidos. Se a estação **B** ouvir o RTS e não está rejeitando dados nesse instante, ela imediatamente responde para **A** com um quadro CTS. Este CTS contém o tamanho dos dados a serem transmitidos solicitados por **B**. Ao

receber o CTS, que é uma autorização para transmissão, a estação **A** inicia a transmissão dos dados. Qualquer estação que por acaso ouvir um quadro RTS adiará todas as transmissões até algum tempo depois do quadro CTS associado tenha sido finalizado. Qualquer estação que receber um quadro CTS, adiará suas transmissões pelo tamanho do quadro de dados autorizado anteriormente.

Com este algoritmo, qualquer estação que ouve um RTS aguardará um tempo suficiente de modo que a transmissão da estação possa receber o CTS. Qualquer estação que ouve o CTS evitará colisão com o retorno da transmissão do dado. Desde que o CTS é enviado para o receptor, cada estação passível de colisão com a transmissão de dados está na faixa do CTS. Veja que as estações que ouvem um RTS, mas não um CTS, podem começar a transmissão, depois que o CTS tenha sido enviado. Isto porque elas não estando dentro da faixa do receptor, elas não podem colidir com a transmissão dos dados [17].

Se uma estação não ouve um CTS em resposta de outra estação, ela irá eventualmente entrar em *timeout* e assumir que ocorreu uma colisão escalonando o pacote para retransmissão. O protocolo MACA usa o algoritmo de *backoff* exponencial binário (BEB) para escolher o tempo para retransmissão [17].

2.7.2 - Multiple Access Collision Avoidance For Wireless (MACAW)

MACAW é o protocolo obtido a partir do aperfeiçoamento do protocolo MACA, sendo muito semelhante a este e resolvendo alguns problemas que o MACA possui. Após o estudo do protocolo MACA apercebeu-se que, pacotes perdidos só seriam reconhecidos muito depois da ocorrência, somente quando a camada de transporte percebesse sua ausência. Para resolver este problema, introduziu-se um ACK depois de cada pacote recebido corretamente. Um outro problema era que duas estações podiam enviar ao mesmo tempo RTS para o mesmo destinatário. Para isso, utilizou-se o CSMA permitindo que várias estações detectassem quando o link está ou não livre. Utilizou-se igualmente o algoritmo de *backoff*

separadamente para cada fluxo de dados origem e destino, e não para cada estação, tornando-o assim justo no acesso ao meio. Finalmente adicionou-se um mecanismo para as estações, possibilitando a troca de informações acerca do congestionamento e também fazendo com que o algoritmo de *backoff* seja menos reagente a variações temporárias na carga da rede, melhorando desta forma o desempenho do sistema [7].

O tamanho dos pacotes RTS e CTS foi definido em 30 bytes. O tempo de transmissão desses pacotes define o *slot* de tempo para retransmissão. As retransmissões ocorrem se e somente se uma estação não receber um CTS em resposta a um RTS. As retransmissões são escalonadas em número inteiro de *slot* de tempo depois do fim do último período de adiamento. Uma estação escolhe randomicamente, com distribuição uniforme, este número inteiro entre 1 e BO, onde BO representa o contador de *backoff*. O algoritmo de *backoff* ajusta o valor de *backoff* através de duas funções F_{dec} (função decremento de *backoff*) e F_{inc}(função incremento de *backoff*). Sempre que um CTS for recebido depois de RTS, o contador de *backoff* é ajustado via função F_{dec} : BO := F_{dec} (BO). Sempre que um CTS não for recebido depois de RTS o contador de *backoff* é ajustado via função F_{inc} : BO := F_{inc} (BO). Para o BEB, $F_{dec}(x) = BO_{min}$ e F_{inc} (x) = $MIN\ [2x, BO_{Max}]$. Onde BO_{min} e BO_{max} representam o menor e maior limite para o contador de *backoff* respectivamente [17].

RESUMO DO CAPÍTULO

Neste capítulo nós apresentamos o padrão 802.11, com ênfase aos componentes de uma rede com infra-estrutura. Mostramos ainda a arquitetura, serviços, e os protocolos do nível MAC bem como o esquema de *backoff* exponencial. Foi visto ainda o nível físico e suas implementações. No próximo capítulo, o foco serão as redes *ad hoc multihop*, onde mostraremos através de simulações, dois problemas do protocolo DFWMAC quando usado em redes *ad hoc multihop.*

Capítulo 3

O Problema do Protocolo MAC IEEE 802.11

3.1 - Apresentação do Problema

Os protocolos de nível MAC para redes 802.11 ainda possuem problemas e estão sendo aprimorados para apresentar maior confiabilidade. O protocolo DFWMAC foi definido como o padrão para redes sem fio e tem sido utilizado freqüentemente, em ambientes de teste, simulações e pesquisas em redes *ad hoc*. Entretanto uma questão importante deve ser levantada: este protocolo trabalha bem em redes *ad hoc multihop* [5] ?

Esta questão fica muito mais evidente quando avaliamos conexões TCP em uma rede local sem fio e verificamos a sua performance. Segundo [6][7] o TCP é um protocolo que trabalha no nível de transporte, portanto acima dos níveis de rede e enlace. Desta forma, os protocolos de nível MAC de redes ad hoc devem suportar o TCP. Se o DFWMAC não puder suportar o TCP, isto já é um indício para que este protocolo não seja usado neste tipo de rede, mesmo que para alguns tipos de redes locais sem fio, ofereça maior confiabilidade.

A apresentação desse problema que ocorre em redes locais sem fio *ad hoc multihop*, será feita tomando-se por base o fato de que tráfego TCP intensifica o problema do protocolo de nível MAC 802.11 em redes locais sem fio *ad hoc multihop*. Apresentaremos duas situações onde poderemos perceber dois problemas existentes nesse tipo de rede. O primeiro problema é de **instabilidade** do TCP, verificado através do seu *throughput* atraso e *jitter*. O segundo é de **injustiça** no acesso ao meio. Veremos que o primeiro problema pode ser contornado de maneira relativamente fácil, enquanto que o segundo não, tornando-se um grande desafio para pesquisas em busca de uma solução definitiva e funcional.

3.2 - Ambiente e Metodologia de Simulação

Afim de visualizar melhor os problemas aqui apresentados, criamos um cenário para simulação, onde tais problemas aparecem. Os resultados obtidos e apresentados nesse capítulo estão baseados nas simulações feitas no NS (*Network Simulation*) de *Lawrence Berkley National Laboratory* (LBDL), versão 2.1b8a, com extensões de MONARCH *Project at Carnegie Mellon* [37] e nos experimentos realizados em [5]. A topologia e condições gerais usadas nas simulações usadas nesse capítulo, seguem os seguintes parâmetros:

- **Quantidade de Estações**..: 8 (numeradas de 0 a 7)
- **Distância entre elas**.....: 200 m (o padrão admite o máximo de 250m)
- **Protocolo de Roteamento:** DSR (*Dynamic Source Routing*)
- **TCP Utilizado**.....: Reno
- **Tamanho do pacote TCP...:** 1460 bytes
- **Velocidade dos Links.....:** 2 Mbps
- **Mobilidade das Estações**: Não. As estações estão estáticas e alinhadas

Em todas as simulações apresentadas nesse capítulo, foram estabelecidas conexões TCP entre dois pontos e interligando transmissores e receptores FTP. A Figura 3.1 mostra a disposição visual do cenário *multihop* escolhido para a simulação [5]. Antes de passarmos a apresentação dos problemas, veremos algumas questões relativas a protocolos de roteamento, problemas de terminal escondido e exposto e o controle de congestionamento do TCP. Estes temas estão relacionados com aos problemas a serem abordados neste capítulo.

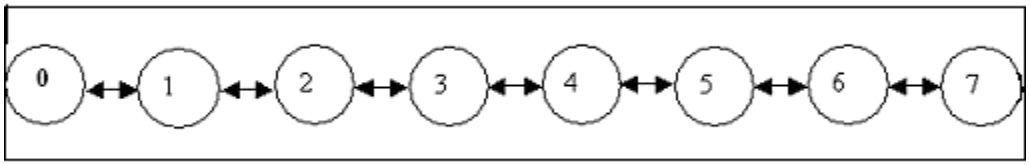


Figura 3.1 – Cenário *Multihop* das Simulações

3.3 - Protocolos de Roteamento Utilizados

Um algoritmo de roteamento precisa ter algumas características ideais. As principais são: escolha da melhor rota, simplicidade, robustez, imparcialidade, estabilidade, rapidez convergência para o caminho ótimo, flexibilidade, aceitar parâmetros de qualidade de serviço (QoS) e independência da tecnologia da rede. A robustez é a principal de todas estas características. É esperado que uma rede fique funcionando sem interrupções ou falhas por anos. O algoritmo de roteamento deve ser robusto o suficiente para suportar isto.

Algoritmos podem ser agrupados em várias classes: adaptativos ou não-adaptativos, distribuídos ou centralizados, pró-ativos ou reativos, um caminho ou vários caminhos, planos ou hierárquicos, *host* inteligente ou roteador inteligente, inter-domínio ou intra-domínio, estado do link ou vetor de distância. Muitas destas características não excluem outras, dizem respeito apenas a características que estamos observando do protocolo.

Em uma rede local sem fio *ad hoc multihop,* cada estação funciona como um roteador, encaminhando pacotes de dados para outras estações. O desafio principal de redes *ad hoc* é o desenvolvimento de protocolos de roteamento que possam encontrar eficientemente rotas entre comunicação de duas estações. Três critérios devem ser sempre considerados quando são projetados protocolos de roteamento para redes *ad-hoc* [14]:

- Inexistência de uma entidade controladora central;
- Possibilidade de rápidas mudanças na topologia da rede ;
- Todas as comunicações ocorrerão através de ondas de rádio.

A falta de um ponto central, coordenando toda a rede, requer algoritmos distribuídos mais sofisticados, para enfrentar o problema de roteamento. A mudança na topologia, pode deixar as informações de localização rapidamente desatualizadas. Um bom algoritmo, deve observar bem estas mudanças.

Outro ponto importante, é a questão envolvendo os gasto com energia. Este ponto é um limitante, e como tal, deve ser observado e considerado, em cada fase do projeto de algoritmos para redes *ad hoc*. Economia de energia aliada a baixo tempo de convergência e robustez, certamente são as mais importantes características algoritmos de roteamento em redes *ad hoc*.

A avaliação quantitativa do desempenho de um protocolo de roteamento pode ser feita através da análise dos seguintes pontos:

- *Throughput* de pacotes de dados fim-a-fim;
- Atraso de pacotes;
- Tempo de descobrimento da rota;
- Percentual de pacotes não entregues na ordem correta;
- Eficiência do protocolo.

Um protocolo deve também atentar para o aspecto da escalabilidade, levando em consideração o tamanho da rede, capacidade dos *links,* mobilidade. Outra característica desejável, é que o protocolo seja capaz de manipular parâmetros de QoS (*Quality of Service*) [67][68].

Nas simulações apresentadas neste trabalho utilizaremos três protocolos de roteamento, que fazem parte das duas classes abaixo:

- **Table-driven:** Nesta classe os protocolos mantêm as informações de roteamento em todos os nós consistentes, de acordo com as tabelas de roteamento. Nesta classificação, DSDV (*Destination-Sequenced Distance-Vector Routing*), WRP (*Wireless Routing Protocol*) e CGSR (*Clusterhead Gateway Switch Routing*) [64].

- **On-Demand:** Os protocolos que fazem parte desta classe, criam rotas de acordo com as solicitações do nó origem. Nesta classe temos o AODV (*Ad Hoc On-Demand Distance Vector Routing*), DSR (*Dynamic Source Routing*), LMR (*Lightweight Mobile Routing*), TORA (*Temporally*

Ordered Routing Algorithm), ABR (*Associativity-Based Routing*) e SSR (*Signal Stability Routing*) [64].

Os protocolos que utilizaremos nas simulações serão o DSR, AODV e DSDV. Procuramos escolher protocolos de duas classes distintas e dentro de uma classe comparar os dois protocolos. A escolha específica de um determinado protocolo não seguiu nenhuma das suas característica especiais ou técnicas, mesmo porque as diferenças percebidas entre eles não foi muito considerável, como será visto no próximo capítulo. Já a escolha do TCP Reno, deu-se em função da popularidade desta versão protocolo TCP e de sua aceitabilidade.

Em seguida veremos o protocolo de roteamento DSR. Os protocolos AODV e DSDV serão vistos no capítulo seguinte juntamente com a simulação envolvendo os mesmos.

3.3.1 – Dynamic Source Routing (DSR)

O DSR [30] é um protocolo de roteamento sob demanda, baseado no conceito de roteamento pela origem. Os nós mantêm um *cache* com todas as rotas conhecidas. Este *cache* é permanentemente atualizado à medida que novas rotas são encontradas. Quando um nó tem um pacote para enviar, ele deve antes checar seu *cache* para verificar a existência de uma rota para o destino. Se ele tiver uma rota em seu *cache,* ele usará esta rota para enviar o pacote. Se ele não conhecer uma rota, ele envia em *broadcast* um pacote de pedido de rota (*route request)*. Este pacote contém o endereço do destino, o endereço da fonte e um número de identificação único. Cada nó ao receber este pacote, verifica a existência de uma rota para o destino em questão. Caso não conheça, ele acrescenta seu próprio endereço no registro de rota do pacote e o envia por seus enlaces de saída. Para controlar o número de mensagens na rede, um nó só encaminha um *route request,* se ele já não tiver passado por ele se o seu endereço não constar no registro de rota do pacote. Caso o pacote chegue no destino ou em um nó intermediário que conheça uma rota para o destino, é gerado um *route reply*, no qual é incluído o

registro de rota completo até o destino. Se o nó que gera o *route reply* conhece uma rota para o nó origem, ele envia o *route reply* diretamente para ele, se não, inicia um *route request* e faz um *piggyback* (em carona) do *route reply*. Quando é verificado algum erro fatal de transmissão, é gerado um pacote *route error* que faz com que o nó em questão seja retirado da *cache* de rotas. Os pacotes de reconhecimento também são usados para verificar a correta operação dos enlaces. A Figura 3.2 ilustra a criação de uma rota DSR entre o nó origem N1 e o destino N8 [15] [16] [35].

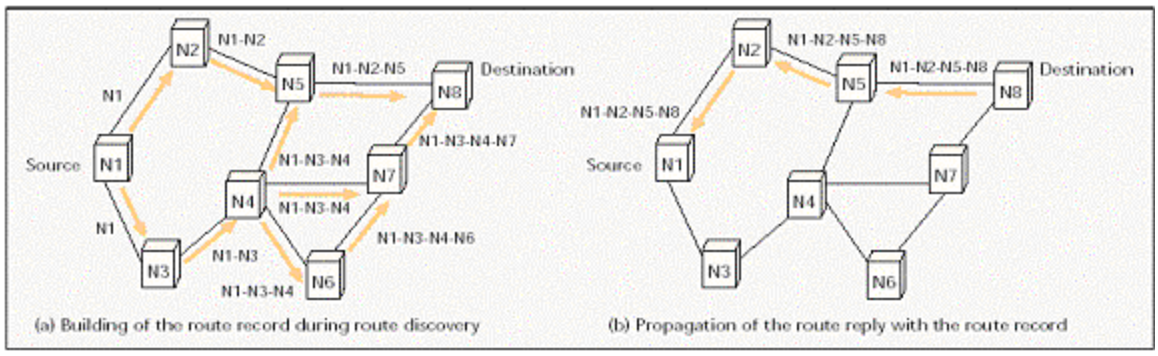


Figura 3.2 – Exemplo de criação de uma rota usando DSR

3.4 - Dois Problemas Típicos em Redes 802.11

Vamos conhecer agora dois problemas típicos em redes 802.11 *ad hoc*: o problema do terminal escondido e exposto, que eventualmente aparecem, em virtude da disposição das estações em um cenário *multihop.*

3.4.1 - O Problema da Estação / Terminal Escondido

Um problema muito comum em redes sem fio *ad hoc* é chamado problema do terminal escondido [60][61]. Ele ocorre na seguinte situação: Considere três estações **A, B** e **C** distribuídas em uma topologia alinhada. A estação **B** consegue detectar transmissões das estações **A** e **C**, contudo **A** e **C** não conseguem detectar-se, pois **A** está fora do alcance de **C,** a mesma coisa acontece com a

estação **C**. Por este motivo, quando **A** envia pacotes para **B**, **C** não consegue detectar esta transmissão e conclui que o meio está livre enviando dados para **B**. Se isso acontecer no mesmo intervalo de tempo, ocorrerá uma colisão em **B** e consequentemente, a informação de **A** ou de **C** chega corrompida em **B**. As estações **A** e **C** são chamados estações ou terminais escondidos. Veja a ilustração a Figura 3.3.

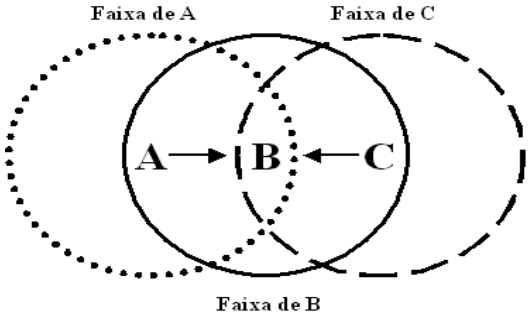


Figura 3.3 – Exemplo de Terminal Escondido

3.4.2 - O Problema do Terminal Exposto

Outro problema bastante comum é o problema do *Terminal Exposto* [60][61]. Este problema ocorre na seguinte situação: Suponha agora a mesma quantidade e topologia descrita acima, adicionando mais uma estação. Uma estação **B** manda pacotes para a estação **A**. A estação **C** detecta esta transmissão. Seria incorreto se **C** concluísse que não poderia transmitir para nenhuma outra estação pelo fato da transmissão de **B** para **A** está ativa. Suponhamos, por exemplo, que **C** queira enviar pacotes para **D**. Isto não seria nenhum problema, visto que a transmissão de **C** para **D** não vai interferir com a transmissão de **B** para **A**. Só interferia se a transmissão fosse de **A** para **B**. **C** é chamado uma estação / terminal exposto. Veja a Figura 3.4 [16] a ilustração deste problema.

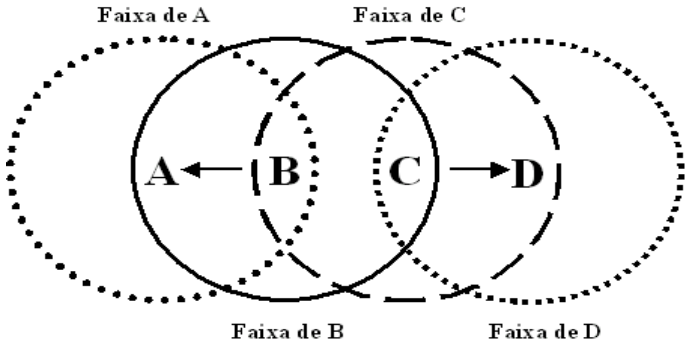


Figura 3.4 – Exemplo de Terminal Exposto

3.5 - Controle de Congestionamento no TCP

O protocolo TCP foi escolhido para as simulações em razão da sua confiabilidade e robustez. Como iremos detalhar nesta seção. A seguir será apresentada uma visão geral do TCP seguido do controle de congestionamento implementado nele.

3.5.1 - Visão Geral do TCP

O TCP foi projetado especificamente para oferecer um fluxo de bytes fim-a-fim confiável em uma subrede não confiável. Uma subrede é diferente de uma única rede porque suas muitas partes podem ter topologias, larguras de banda, retardos, tamanhos de pacotes e outros parâmetros completamente diferentes de uma rede propriamente dita. O TCP foi projetado para adaptar-se dinamicamente às propriedades da inter-rede e para ser robusto diante de muitas falhas que podem ocorrer. O TCP foi formalmente definido na RFC 793 [38], sofreu várias modificações e suas extensões foram definidas na RFC 1323 [39] .

Cada máquina compatível com o TCP tem uma entidade de transporte TCP, que pode ser um processo de usuário ou parte do *kernel* que gerencia fluxos e interfaces TCP para a camada IP(Protocol Internet) [66]. Uma entidade TCP aceita fluxos de dados do usuário provenientes de processos locais, divide-os em segmentos de no máximo 64 Kb e envia cada parte em um datagrama IP distinto.

Quando os datagramas IP que contém dados TCP chegam a uma máquina, eles são enviados à entidade TCP, que restaura os dados originais. Ou seja, às vezes utilizaremos somente o termo TCP para fazer referência tanto a entidade de transporte (um software) quanto ao protocolo TCP (um conjunto de regras). Pelo contexto, fica claro a que estaremos nos referindo. A camada IP não oferece qualquer garantia de que os datagramas serão entregues corretamente, ela implementa o serviço de melhor esforço [54]. Portanto cabe à camada de transporte, através do TCP, administrar os temporizadores e retransmití-los sempre que for necessário. Os datagramas também podem chegar fora de ordem. A camada de transporte terá de reorganizá-los em mensagens na sequência correta. Em suma, o TCP deve oferecer a confiabilidade que a maioria dos usuários deseja mais que o IP não oferece [7].

Veja algumas das vantagens do TCP:

- **Confiabilidade:** 90% do tráfego da Internet é gerado pelo protocolo TCP. Este tráfego está dividido em HTTP (*Hipertext Transfer Protocol*), FTP (*File Transfer Protocol*), Telnet e outros, oferecendo garantias na transmissão;

- **Fácil Adaptação às situações da rede:** o TCP se adapta às condições da rede, ou seja diante de uma situação de erros eventuais na transmissão, ele oferece mecanismos de recuperação de tais falhas, recuperando sempre o que for possível;

- **Controle de Congestionamento:** o TCP possui mecanismos para evitar e tratar o congestionamento. A seguir veremos alguns.

3.5.2 - O Controle de Congestionamento do TCP

O controle de congestionamento em uma rede está associado a dois aspectos: **capacidade da rede** e **capacidade do receptor**. Veja a Figura 3.5. No primeiro caso, a rede não consegue dar vazão aos pacotes em que é submetida, gerando congestionamento na rede como mostra a Figura 3.5a [7]. No segundo caso, não há problema na capacidade de transmissão da rede e sim a existência de um receptor lento que não consegue receber pacotes na mesma velocidade que a rede entrega como mostra a Figura 3.5b [7]. Existem duas formas de resolver o problema do congestionamento no TCP. A primeira seria antecipar-se ao problema evitando que ele não apareça e a segunda seria, uma vez que o congestionamento está evidente resta agora tratá-lo. Cada um desses problemas deverá ser tratado em separado. Desta forma cada transmissor mantém duas janelas: a janela fornecida pelo receptor e a janela de congestionamento. Cada uma dessas janelas deve mostrar o número de bytes que o transmissor poderá enviar, que é o valor mínimo das duas janelas. Portanto a janela efetiva será o mínimo do que transmissor e receptor consideram viável.

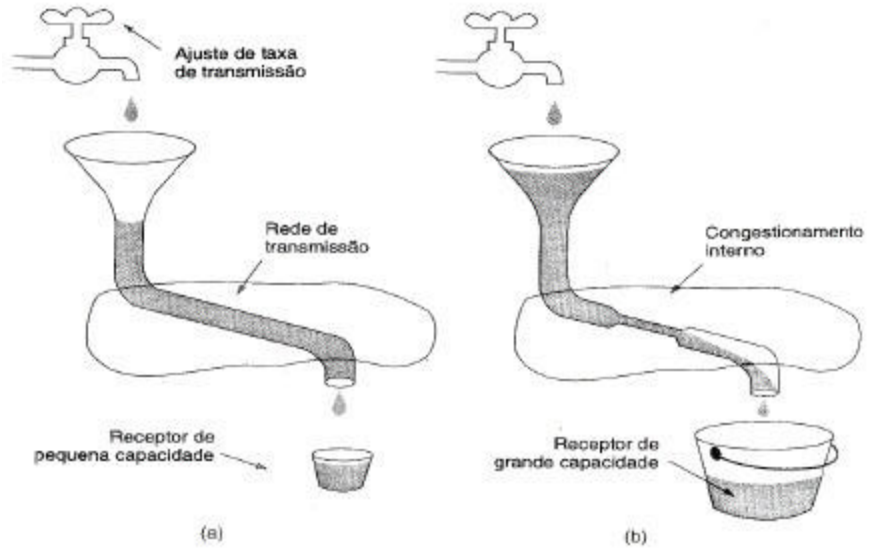


Figura 3.5 - Ilustração de Congestionamento

Existem vários algoritmos de controle de congestionamento, passaremos a conhecer alguns deles em seguida.

3.5.2.1 - Inicialização Lenta (Slow Start)

Apesar do nome, este algoritmo não é lento e sim exponencial. Quando uma conexão é estabelecida, o transmissor ajusta a janela de congestionamento ao tamanho do segmento máximo em uso na conexão. Em seguida, ele envia um segmento máximo. Se esse segmento for confirmado antes de ocorrer um *timeout*, o transmissor incluirá o número de *bytes* de um segmento na janela de congestionamento de modo que ela tenha capacidade equivalente a dois segmentos máximos e enviará dois segmentos. À medida que cada um desses segmentos for confirmado, a janela de congestionamento é aumentada em um tamanho de segmentos máximo. Quando a janela de congestionamento chegar a *n* segmentos em tamanho, e se todos os n segmentos forem confirmados a tempo, a janela de congestionamento será aumentada no número de *bytes* correspondentes aos *n* segmentos. Na prática, cada rajada confirmada duplica a janela de congestionamento. O crescimento é exponencial até que ocorra um *timeout* ou que a janela do receptor seja alcançada.

A idéia é que se rajadas de 1.024, 2.048 e 4.096 *bytes* forem recebidas, mas a de 8.192 causar *timeout*, então a janela de congestionamento deverá ser mantida em 4.096. Desde que a janela de congestionamento seja mantida em 4.096 bytes, nenhuma rajada superior a esta será enviada, não importando quanto espaço de janela o receptor ofereça. O TCP utilizado é o de *Van Jacobson* definido na RFC 1332 [40]. Todas as implementações devem ser compatíveis com ele.

3.5.2.2 - Algoritmo de Controle de Congestionamento da Internet

Este algoritmo utiliza além da janela do receptor e da janela de congestionamento, um terceiro parâmetro, o **limitante** (*threshold*). Em princípio o valor do limitante é 64KB. Quando há um *timeout*, o limitante é atribuído à metade da janela de congestionamento atual, e a janela de congestionamento é reinicializada para um tamanho de segmento máximo. Em seguida, o algoritmo de inicialização lenta é usado para determinar o que a rede é capaz de gerenciar, só que agora o

crescimento exponencial é interrompido quando o limite é alcançado. A partir daí, as transmissões bem sucedidas proporcionam um crescimento linear à janela de congestionamento. O aumento é de um segmento máximo para cada rajada em vez de um para cada segmento. Na prática, esse algoritmo diminui o tamanho da janela de congestionamento à metade, e depois retoma o seu crescimento. Veja a Figura 3.6 que ilustra este algoritmo [7].

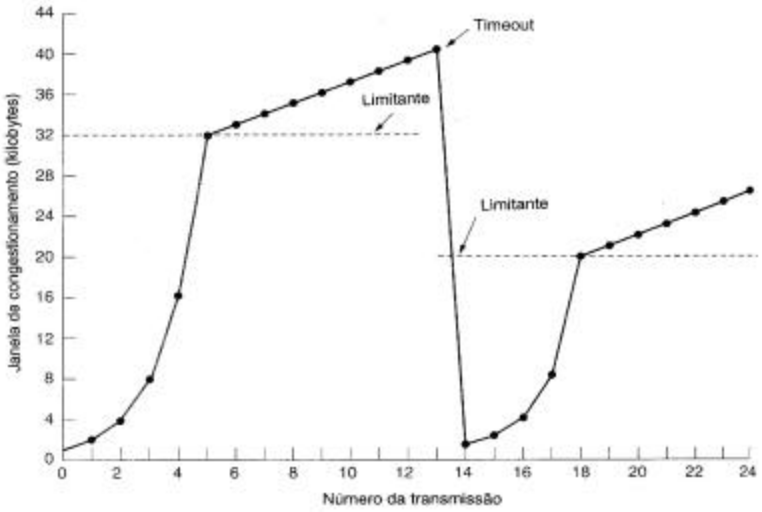


Figura 3.6 – Exemplo do Algoritmo de Congestionamento para a Internet

Veja que o tamanho máximo do segmento é 1.024 *bytes*. A janela de congestionamento inicialmente tem 64 KB, mas há um *timeout*; portanto foi atribuído ao limite o valor de 32 KB e à janela de congestionamento 1 KB para transmissão 0. Em seguida a janela de congestionamento cresce de maneira exponencial até chegar ao limite, 32 KB. A partir daí seu crescimento é linear. A transmissão 13 não tem muita sorte, havendo um *timeout*. Ao limite é atribuído um valor que é a metade da janela atual, que atualmente está com 40 KB, ficando o limite de 20 KB. O algoritmo de inicialização lenta começa outra vez. Quando as confirmações da transmissão 18 começarem a chegar, os quatro primeiros incrementam a janela de congestionamento em um segmento máximo cada um; depois disso, o crescimento é linear. Se houver outro *timeout*, a janela de congestionamento continuará a crescer até atingir o tamanho da janela do receptor. Nesse ponto ele pára de crescer permanecendo constante desde que não ocorra outro *timeout* e que a janela do receptor não mude de tamanho.

Na maioria dos protocolos de enlace o gerenciamento de janelas não é diretamente ligado à confirmações, como ocorre com o TCP. Suponha que o receptor tem um *buffer* de 4.096 *bytes*. Se o transmissor enviar um segmento de 2.048 *bytes* que é recebido de forma correta, o receptor confirmará o segmento. Porém, como agora ele só tem 2.048 *bytes* de espaço disponível em seu buffer, o receptor anunciará uma janela de 2.408 *bytes* no próximo *byte*. Veja a ilustração na Figura 3.7 [7].

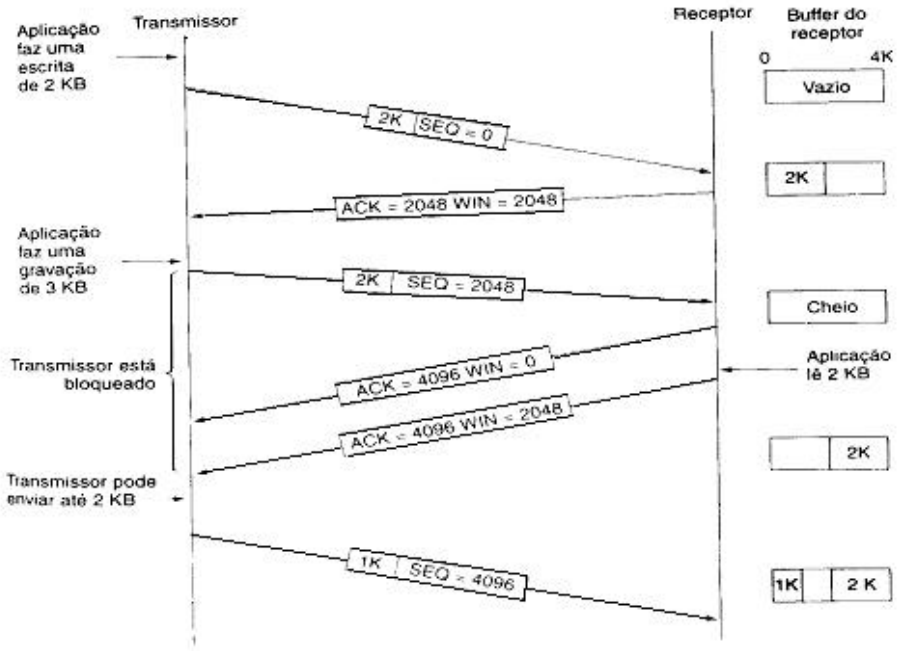


Figura 3.7 – Gerenciamento de Janelas no TCP

Agora o transmissor envia outros 2.048 *bytes*, que são confirmados, mas a janela anunciada é 0. O transmissor deverá parar até que o processo da aplicação no *host* receptor remova alguns dados do buffer, quando então o TCP poderá anunciar uma janela maior.

As implementações do TCP são diferentes para redes com fio e sem fio. Isso porque em função das características do meio torna-se necessário a utilização customizada para o meio físico em uso. Como consequência, podemos chegar a implementações que estão logicamente corretas, mas que possuem um desempenho péssimo. O principal problema dessas implementações é o algoritmo

de controle de congestionamento. Quase todas as implementações do TCP atuais assumem que os *timeouts* ocorrem devido a congestionamentos, e não devido a perca de pacotes. Conseqüentemente, quando um temporizador expira, o TCP diminui o ritmo e começa a transmitir mais lentamente.

Infelizmente os enlaces de dados das transmissões sem fio não são confiáveis. Eles perdem pacotes o tempo todo. A melhor estratégia para lidar com pacotes perdidos é enviá-los novamente o mais rápido possível. Diminuir o ritmo nesse caso tornará a situação ainda pior. Se por exemplo 20 por cento de todos os pacotes se perderem, quando o transmissor enviar 100 pacotes/segundo, o *throughput* será de 80 pacotes/segundo. Se o transmissor diminuir a carga para 50 pacotes/segundo o *throughput* cairá para 40 pacotes/segundo Em uma rede com fio, quando um pacote é perdido o transmissor não deve diminuir o ritmo. Da mesma forma, quando isso ocorre com redes sem fio, o transmissor não deve diminuir o ritmo de transmissão [7].

Com freqüência o caminho entre transmissor e receptor não é homogêneo. Parte do caminho pode ser controlado por uma rede com fio e outra por uma rede sem fio. Nessas circunstâncias, é mais difícil ainda tomar uma decisão em relação ao *timeout*, pois é necessário saber onde ocorreu o problema. Uma solução proposta é a utilização do **TCP Indireto** (TCP *Indirect*). Ele divide a conexão TCP em suas conexões separadas. A primeira conexão vai do transmissor à estação base. A segunda, vai da estação base ao receptor. A estação base simplesmente copia pacotes entre as conexões em ambas as direções. A vantagem desse mecanismo é que agora as duas conexões são homogêneas. Os *timeouts* da primeira conexão podem fazer com que o transmissor diminua o ritmo, enquanto que na segunda eles fazem com que o ritmo da transmissão aumente. Outros parâmetros também podem ser ajustados separadamente para as duas conexões. A desvantagem é que o mecanismo viola a semântica do TCP. Como cada parte da conexão é uma conexão TCP, completa, a estação base confirma cada segmento de maneira usual. A diferença é que quando o transmissor recebe uma confirmação, isso não quer dizer que o segmento chegou ao receptor, mas sim àestação base.

A outra grande desvantagem é que se o enlace de dados sem fio perder muitos pacotes, a máquina de origem poderá sofrer um *timeout* enquanto espera por uma confirmação e acabar chamando o algoritmo de controle de congestionamento. Com o TCP indireto, o algoritmo de controle de congestionamento jamais será inicializado, a menos que exista realmente um congestionamento na parte com fio da rede.

3.6 - O Problema da Instabilidade do TCP

Nessa primeira simulação foi utilizada apenas uma conexão TCP entre duas estações com quatro saltos, sendo 1 o nó emissor e 5 o nó receptor. Medimos então os pacotes que chegaram com sucesso a estação 5 durante todo o tempo de conexão. Nesta simulação não levamos em consideração nenhum outro tipo de tráfego da rede para que pudéssemos avaliar o *throughput,* atraso e *jitter* da conexão TCP entre esses dois pontos. A Figura 3.8 ilustra o cenário da simulação.

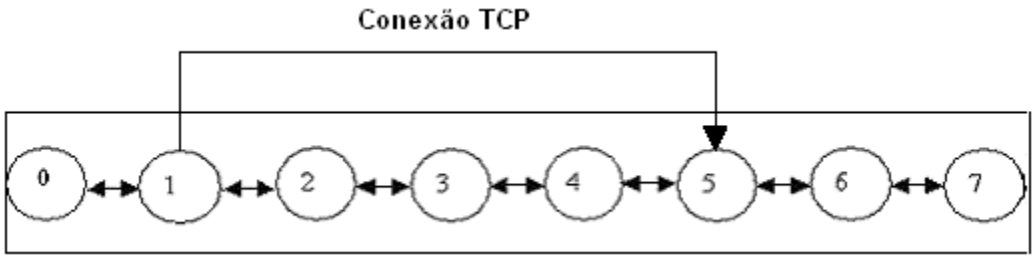


Figura 3.8 – Cenário *Multihop* da Simulação 1

O tempo total de simulação foi de 120s sendo que o *throughput*, atraso e *jitter* foram medidos a cada intervalo de 1.0s durante toda a simulação. Para esta simulação o parâmetro do TCP, tamanho máximo da janela do transmissor (*window_)*, foi atribuído o valor 32. Observando o Gráfico 3.1, percebemos que em 17 (dezesseis) instantes o *throughput* se aproxima ou alcança zero. Nesse intervalo de 1.0s quase nenhum pacote TCP foi recebido com sucesso. Isto mostra que a performance do TCP sofreu sérias degradações. Visto que somente uma conexão existe na simulação, este tipo de ocorrência não deveria acontecer. Esta oscilação do TCP pode ser explicada pelo fato de que esta versão do TCP não trabalha bem

com redes sem fio *ad hoc multihop*. Nós chamamos este problema de **Instabilidade do TCP.** Veja o Gráfico 3.1.

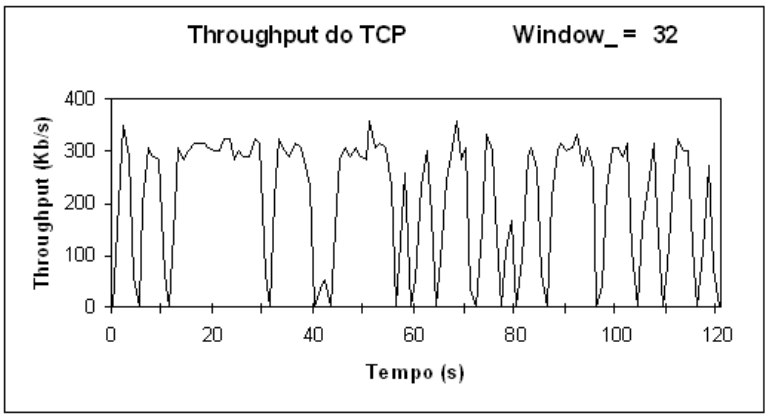


Gráfico 3.1 – *Throughput* da Simulação 1 com *window_* = 32

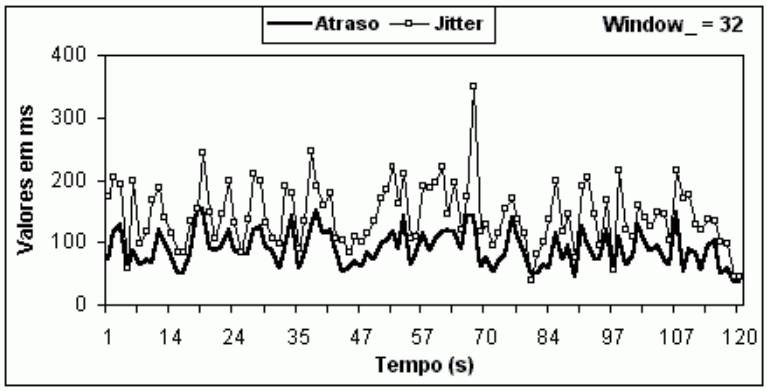


Gráfico 3.2 – Atraso e *Jitter* da Simulação 1 com *window_* = 32

A análise dos *traces* mostrou que o *throughput* agregado foi de 25.260, com média de 210,50 Kbps, e **variação** de 357,76 Kbps, sendo este o **maior** valor e o **menor** sendo zero. As médias de atraso e *jitter* foram respectivamente 75,74 ms e 121,13 ms. Veja o Gráfico 3.2.

Conforme já dissemos acima, o tamanho da janela de transmissão do TCP pode causar problemas. Assim sendo, o problema de instabilidade do TCP pode ser minimizado e até mesmo eliminado, com a diminuição do tamanho máximo da janela de transmissão do TCP. Para comprovar isto, promovemos mudanças no tamanho da janela a fim de verificar se o problema ainda aparecia. Atribuímos ao parâmetro *window_* o valor 8 e simulamos novamente. O resultado pode ser visto no Gráfico 3.3.

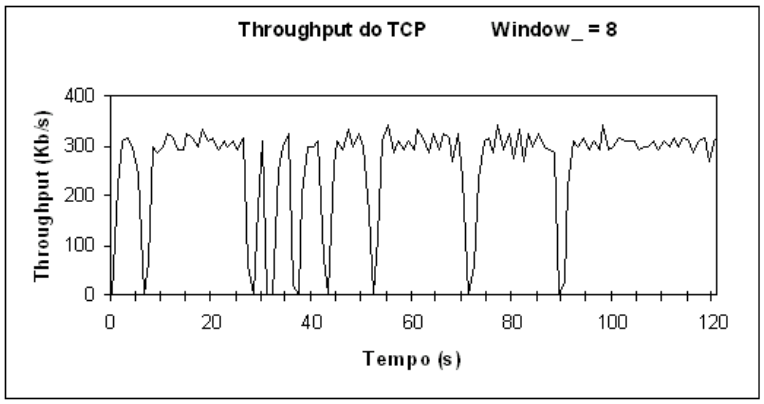


Gráfico 3.3 – *Throughput* da Simulação1 com *window_* = 8

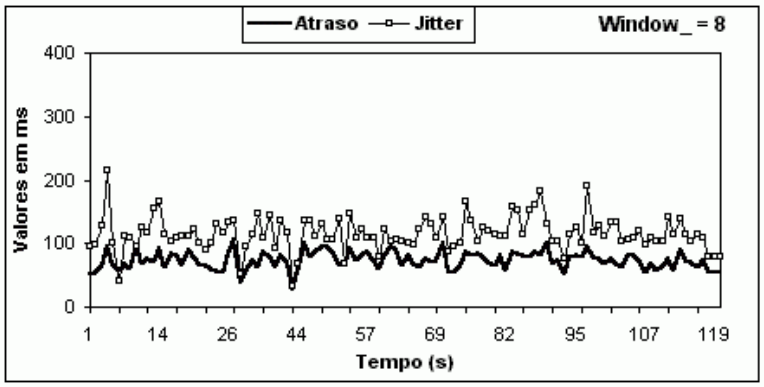


Gráfico 3.4 – Atraso e Jitter da Simulação 1 com window_ = 8

Veja que ainda continua existindo problema de instabilidade. O *throughput* oscila muito durante os 120s de simulação. Entretanto o resultado melhorou em relação ao Gráfico 3.1 onde a vazão chegou a zero por cerca de 17 (dezessete) vezes. Já no Gráfico 3.3 a vazão chegou ou se aproximou de zero apenas 8 (oito) vezes.

A análise dos *traces* mostrou que o *throughput* agregado foi de 31.317, com média de 260,96 Kbps, e **variação** de 341.12 Kbps, sendo este o **maior** valor e o **menor** sendo zero. Isto representa um ganho do no *throughput* agregado de 23.97% em relação ao Gráfico 3.1. As médias de atraso e *jitter* foram respectivamente 66.91 ms e 105,21 ms que também é melhor que o resultado anterior. Veja o Gráfico 3.4.

Dando continuidade à simulação alteramos mais uma vez o tamanho máximo da janela do TCP agora para 4 e submetemos às mesmas condições anteriores. O resultado está ilustrado no Gráfico 3.5.

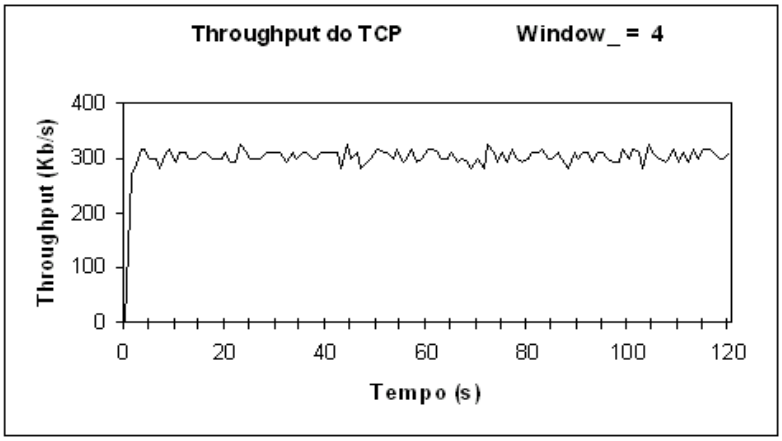


Gráfico 3.5 – *Throughput* da Simulação1 com *window_* = 4

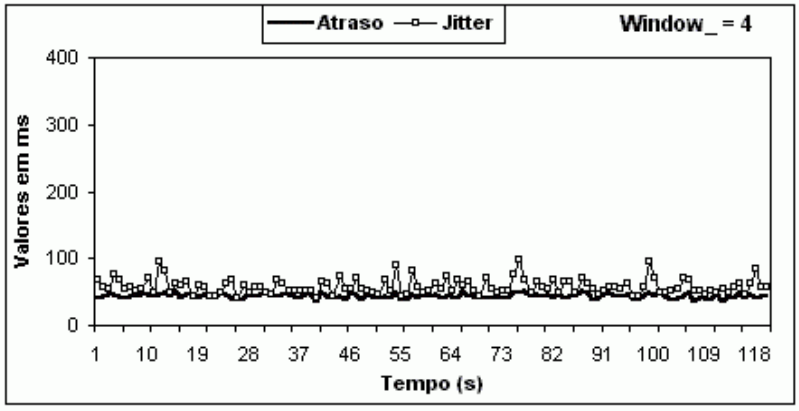


Gráfico 3.6 – Atraso e *Jitter* da Simulação 1 com *window_* = 4

A análise dos *traces* mostrou que o *throughput* agregado foi de 36.309, com média de 302,57 Kbps, e **variação** de 57,60 Kbps, sendo o **maior** valor 324,48 Kbps e o **menor** 266,88 Kbps. Isto representa um ganho no *throughput* agregado de 43.74% em relação ao Gráfico 3.1 e de 15,94% em relação ao Gráfico 3.3. As médias de atraso e *jitter* neste caso foram respectivamente 43,84 ms e 58,15 ms conforme mostra o Gráfico 3.4. Veja na Tabela 3.1 e na Tabela 3.2 um resumo das simulações apresentadas acima.

Windows_	Throughput	Média	Variação	Maior Taxa	Menor Taxa
32	25.260,16	210,50	357,76	349,44	0
8	31.317,12	260,97	341,12	341,12	0
4	36.309,12	302,57	57,60	324,48	266,88

Tabela 3.1 – Quadro comparativo do *Throughput* da Simulação 1

Windows_	Atraso	Atraso Médio	Jitter	Jitter Médio
32	9.089,48	75,74	14.535,68	121,13
8	8.030,31	66,91	12.625,42	105,21
4	5.260,90	43,84	6.978,99	58,15

Tabela 3.2 – Quadro comparativo do Atraso e *Jitter* da Simulação 1

Observe nas tabelas acima que diminuindo o tamanho da janela, o *throughput* agregado aumenta diminuindo a sua variação. A mesma coisa acontecendo com o atraso e *jitter*.

Como pode ser visualizado no Gráfico 3.5, o resultado mostra que o *throughput* do TCP estabilizou-se na faixa de 300 Kbps, não havendo tanta oscilação como nos dois casos tratados acima. Isso mostra a princípio, que o tamanho máximo da janela de transmissão do TCP, em redes locais sem fio *ad hoc multihop*, interfere na sua performance do TCP e que a diminuição desse parâmetro ameniza o problema da instabilidade do TCP.

3.6.1 - Estudo dos Resultados

Pela análise dos dados da simulação (*traces*), percebemos que o problema acontece sempre devido à falha de um nó em alcançar o seu adjacente, gerando uma falha de rota. Se este for um nó intermediário, este nó descarta todos os pacotes da fila para que o nó adjacente reporte uma falha de rota para o nó origem. Após o nó origem receber esta mensagem, ele inicia o procedimento para encontrar uma nova rota. De acordo com o protocolo MAC IEEE 802.11, 7 (sete) [1] é o número de tentativas que um nó deve tentar para ter acesso ao meio antes de reportar uma falha de quebra de *link* [7]. Veja quando isso ocorre.

Observando os *traces* da simulação e o Gráfico 3.3, vemos que a vazão cai para zero em vários momentos. Para entendermos porque isso ocorre, analisamos um trecho do *trace* no intervalo de tempo compreendido entre 28.839s e 28.860s, onde a vazão atinge zero. O que vimos é que o nó 3 não consegue alcançar o nó 4 e falha durante 7 tentativas após sucessivas colisões. Como já foi atingido o número máximo de tentativas definidas pelo protocolo, todos os pacotes enfileirados nesse nó são descartados, e logo em seguida é reportado uma falha na rota do nó 4. Veja na Tabela 3.3, nas linhas 1 a 7 as colisões no nó 3, na linha 8 os pacotes enfileirados no nó 3 são descartados, na linha 9 é indicada uma falha na rota no nó 4. em seguida nas linhas 10 e 11 é dado início o estabelecimento de uma nova rota pelo protocolo DSR.

Veja no Anexo A, detalhes do *trace* resumido na Tabela 3.3 – Parte do *trace* gerado pela simulação 1 com *window_* = 8.

Linha	Tipo	Tempo	Nó Origem	Nó Destino	Evento	Pacote
. . .	. . .	. . .	. . .	. . .	. . .	. . .
1	D	28.839	_**3**_	_4_	**COL**	MAC
2	D	28.840	_**3**_	_4_	**COL**	MAC
3	D	28.841	_**3**_	_4_	**COL**	MAC
4	D	28.854	_**3**_	_4_	**COL**	MAC
5	D	28.857	_**3**_	_4_	**COL**	MAC
6	D	28.858	_**3**_	_4_	**COL**	MAC
7	D	28.859	_**3**_	_4_	**COL**	MAC
8	D	28.860	_3_	_**4**_	**RET**	MAC
9	D	28.860	_3_	_**4**_	**NRTE**	ACK
10	S	28.860			---------	ACK
11	R	28.860			---------	DSR
. . .	. . .	. . .	. . .	. . .	. . .	. . .

Tabela 3.3 – Parte do *trace* gerado pela simulação 1 com *window_* = 8

A colisão e o problema do nó exposto no nó 3, impede o nó 3 alcançar o nó 4. Sabendo que o nó 3 pode perceber o nó 4, ele adiará respostas quando o nó 4 estiver transmitindo. O resultado é que o nó 4 não pode devolver CTS mesmo que ele recebesse o RTS do nó 3 corretamente. Após a falha em receber o RTS 7

(sete) vezes, o nó 1 reporta uma quebra de *link* para o nível superior. Ocorrendo desta forma um evento de falha de rota. Veja que o nó 3 está dentro da faixa de interferência do nó 5.

Agora está mais claro que o problema do nó exposto e colisão evitam um nó intermediário alcançar o seu próximo salto. O esquema de *backoff* randômico usado no nível MAC piora ainda mais esta situação. Considerando que o tamanho de pacotes de dados muito grandes e o envio de pacotes *back-to-back (*colados*)* aumentam a possibilidade do nó intermediário falhar na obtenção do canal, o nó espera um tempo aleatório e tenta outra vez. Isto incrementará o atraso de pacotes ACKs se ele tiver sucesso. Se ele ainda falhar depois de 7(sete) tentativas, uma quebra de *link* será declarada. O resultado disso é reportado sob a forma de falha de rota. Isto explica porque no Gráfico 3.5 não existe o problema da instabilidade, pois o número máximo possível de *back-to-back* é 4 (quatro). Isto reduz em muito a possibilidade de outros nós falharem ao acessar o canal 7 (sete) vezes. Assim, não ocorre falha de rota.

Portanto fica claro que ajustando o parâmetro do TCP, *window_,* é possível diminuir e até mesmo eliminar o problema de instabilidade do TCP. Entretanto o problema ainda existe. Veremos a seguir outro sério problema em redes sem fio *ad hoc multihop* que não pode ser resolvido apenas com ajustes de parâmetros do TCP.

3.7 - O Problema de Injustiça de um Salto

Veremos agora um outro problema sério existente em redes locais sem fio *ad hoc multihop*. Este problema ocorre também no nível MAC do IEEE 802.11. Nós denominamos este problema de **injustiça de um salto.**

Na simulação que mostra este problema, trabalhamos com o mesmo cenário da Figura 3.1, só que agora foram estabelecidas duas conexões TCP, sendo que a primeira começando com 10s e a segunda depois de 30s de simulação. O nosso objetivo continua sendo o mesmo, ou seja verificar a performance do TCP, medindo

o *throughput,* atraso e *jitter* entre estas conexões. Veja a Figura 3.9 que ilustra de maneira geral o cenário da simulação.

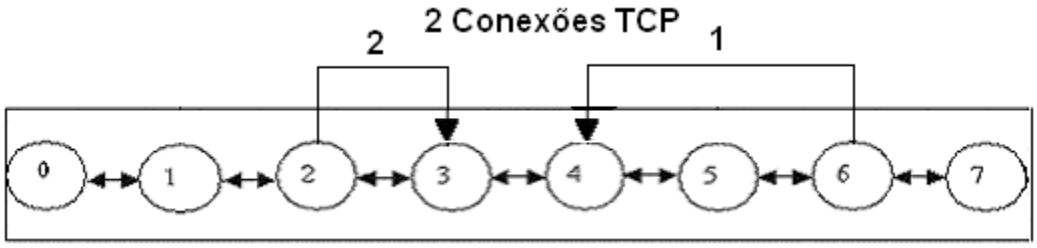


Figura 3.9 – Cenário *Multihop* da Simulação 2

Observe que na primeira conexão existem dois saltos, 6➔5➔4, e a segunda é de apenas um, 2➔3. Fizemos então a simulação de acordo com o cenário acima e medimos o *throughput* do TCP entre estas conexões. Conforme concluímos na Seção 3.6, para evitar problema de instabilidade do TCP, é melhor trabalhar com o tamanho máximo da janela pequeno, que na nossa simulação mostrou que o valor 4 é o ideal. Desta forma continuamos a trabalhar com o parâmetro *window_* com este valor. O resultado pode ser visto no Gráfico 3.7.

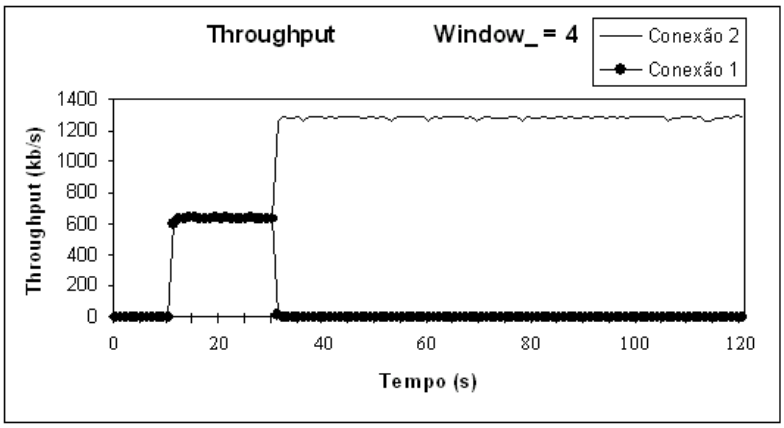


Gráfico 3.7 – *Throughput* da Simulação 2 com *window_* = 4

Analisando o Gráfico 3.7 e os traces da simulação, observamos que o *throughput* da primeira conexão, manteve-se na faixa de 643 Kbps no intervalo de 10s a 30s. O *throughput* agregado foi de 12.714 com **média** de 115,58 Kbps e **variação** de 619,20 Kbps, sendo 635,71Kbps o **maior** valor e 16,51 Kbps o **menor.** No entanto o *throughput* cai completamente para zero depois que a segunda conexão inicia, o

que ocorre aos 31s de simulação, sem conseguir novamente transmitir nenhum pacote. O *througput* da segunda conexão mantém-se sempre na faixa de 1258 Kbps durante o intervalo de 31 a 120s. O *throughput* agregado foi de 115.113 Kbps com **média** de 1.278,92 Kbps e variação de 57,02 Kbps, sendo o **maior** valor 1.299,39 e o **menor** 1.242,36 Kbps. Evidentemente que isto não é justo uma vez que apenas uma conexão monopoliza por completo o meio não permitindo que outra conexão o compartilhe.

Visando resolver este problema aplicamos o mesmo procedimento da seção 3.6, ou seja diminuímos ainda mais o tamanho da janela do TCP para o valor mínimo, 1, submetendo mais uma vez a simulação como o mesmo ambiente anterior. Para este caso, calculamos também o atraso e *jitter.* O resultado pode ser visto no Gráfico 3.8. e no Gráfico 3.9.

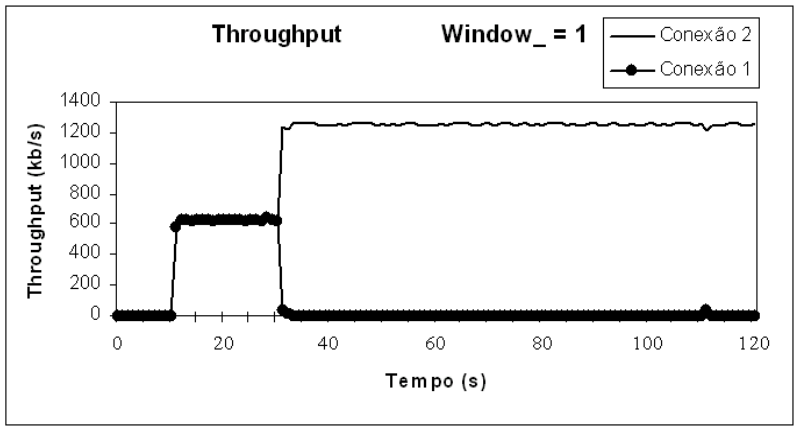


Gráfico 3.8 – *Throughput da* Simulação 2 com window_ = 1

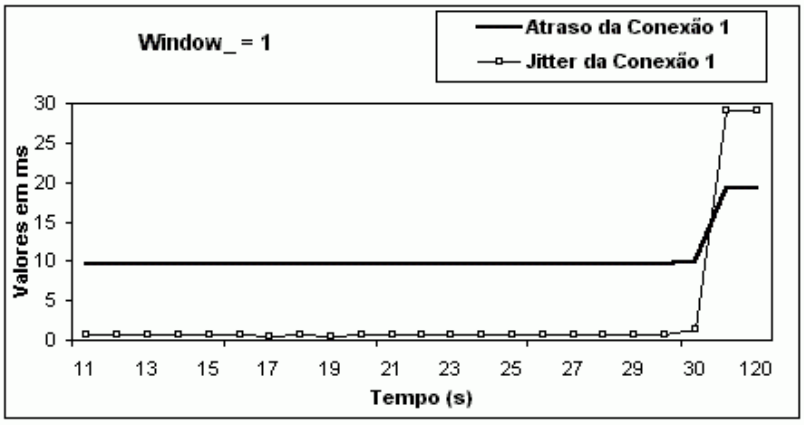


Gráfico 3.9 – Atraso e *Jitter* da Conexão 1 da Simulação 2

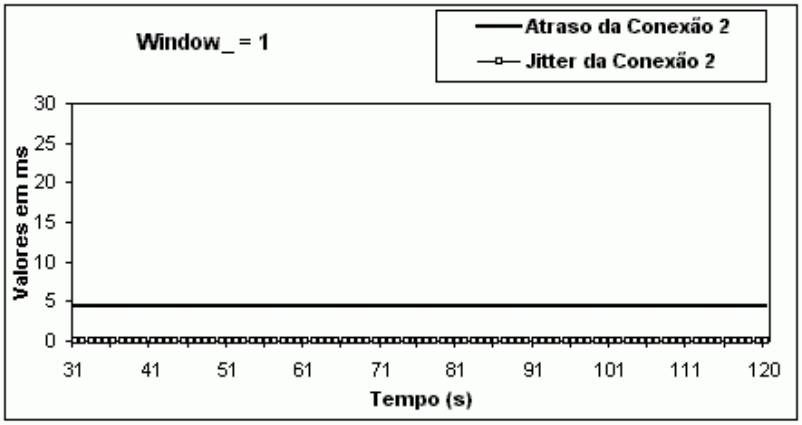


Gráfico 3.10 - Atraso e *Jitter* da Conexão 2 da Simulação 2

Observamos que o resultado não se alterou muito, apenas mudanças pequenas em relação o *throughput* que só foram percebidas analisando os traces da simulação. A primeira conexão teve um *throughput* agregado de 12.557 com **média** de 114,16 Kbps e **variação** de 627,45 Kbps, sendo que o **maior** valor foi 635,71 Kbps e o **menor** 8,26 Kbps**.** Já a segunda conexão teve um *throughput* agregado de 112.776 Kbps com **média** de 1.253,06 Kbps e **variação** de 57,03 Kbps, sendo que o **maior** valor foi 1.258,27 Kbps e o **menor** 1.201,24 Kbps**.** Houve uma ligeira reação da conexão 1 no tempo 110s que gerou as alterações vistas nos gráficos logo acima, fazendo o atraso e *jitter* terem um comportamento diferente conforme mostra o Gráfico 3.9. Entretanto esta reação não foi suficiente para modificar o resultado geral. Podemos observar ainda que tanto atraso quanto *jitter* permaneceram praticamente constantes durante todo o tempo. Isto acontece devido à estabilidade do *throughput.* Veja abaixo um comparativo entre as duas variações do parâmetro *window_* .

Window_	Throughput	Média	Variação	Maior Taxa	Menor Taxa
4	12.714	115,58	619,20	635,71	16,51
1	12.557	114,16	627,45	635,71	8,26

Tabela 3.4 – Quadro Comparativo da Conexão 1

Window_	Throughput	Média	Variação	Maior Taxa	Menor Taxa
4	115.113	1.278,92	57,02	1.299,39	1.242,36
1	112.776	1.253,06	57,03	1.258,27	1.201,24

Tabela 3.5 - Quadro Comparativo da Conexão 2

De maneira geral, a primeira conexão permaneceu em zero praticamente o tempo todo, assim que a segunda iniciou. Isso mostra que este problema não pode ser eliminado apenas com ajuste da janela do TCP, como ocorreu no problema da instabilidade descrito na seção 3.6.

3.7.1 - Estudo dos Resultados

A análise dos *traces* mostra que após a ocorrência de uma falha na rota, reportada pelo nó 5, é iniciado o procedimento para encontrar uma nova rota para o nó 4. Enquanto esta nova rota não é encontrada, os pacotes encaminhados pelo nó 6 ficarão enfileirados aguardando o estabelecimento de uma nova rota. A rota é então estabelecida mas mesmo assim os pacotes TCP do nó 6 não conseguem mais alcançar o nó 4. Esta falha de rota acontece novamente seguidas vezes, impedindo que o nó 6 envie pacotes para o nó 4. Considerando que não há mobilidade dos nós no cenário da Figura 3.9, esta falha na rota, a princípio, não se justifica.

Mas porque ocorre a falha na rota? Analisando os traces da simulação, verificamos que o problema está no nível MAC. Depois que o nó 5 tenta alcançar o nó 4 por 7 vezes, o nível MAC reporta uma quebra de link para o nível superior. A maior causa do nó 5 não alcançar o nó 4 é a colisão [29] neste nó. A Tabela 3.3 mostra parte do trace onde estes eventos acontecem. Segundo [21] a faixa de interferência e percepção (sensibilidade) em redes sem fio é geralmente maior que a faixa de transmissão. Sabendo disso, vemos que o nó 5 pode perceber (sentir) o nó 3, uma vez que o nó 3 está dentro da área de percepção do nó 5, o nó 5 adiará suas transmissões quando o nó 3 estiver transmitindo, assim o nó 5 só pode enviar RTS somente quando o nó 3 não estiver transmitindo. O resultado é que o nó 5 não pode enviar retorno CTS mesmo que receba RTS corretamente do nó 6. Esta, é a reação de uma colisão ocorrida no nó 4 quando o nó 2 e o nó 5 estão transmitindo ao mesmo tempo, apesar do nó 4 não se comunicar diretamente com o nó 2. O nó 4 está dentro da faixa de interferência do nó 2. Veja no Anexo B os detalhes dos *traces* resumidos na Tabela 3.6.

Linha	Tipo	Tempo	Nó Origem	Nó Destino	Evento	Pacote
. . .	. . .	. . .	. . .	. . .	. . .	. . .
1	D	30.064	_5_	_**4**_	**COL**	MAC
2	D	30.065	_5_	_**4**_	**COL**	MAC
3	D	30.067	_5_	_**4**_	**COL**	MAC
4	D	30.072	_5_	_**4**_	**COL**	MAC
5	D	30.085	_5_	_**4**_	**COL**	MAC
6	D	30.092	_5_	_**4**_	**COL**	MAC
7	D	30.106	_5_	_**4**_	**COL**	MAC
8	D	30.106	_**5**_	_4_	**RET**	MAC
9	D	30.106	_**5**_	_4_	**NRTE**	TCP
10	S	30.106			---------	ACK
11	R	30.106			---------	DSR
. . .	. . .	. . .	. . .	. . .	. . .	. . .
12	D	31.209	_**5**_	_4_	**NRTE**	TCP
13	D	42.286	_**5**_	_4_	**NRTE**	TCP
14	D	42.407	_**5**_	_4_	**NRTE**	TCP
15	D	42.407	_**5**_	_4_	**NRTE**	TCP
16	D	42.407	_**5**_	_4_	**NRTE**	TCP
17	D	110.505	_**5**_	_4_	**NRTE**	TCP
. . .	. . .	. . .	. . .	. . .	. . .	. . .
18	D	80.157	_**6**_	-	**TOUT**	TCP
19	D	99.401	_***6***_	-	**TOUT**	TCP

Tabela 3.6 – Parte do *trace* gerado pela Simulação 2 com *window_* = 1

Entretanto, ainda devemos considerar também o fato de que a segunda conexão TCP é de somente um salto. Depois que o nó 2 recebe um pacote ACK do nó 3, ele envia um RTS para requisitar o canal novamente, preparando-se para enviar outro pacote de dados TCP. Uma vez que o nó 3 recebe este RTS, ele responde com um CTS. O nó 2 inicia o envio de pacotes TCP, repetindo este processo durante todo tempo da simulação. Normalmente o tamanho desses pacotes de dados é muito maior que o tamanho dos pacotes de controles. Se o nó 5 enviar um RTS para o canal do nó 4, este RTS irá colidir no nó 4. Assim a única possibilidade do nó 5 acessar o canal do nó 4 é enviando um RTS antes que o nó 2 envie um RTS para o nó 3. Veja que isto só acontecerá após o nó 3 finalizar o envio de pacotes ACK, além disso o tempo de abertura da janela do nó 5 para acessar o canal é muito pequeno. Outro fator que contribui é o fato do esquema de *backoff* exponencial

binário do nível MAC, favorecer sempre a última estação que transmitiu com sucesso, que neste caso foi o nó 2. Assim dificilmente o nó 5 vencerá a disputa. Depois de sete falhas ele desiste e reporta uma quebra de link para o nível acima, ocorrendo um evento de falha de rota.

Observe que a falha da rota ainda ocorrerá durante 6 (seis) vezes entre os tempos 31.209s e 110.505 pelo mesmo motivo. Entre este intervalo de tempo ocorreram dois eventos de *timeout* (TOUT) o primeiro no tempo 80.157 e outro no tempo 90.401 conforme mostra a Tabela 3.6. Este evento deve-se às inúmeras tentativas feitas sem sucesso após a falha de rota.

Além desse, ainda existem outros problemas de injustiça. A causa de todos eles é sempre a mesma: o nível MAC não trabalha bem em redes locais sem fio *ad hoc multihop*. Algumas outras abordagens para este problema podem ser encontradas em [22] [23] [24] [25].

RESUMO DO CAPÍTULO

Neste capítulo vimos dois problemas que ocorrem em redes locais sem fio *ad hoc multihop*. Que são a **instabilidade** e a **injustiça.** Estes problemas foram percebidos pelo TCP, causando sérias degradações na performance e injustiça na disputa pelo meio. Vimos que o primeiro foi contornado apenas modificando-se o tamanho da janela de transmissão, enquanto que o segundo não pode ser resolvido da mesma forma. No próximo capítulo, estaremos propondo uma solução e estudando outras possíveis que tentam resolver ou amenizar tal problema ou que estejam relacionadas com o este trabalho.

Capítulo 4

Soluções Propostas e Análises

Ficou claro pelo que foi mostrado no capítulo anterior que o protocolo MAC IEEE 802.11 não trabalha bem em redes *ad hoc multihop*. Vimos dois problemas um que tratava da **instabilidade** do TCP e outro da **injustiça** de um salto. No primeiro problema vimos que isto pode ser amenizado e em alguns casos até mesmo resolvido apenas ajustando o parâmetro *window_* do TCP, onde concluímos que a partir do tamanho 4 *o throughput* medido do TCP apresentava-se estável. Como conseqüência, o atraso e *jitter* também estabilizaram-se. Já o segundo problema não foi possível resolver apenas com ajustes desse parâmetro. Para verificarmos isso atribuímos o valor mínimo, 1, e mesmo assim o problema da injustiça persistia. Neste capítulo estaremos buscando alternativas para que de alguma forma possamos amenizar este segundo problema de maneira que o meio não seja monopolizado por uma conexão e sim compartilhado de maneira ideal e justa para meios com esta característica.

Segundo a literatura existente, as soluções prováveis e possíveis, implementadas que de alguma forma amenizam tal problema, estão localizadas em dois níveis: MAC e roteamento.

As soluções propostas para o nível MAC estão baseadas em:

A) Alterações na política de funcionamento do algoritmo de *backoff*

As soluções baseadas nesse mecanismo, procuram estabelecer uma política diferentes da implementada originalmente pelo protocolo MAC IEEE 802.11. Desta forma cria-se uma nova lógica de incremento desse contador, de maneira que ele funcione melhor que o original.

B) Alteração no controle janela de contenção

A janela de contenção ou disputa, é incrementada ou decrementada de acordo com o grau de disputa pelo meio. As alterações visando uma melhora desse procedimento, giram em torno de criação de novos intervalos ou alteração dos limites dos atuais.

C) Criação de índices de justiça para acesso ao meio

O acesso ao meio eventualmente pode tornar-se uma tarefa complicada. Pelo que foi visto no capítulo anterior, isto às vezes prejudica seriamente conexões em função de outras. A criação de índices de justiça vista estabelecer critérios justos para as estações acessarem o meio e o compartilharem simultaneamente, sem privilegiar nenhuma ou outra estação.

D) Mecanismos de prioridade para fluxos e Requisitos de Qualidade de Serviço (QoS);

Criando mecanismos de prioridades para fluxos, estamos tratando os fluxos de forma diferenciada de acordo com a sua importância. Desta forma poderemos refinar cada vez mais este mecanismos incluindo novos requisitos de Qualidade de Serviço [67][68] que podem culminar com a resolução ou amenização dos problemas citados no capítulo anterior.

As soluções propostas a nível de roteamento, baseiam-se em:

A) Verificação de protocolos de roteamento mais apropriados para determinados cenários

As abordagens relacionadas a este aspecto não resolvem os problemas descritos anteriormente, elas apenas tentam verificar dentro de determinadas situações de cenários, tipo de tráfego qual protocolo de roteamento apresenta melhores resultados. O protocolo sugerido, neste caso, deverá ser o melhor se satisfeitas um conjunto de condições pré-estabelecidas.

B) Modificação no funcionamento dos atuais protocolos de roteamento

Esta estratégia visa analisar o funcionamento dos protocolos de roteamento atuais, associado aos problemas existentes e propor mudanças de forma a evitar que o

problema não ocorra mais. Isso passa pela construção de novas classes ou alterações das existentes visando uma melhor resultado que o protocolo original.
Geralmente as estações que formam uma rede *ad hoc* compartilham um canal *broadcast* por isso um protocolo de acesso ao meio eficiente e efetivo torna-se extremamente necessário no compartilhamento dos recursos de largura de banda. Devido a natureza distribuída de uma rede *ad hoc* e da ausência de um controle central, protocolos de controle de acesso randômico distribuído são mais freqüentemente preferidos, para que de maneira centralizada coordene o acesso ao meio. Este segundo motivo proíbe os nós trocarem altas quantidades de informações a fim de manter o estado global da rede sob controle e estável. Este fato torna-se muito mais crítico em cenários com a presença da mobilidade entre as estações. Porém, protocolos de acesso randômico usualmente deparam-se com os problemas de terminal escondido e exposto. Hoje, o protocolo conhecido mais importante para redes *ad hoc* é de fato o IEEE 802.11 DFWMAC que tem sido usado extensivamente para estudar a performance de protocolos dos níveis superiores como, roteamento etc [26]. No entanto na presença de terminais escondidos e eventualmente, a mobilidade, o IEEE 802.11 pode induzir sérias injustiças entre estações que disputam acesso ao meio, conforme vimos no capítulo anterior. Uma análise mais criteriosa desse protocolo mostra que isto acontece principalmente, **mais não exclusivamente,** devido ao algoritmo de *backoff* exponencial binário do (BEB) usado pelo DFWMAC para minimizar contenção quando a densidade do tráfego aumenta. De acordo com o algoritmo BEB, descrito no capítulo 3, o tamanho da janela de contenção de uma estação dobra a cada transmissão sem sucesso até que este valor alcance o valor máximo estabelecido, e retorna para o valor mínimo se o pacote de dados foi transmitido com sucesso. No entanto em uma rede *ad hoc* com desigual tráfego de dados distribuída, uma estação que menos competiu pelo acesso ao canal, tem mais chances de transmitir um pacote de dados e ao final atribuir à janela de contenção o valor mínimo. Assim sendo aquelas estações terão menor tempo de *backoff* que outras estações que já tenham falhado alguma vez.

Quando a carga do tráfego é alta, as estações que tiverem maior tempo de *backoff* podem sofrer excessivo atraso de acesso, sérias degradações do *throughput* e por último *starvation, à*medida que a carga do tráfego aumenta.

Neste capítulo estaremos propondo uma solução e abordando outras que já foram propostas baseadas em alguma dos aspectos citados acima, no sentido de minimizar os efeitos da instabilidade e da injustiça na disputa pelo acesso ao meio compartilhado. Os cenários apresentados neste capítulo seguem de maneira geral o mesmo ambiente de simulação apresentado na seção 3.2 do capítulo anterior. Estaremos também verificando e comparando a performance dos protocolos de roteamento DSR, AODV e DSDV com o novo cenário proposto.

4.1 - Solução Proposta: Distribuição Não-Alinhada dos Nós (DNA)

Considerando a análise dos problemas e o cenário das simulações feitas no capítulo anterior, propomos mudar a disposição das estações objetivando oferecer mais rotas alternativas para os fluxos de dados de maneira que não tenhamos apenas uma opção de rota como no cenário anterior. Com esta mudança, a faixa de transmissão entre as estações que competem pelo meio será percebida entre elas. Desta forma esperamos oferecer condições de disputas mais justas, objetivando amenizar o problema da falha de rota causada por eventuais terminais escondidos e expostos muitos, comuns em um cenário com estações alinhadas. Consideramos também que em função da liberdade e flexibilidade de localização das estações, características atraentes de redes locais sem fio, as situações de alinhamento similar serão minimizadas. No nosso caso, mesmo desconsiderando a mobilidade, ainda existe o outro fator que é a liberdade de localização dentro de uma organização, obedecendo é claro, os limites impostos pelo padrão. Assim sendo, se conseguirmos estabelecer cenários estáticos onde as estações estejam dispostas em uma organização aleatória, não exatamente alinhadas e localizadas em distâncias distintas, estaremos oferecendo um cenário que possibilita situações mais justas de disputa. Veja o cenário proposto na Figura 4.1.

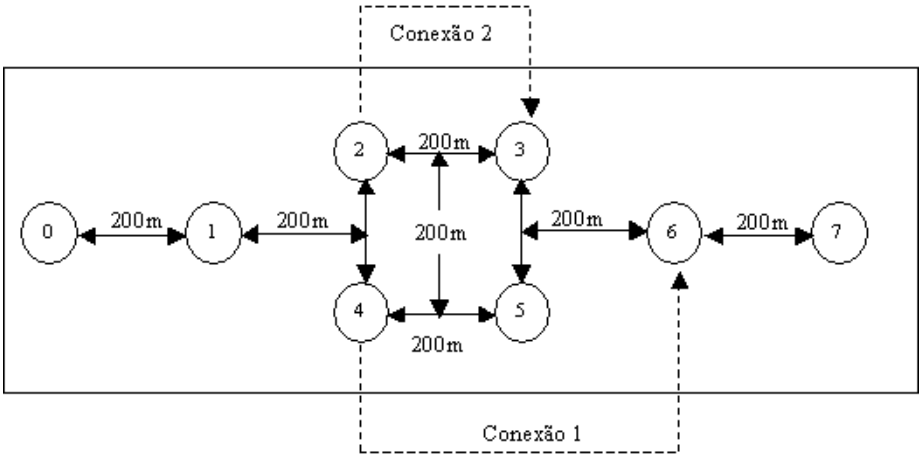


Figura 4.1 – Novo Cenário *Multihop* Proposto

No cenário proposto consideramos duas conexões TCP, sendo a primeira estabelecida entre os nós 4 e 6 iniciando em 10s e a segunda entre os nós 2 e 3 iniciando em 30s. Observe também que procuramos manter a distância a mesma entre os nós adjacentes das simulações 1 e 2 do capítulo anterior, ou seja 200m. Isto só não aconteceu entre os seguintes nós em função da própria distribuição dos nós no cenário:

- **1 - 2, 1 - 4, 3 - 6 e 5 - 6 :** distância de aproximadamente 173,20m ;
- **2 - 5 e 3 - 4:** distância de aproximadamente 282m, maior que o permitido pelo padrão que é 250m .

Veja que com este cenário o nó 4 pode alcançar o nó 2 mas não pode alcançar o nó 3. Da mesma forma o nó 2 pode alcançar o nó 4, pois eles estão dentro da mesma faixa de transmissão. Isto não acontece entre os nós 3 – 4 e 2 – 5. Isto significa que sempre que o nó 4 estiver transmitindo, o nó 2 saberá disso pois o nó 2 pode ouvir o nó 4.

O tempo simulado também foi mantido em 120s. A escolha desse cenário simétrico foi feita com o objetivo de conseguir-se um cenário próximo ao mostrado nas simulações anteriores, diferenciando-se apenas na disposição das estações, de maneira que os resultados das simulações pudessem ser melhor comparados. O resultado da simulação é mostrado no Gráfico 4.1.

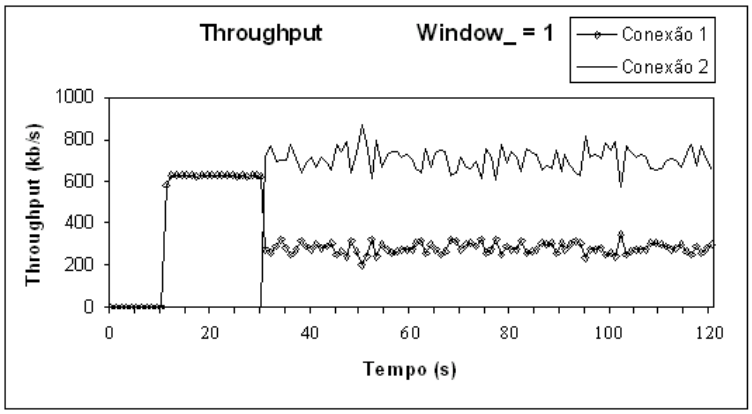


Gráfico 4.1 – *Throughput* Simulação 3 usando o protocolo DSR

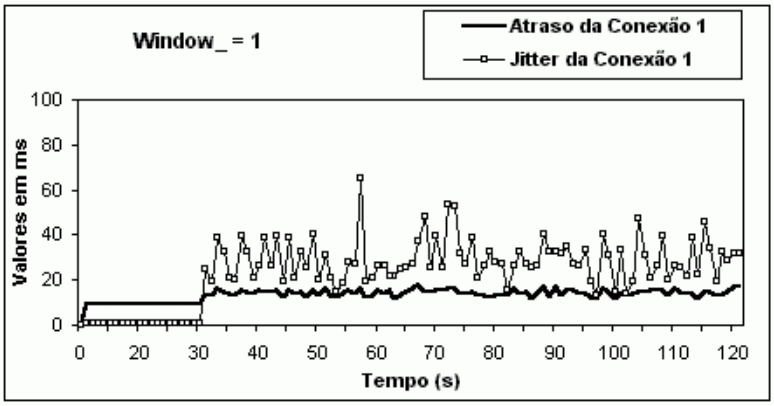


Gráfico 4.2 - Atraso e Jitter da Conexão 1 da Simulação 3

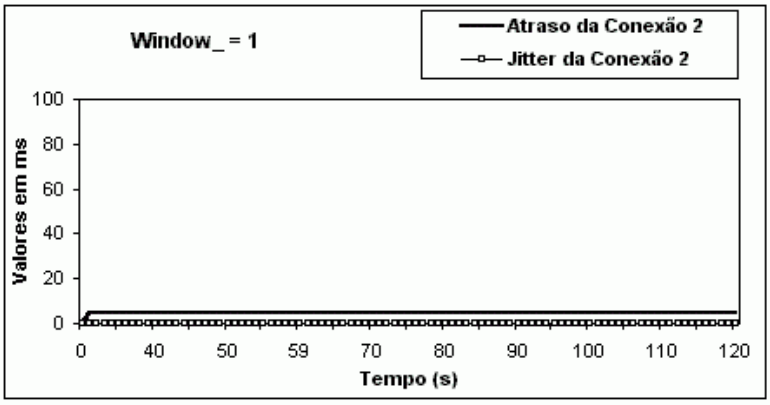


Gráfico 4.3 - Atraso e Jitter da Conexão 2 da Simulação 3

Analisando o Gráfico 4.1 e os *traces* gerados, podemos perceber que o *throughput* da conexão 1 no intervalo de 10 a 30 segundos permanece na faixa de 627 kbps com pequenas variações. A partir do tempo 31s o *throughput* cai para a faixa de 250 Kbps com **média** aproximada de 279 kbps, **variação** de 148 kbps, sendo o **menor** *throughput* 198 kbps, ocorrendo no tempo 50s, e o **maior** 346 kbps, ocorrendo no tempo 102s. Já a conexão 2 permanece com o *throughput* na faixa de 700 kbps com **média** aproximada de 713 kbps, **variação** de 296 kbps, sendo o **menor** *throughput* 575 kbps, ocorrendo no tempo 102s e o **maior** 871, ocorrendo no tempo 50s. Observe que a variação do *throughput* é muito maior na segunda em relação à primeira conexão. Em relação ao atraso e *jitter* percebemos que para a conexão 1 há uma variação muito grande com um **atraso médio** de 12,38 ms e a **média do *jitter*** com 22,05 ms. Para a conexão 2 eles são praticamente constantes durante todo o tempo com **atraso médio** de 3,25 ms e o ***jitter*** com valor zero. Essas diferenças entre a conexão 1 e a conexão 2 ocorre em função da primeira ser de apenas um salto. Entretanto esse resultado já é melhor que os resultados da simulação 2.

Fazendo uma comparação entre as simulações 2 e 3, podemos encontrar duas diferenças entre o Gráfico 3.7 e o Gráfico 4.1. A primeira, é que no Gráfico 4.1 o problema da injustiça não aparece. As duas conexões conseguem transmitir pacotes simultaneamente, o que é justo. A segunda diferença fica por conta da grande variação do *throughput* entre um gráfico e outro. Precisamos agora responder a duas perguntas:

1. Por que a variação do *throughput* é tão grande entre os gráficos?
2. Por que o problema da injustiça não ocorreu?

Análise da Variação do Throughput

Nós trabalhamos nessa simulação com o parâmetro *window_* igual a 1. Conforme já foi visto no capítulo anterior este valor já é o mínimo e apresentou-se como sendo o mais estável entre os tamanhos 32, 8 e 4. Isto mostra que se aumentarmos o tamanho da janela, a variação da vazão também irá aumentar. Mas, como não é

possível encontrar um resultado melhor para a variação da vazão, modificando apenas o parâmetro *window_,* por que ela varia tanto?.

Analisando os *traces,* da simulação percebemos que isto ocorre em função das colisões no nível MAC. Todas as colisões ocorreram sempre nos nós 2, 4 ou 5. A colisão no nó 2 acontece sempre que este nó tenta acesso ao canal do nó 3. O mesmo acontece de 4 para 5 e de 5 para 6. O motivo das colisões nesses nós é sempre o mesmo mudando apenas de nó. Para entendermos melhor, fixamos como exemplo as colisões que ocorrem no nó 2. Sempre que o nó 2 tiver tentando acesso ao meio, ele irá transmitir um pacote RTS. Se ao mesmo tempo o nó 4 tentar acessar o canal do nó 5 enviando um pacote RTS ou que o nó 5 esteja transmitindo, ocorrerá uma colisão pois o nó 4 está dentro da faixa de transmissão do nó 2. Assim, toda comunicação que ocorre no nó 2 é ouvida pelo nó 4 e vice-versa. Da mesma forma podem ser explicadas as colisões que acontecem nos nós 4 e 5.

Perceba que apesar do número alto de colisões que ocorrem, as duas conexões conseguem transmitir pacotes, mesmo que para isso o *throughput* sofra alguma variação, que não chega a comprometer a transmissão. A seguir iremos responder a segunda pergunta feita acima.

Análise do Problema de Injustiça

Conforme mostrado no Gráfico 3.7., quando a conexão 2 iniciava, a vazão da conexão 1 caia subitamente para zero permanecendo até o final da simulação, o que não ocorre neste cenário. Considerando que o meio do IEEE 802.11 é compartilhado, ele deve ser usado por todas as conexões sem que nenhuma delas o monopolize por completo, o que ocorre no Gráfico 3.7. O cenário proposto tem o objetivo de proporcionar uma disputa pelo meio mais justa, distribuindo melhor as estações de maneira que não se formem cenários que proporcionem situações que levem a injustiças no acesso ao meio, devido a problemas de terminal escondido, exposto ou interferência entre estações que disputam o acesso ao meio.

No momento que deslocamos o nó 4 para uma localização próxima do nó 2 e o nó 5 para próximo do nó 3, deixando-os a uma distância em que um ouve o outro, no neste caso 200m, estamos proporcionando uma disputa mais justa entre eles, já que problemas de terminal escondido,ou exposto são minimizados. Sempre que, por exemplo, o nó 2 estiver tentando acesso ao meio, o nó 4 estará ouvindo e fazendo o mesmo, uma vez que as conexões 1 e 2 são simultâneas a partir do tempo 30s. Desta forma a política de *backoff* do MAC poderá funcionar melhor pois esta política tem um critério de beneficiar sempre a última estação que conseguiu acesso ao meio. Como elas estarão disputando em condições iguais o acesso ao meio, a política de *backoff* do MAC ora beneficiará uma, ora beneficiará a outra. Eliminando neste caso, o problema de injustiça descrito no capítulo anterior, apesar do alto número de colisões.

O problema de colisões continua ocorrendo e degradando a performance do TCP. Entretanto a análise do *traces* também mostrou que por nenhum momento ouve eventos de falha (NRTE) de rota ou *timeout* (TOUT). Isso pode ser visto pelo Gráfico 4.1, em função do *throughput* que nunca chegou a zero em nenhuma das conexões a partir do momento que elas iniciaram. Isto prova que nenhuma das estações precisou fazer acesso ao meio sem sucesso, 7 vezes, o que ocasionaria uma ocorrência de falha de rota acionando os níveis acima para o estabelecimento de uma nova rota.

Para validarmos ainda mais esta proposta, simulamos vários cenários diferentes, com outras conexões entre outros nós, mais sempre oferecendo condições de disputa entre os nós. Uma das situações foi similar à proposta, alterando o sentido de transmissão do nó 6 para o nó 4 e mantendo o restante das configurações idênticas as condições anteriores. O resultado pode ser visto no Gráfico 4.4.

Os resultados encontrados são similares aos do Gráfico 4.1., inclusive os valores da vazão. A diferença fica apenas por conta das variações que são diferentes em função das conexões serem diferentes.

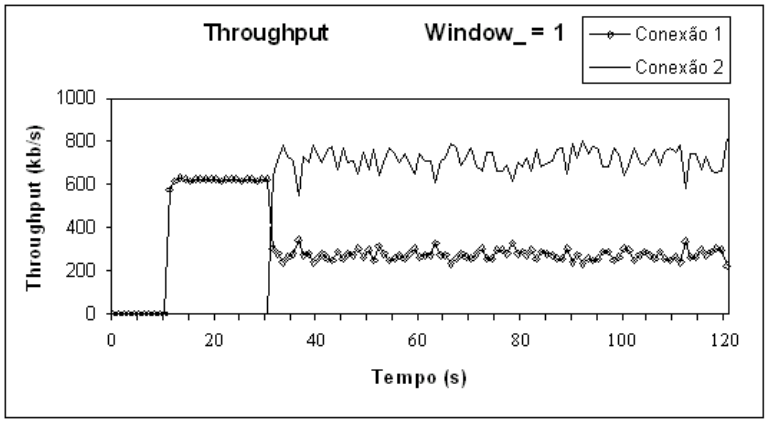


Gráfico 4.4 – *Throughput* Simulação 3 usando protocolo DSR (Conexão 1 - 6➔4)

Criamos então um outro cenário similar ao mostrado na Figura 4.1, só que agora estabelecendo a conexão 1 entre os nós 5➔4 e mantendo a conexão 2 entre os nós 2➔3, como mostra a Figura 4.2.

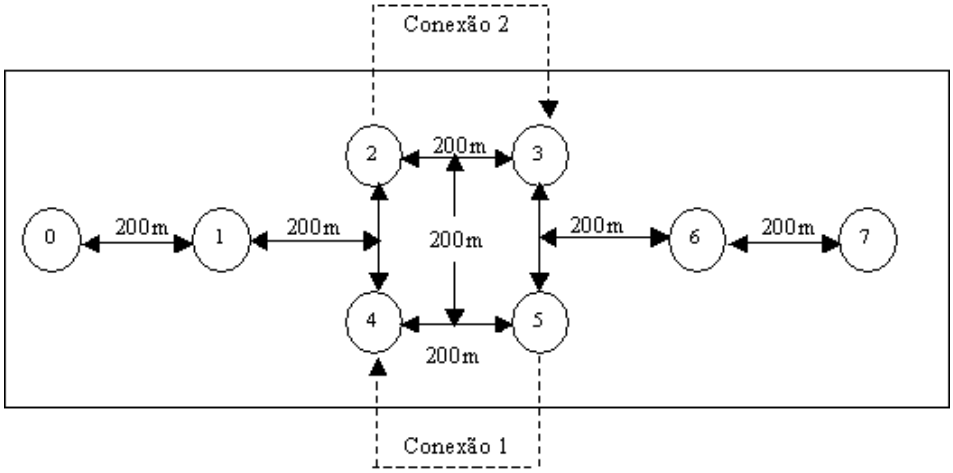


Figura 4.2 – Novo Cenário Alterado

A simulação foi feita levando-se em consideração o mesmo ambiente da anterior em relação a protocolos, tempo de inicio e final de cada conexão. O resultado é mostrado no Gráfico 4.5.

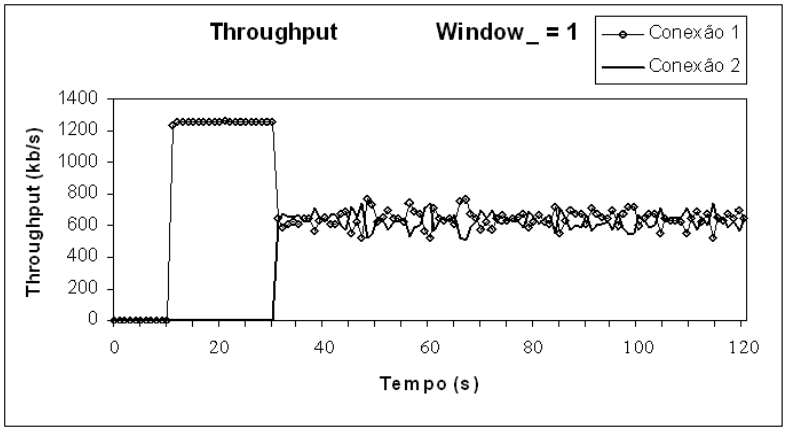


Gráfico 4.5 – *Throughput da* Simulação 4 com disputa justa

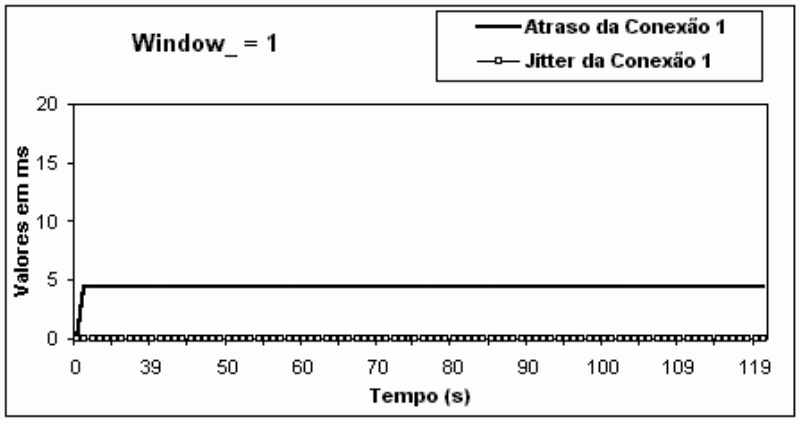


Gráfico 4.6 - Atraso e Jitter da Conexão 1 da Simulação 4

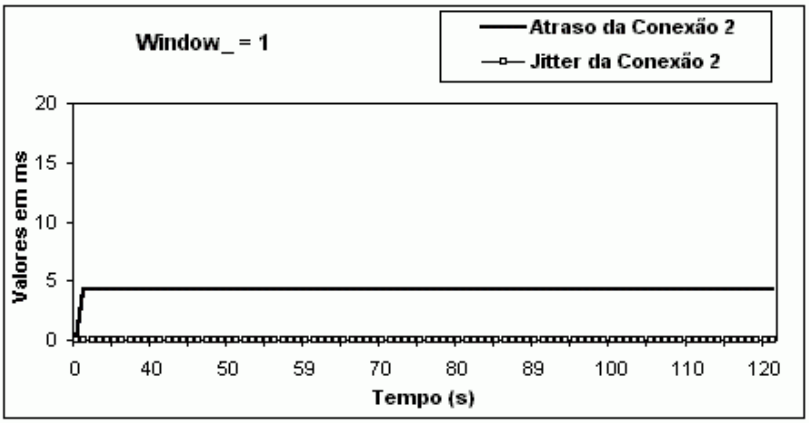


Gráfico 4.7 - Atraso e Jitter da Conexão 2 da Simulação 4

Observando o Gráfico 4.5, percebemos que o *throughput* das duas conexões é aproximadamente a mesmo após os 30s da simulação, tendo uma variação muito próxima. O *throughput* da conexão 1 permanece na faixa de 1250 Kbps com pequenas variações até os 30s caindo em seguida para a faixa de 700 Kbps com alguma variação juntamente com a conexão 2. O fato das duas conexões ficarem com *throughput* aproximadamente iguais, é devido as duas conexões serem apenas de um salto. Veja que o *throughput* da primeira cai quase pela metade e permanece com este valor até o final da simulação. A mesma coisa acontece com a segunda. Perceba que a largura da banda foi dividida entre as duas conexões. Em relação ao atraso e *jitter*, verificamos que eles ficaram rigorosamente iguais, sendo o que o atraso ficou constante em 4,34 ms e o *jitter* em zero, não tendo variação.

Fizemos a mesma simulação proposta no cenário da Figura 4.2, só que modificando o tamanho do parâmetro *window_* de 1 para 4. O resultado está ilustrado no Gráfico 4.8.

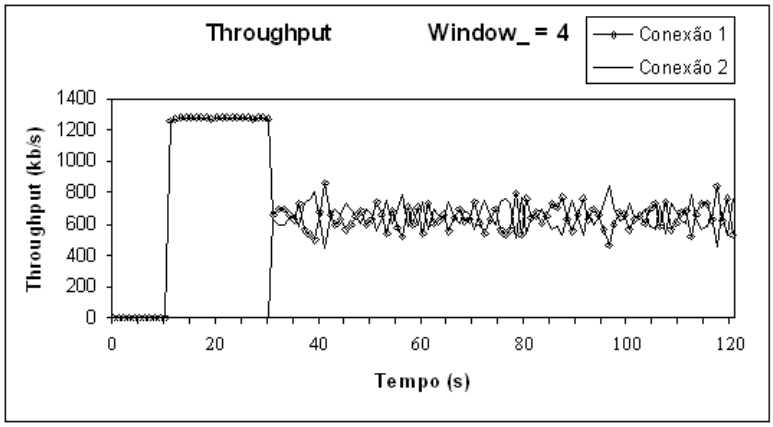


Gráfico 4.8 – *Throughput* da Simulação 4 com window_ = 4

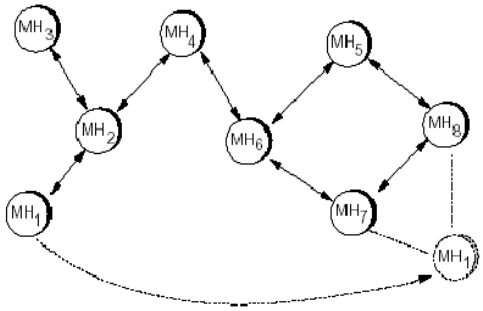
Observe que os resultados gerados no Gráfico 4.8, são semelhantes ao do Gráfico 4.5. As diferenças ficam apenas por conta da variação do *throughput* que oscila um pouco mais no Gráfico 4.8. Fizemos a mesma simulação com o parâmetro *window_* com valores 8 e 32. O resultado foi que *throughput* variou mais ainda em relação ao Gráfico 4.5. Isto ocorre pelos mesmos motivos explicados no capítulo anterior. Da mesma forma atraso e o *jitter* permaneceram dentro dos mesmos valores da simulação anterior.

4.2 - Performance do TCP com os Protocolos DSDV e AODV

Em todas as simulações utilizadas até agora, utilizamos sempre o protocolo de roteamento DSR. Vamos verificar agora o comportamento do *throughput* para o mesmo cenário proposto na Figura 4.1 sendo usado outros protocolos de roteamento como AOVD e DSDV [64]. Antes vamos conhecer uma visão geral de cada um desses protocolos

4.2.1 - Destination-Sequenced Distance-Vector Routing (DSDV)

DSDV [64] é um protocolo pró-ativo baseado no mecanismo clássico de roteamento *Bellman-Ford* [65], com a eliminação de *loops*. Em cada nó existe uma tabela com todos os possíveis destinos dentro da rede, e o número de saltos até cada um deles. Cada entrada nesta tabela é marcada com um número de sequência determinado pelo nó destino. Este número tem a função de distinguir rotas velhas de rotas novas, evitando a formação de *loops*. Para realizar a atualização o protocolo dispõe de dois tipos de pacotes chamados *full dump*, onde todas as informações de roteamento são transmitidas, e os pacotes *increment*, que apenas completam a informação enviada no último *full dump*. Cada *broadcast* de novas rotas inclui o endereço do destino, o número de saltos até o destino, o número de sequência da informação original sobre o destino e um número de sequência para o broadcast. Utiliza-se sempre a rota com o número de sequência mais recente. No caso de duas atualizações terem o mesmo número de sequência, utiliza-se a rota com menor métrica. Veja a Figura 4.3.



Destination	Metric	Sequence number
MH_1	2	S406_MH_1
MH_2	1	S128_MH_2
MH_3	2	S564_MH_3
MH_4	0	S710_MH_4
MH_5	2	S392_MH_5
MH_6	1	S076_MH_6
MH_7	2	S128_MH_7
MH_8	3	S050_MH_8

Destination	Metric	Sequence number
MH_4	0	S820_MH_4
MH_1	3	S516_MH_1
MH_2	1	S238_MH_2
MH_3	2	S674_MH_3
MH_5	2	S502_MH_5
MH_6	1	S186_MH_6
MH_7	2	S238_MH_7
MH_8	3	S160_MH_8

Figura 4.3 – Movimento e atualização de tabelas de roteamento

Veja no Gráfico 4.9 o resultado da simulação utilizando este protocolo. Podemos observar que de uma maneira geral o *throughput* não alterou-se de maneira significativa, apenas mudanças pontuais. Desta forma o atraso e o *jitter* não irá mudar muito também.

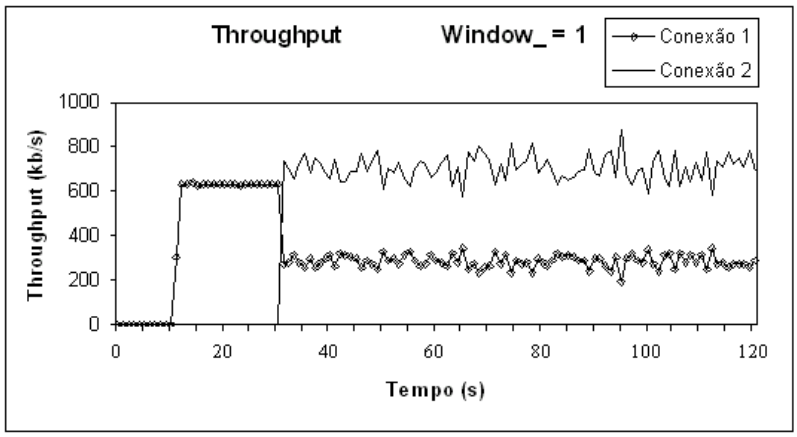


Gráfico 4.9 - *Throughput* da Simulação 3 Utilizando DSDV

4.2.2 - Ad-Hoc On-Demand Distance Vector Routing (AODV)

No protocolo AODV [64], o nó origem, antes de requisitar uma rota para um destino, consulta sua própria tabela de rotas. Caso não encontre nenhuma informação de rota para o destino, é feita uma requisição de rotas aos nós vizinhos. Quando as requisições de rotas são propagadas pela rede, todos os nós atualizam suas tabelas com relação ao nó origem. Se em um determinado momento o nó origem não receber uma resposta, ele pode fazer uma nova requisição ou assumir que o destino está indisponível.

Cada nó ao receber uma requisição de rotas verifica se o pedido é para ele. Se for, o nó envia uma resposta à requisição de rotas por *unicast* pelo caminho reverso e com o mesmo número de seqüência da requisição. Caso a requisição não seja para ele, o nó verifica se existe em sua tabela de roteamento uma rota para o destino. Se existir, ele responde à requisição de rotas por *unicast* pelo caminho reverso. Esta resposta tem o mesmo número de seqüência da requisição de rotas. Caso não tenha a informação em sua tabela, repassa o pedido aos seus vizinhos.

Cada nó ao repassar a mensagem aos vizinhos cria uma rota reversa para que a resposta de rota possa voltar até o nó origem. Quando a requisição de rota chega no destino, este sabe exatamente quantos nós existem no caminho. O destino gera uma resposta com a seqüência da requisição. Cada nó que participou do caminho da requisição repassa a resposta para o nó anterior até que chegue à origem, quando a rota completa é criada. Cada nó conhece apenas as informações referentes ao seu nó e não a rota completa, como no DSR. Na Figura 4.4a é mostrada a requisição de rotas no algoritmo AODV partindo da origem (O) para o destino (D) e na Figura 4.4b a resposta a este pedido. Sendo que este ocorre pelo primeiro caminho pelo qual a requisição chegou ao destino, só que desta vez os nós sabem apenas quem são os seus vizinhos [36].

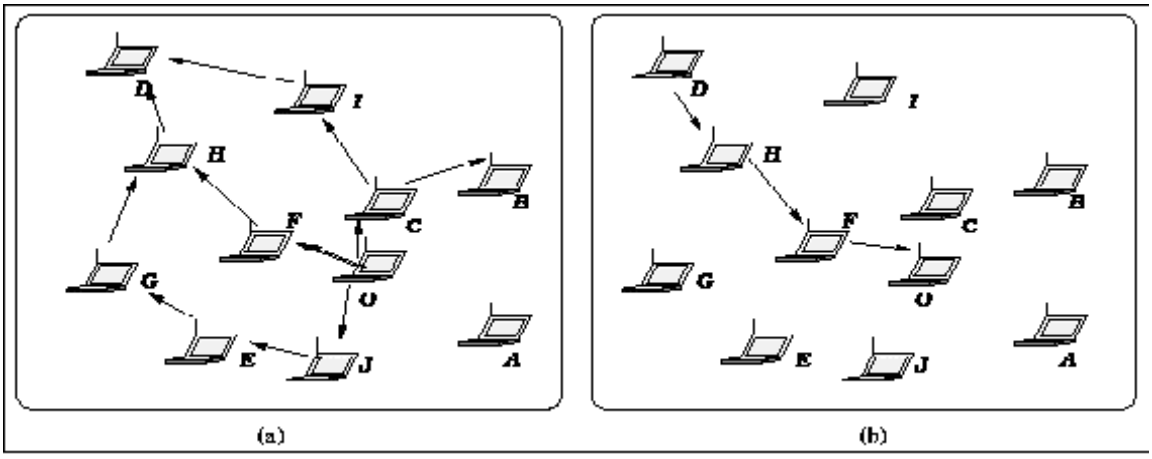


Figura 4.4 – Exemplo de Funcionamento do AODV

Para manter as rotas funcionando de forma correta, o AODV exige que cada nó envie uma mensagem de *Hello* periodicamente aos seus vizinhos. Esta mensagem significa que o nó continua presente e que as rotas dependentes dele continuam válidas. Se algum nó parar de enviar mensagens de *Hello*, o vizinho assume que o nó se moveu e assinala o *link* com o nó como perdido. Nesse caso o nó avisa a todos os nós que dependiam desse *link*, através de uma requisição não solicitada de rotas informando que o mesmo não está mais disponível. Este aviso é propagado pela rede até a origem, que escolhe se utiliza novamente o protocolo de requisição de rotas ou simplesmente interrompe a transmissão para o destino. Veja no Gráfico 4.10 o gráfico gerado utilizando este protocolo.

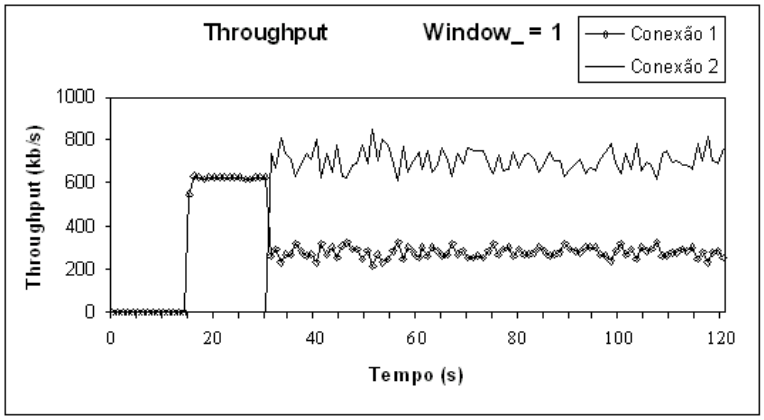


Gráfico 4.10 – *Throughput* da Simulação 3 Utilizando AODV

Observe que as alterações na variação o *throughput* de ambas as conexões não são significativas. Isto mostra que nesse cenário o protocolo de roteamento não interfere de maneira a comprometer o tráfego. Veja abaixo um resumo comparativo dos resultados obtidos utilizando os três protocolos de roteamento: DSR, AODV e DSDV.

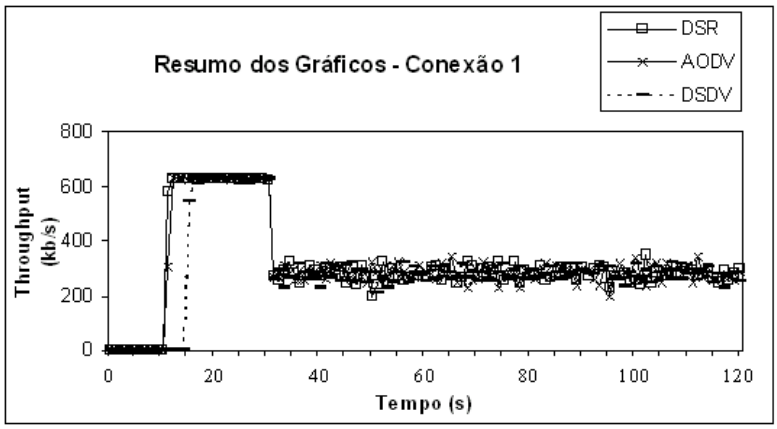


Gráfico 4.11 – Resumo Comparativo da Conexão 1

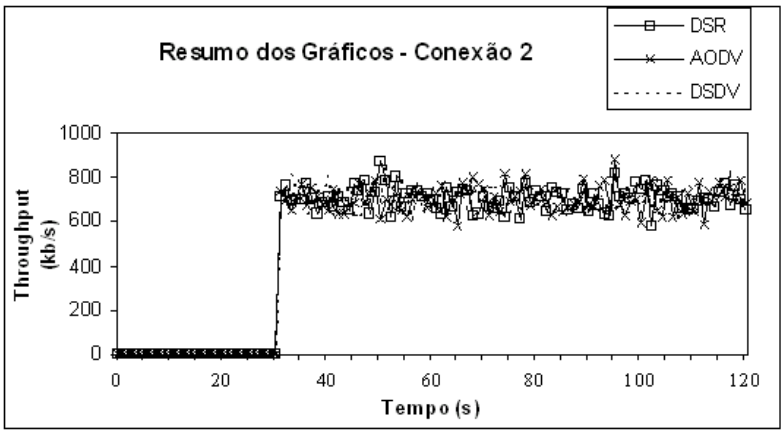


Gráfico 4.12 – Resumo Comparativo da Conexão 2

4.3 - Outras Soluções Propostas e Trabalhos Relacionados

Abordaremos agora alguns trabalhos que foram propostas objetivando resolver ou amenizar o problema do protocolo MAC IEEE 802.11. Conforme já antecipamos no início deste capítulo as abordagens para atacar este problema podem se concentrar tanto no nível MAC quanto nos protocolos de roteamento utilizados nesse padrão.

4.3.1 – Abordagens Baseadas no Nível MAC

Várias foram os trabalhos que trataram o problema descrito através de algum mecanismo que fosse inserido no nível MAC. Os trabalhos apresentados abaixo visando tratar os problemas do protocolo MAC IEEE 802.11 estão baseados em alguma mudança do algoritmo de *backoff*, da janela de contenção, criação de índices de prioridades justos ou implementação de algum mecanismo de QoS (*Quality of Service*) .

As abordagens precisam levar em consideração os dois tipos de rede local sem fio: as que são providas com estação base (AP) e as redes *ad hoc.* Em redes com estação base, o próprio AP atua com um ponto central arbitrando sobre demandas requeridas. Já em redes desprovidas de um nó central, as redes *ad hoc* a avaliação relativa a prioridade de um pacote fica mais complicada. Em redes *multihop* em que pacotes são encaminhados em cima de múltiplas regiões de *broadcast,* torna-se cada vez mais desafiante satisfazer fluxos QoS fim-a-fim [41]. Vejamos então algumas soluções:

A) Fair Medium Access in 802.11 Based Wireless Ad-Hoc Network [22]

Baseia-se na idéia da fila justa de redes com fio e define o índice de justiça para redes *ad hoc* para quantificar a justiça, de maneira que o objetivo de alcançar justiça torne-se equivalente a minimização do índice de injustiça. Para isso é proposto de um esquema diferente de *backoff* para o IEEE 802.11 DFWMAC ao invés do esquema original de *backoff* exponencial binário. Inicialmente é definido o índice de justiça que é baseado na razão entre máximo e mínimo *throughput* do *link* da seguinte forma:

$$\mathbf{FI} = \mathbf{Max}\{ (\forall\, i,j{:}\mathbf{Max}\ (W_i/\phi_i, W_j/\phi_j) / \mathbf{Min}\ (W_i/\phi_i, W_j/\phi_j)\},$$

Onde:

W_i = Atual *throughput* medido pela estação *i*;

ϕ_I = É uma constante pré-definida; (de acordo com cálculos chegou-se ao valor aproximado de 0,67 como sendo o ideal [22].

Após isso Cada estação calcula a probabilidade *p(ij)* de acessar o *link*. Isso pode ser feito de duas formas: a primeira é **baseada nas conexões** da própria estação e de sua vizinhança e a segunda é **baseada na média** dos períodos de tempo de disputa da estação e dos *links* individuais das outras estações, sendo definido:

W_{ei} = Estimativa de compartilhamento da própria estação;
W_{eo} = Estimativa de compartilhamento das outras estações;
T_{type} = Tempo de transmissão de um pacote do tipo *type.*

O algoritmo de estimativa de índice de justiça ficou assim:

Switch (received packet type) {
Case RTS:
If (DestID != localID) W_{eo} += T_{rts}
else { send **CTS** packet;
W_{eo} += (T_{rts} + T_{cts})}
Case CTS:
If (DestID != localID) W_{eo} += (T_{rts} + T_{cts})
else { send **DATA** packet;
W_{el} += (T_{rts} + T_{cts} + T_{data}) }
Case DATA:
If (DestID != localID)
W_{eo} += (T_{rts} + T_{cts} + T_{data})
else { send **ACK** packet;
W_{el} += (T_{rts} + T_{cts} + T_{data} + T_{ack}) }
Case ACK:
If (DestID != localID)
W_{eo} += (T_{rts} + T_{cts} + T_{data} + T_{ack})
else { W_{el} += (T_{rts} + T_{cts} + T_{data} + T_{ack}) }
}

É feita também, uma alteração no funcionamento da janela de contenção. Para isso é proposto um **índice de justiça estimado (FI_e),** definido como sendo:

$$FI_e = (W_{ei}/\phi_I) / (W_{eo}/\phi_o)$$

O novo algoritmo de controle da janela de contenção ficou assim:

Switch (FI_e) {
  case > C:
    $CW_{new} = \min (CW_{new} * 2, CWMax)$
  case (1/C,C):
    $CW_{new} = CW_{old}$
  Case < 1/C:
    $CW_{new} = \max (CW_{old} / 2, CWMin)$
}

No esquema proposto cada estação estimará o seu compartilhamento do canal e das outras estações e então ajustará a janela de disputa por conseguinte.

O esquema proposto não leva em consideração a topologia e a distribuição das estações na rede. De fato, a nova política de *backoff* do MAC através da utilização de índices de justiça onde cada estação ajusta a sua janela de contenção, é muito mais justa que o esquema original, entretanto este esquema sacrifica o *throughput.*

B) DFS - Distributed Fair Sheduling [27]

É um algoritmo de escalonamento distribuído justo chamado **DFS** (*Distributed Fair Scheduling*). A idéia é fazer com que alocação de canal para compartilhamento de pacotes seja feita de acordo com a importância dos fluxos. Ou seja, serão definidas várias filas de acordo com o tipo de fluxo e atribuído uma prioridade para cada fila. O compartilhamento ou solicitação para alocação de canal, será feita mediante a consulta nestas filas obedecendo a ordem de prioridade. Isto significa que pacotes

considerados mais importantes, terão maior prioridade que aqueles ditos com pouca ou nenhuma importância. O algoritmo está baseado em alterações no DCF original do MAC. O algoritmo é descrito de maneira geral assim:

3. Cada pacote transmitido é etiquetado com um *tag* de fim;
4. Quando no tempo *t* o nó *i* escutar ou transmitir um pacote com tag Z de fim, o nó *i* atribui ao seu *clock* virtual v_i o máximo de (v_i (t), Z);
5. As *tags* de inicio e fim para um pacote não são calculadas quando o pacote chega. Em vez disso, as *tags* dos pacotes são calculadas quando o pacote atinge o começo do fluxo;
6. Em seguida é escolhido o intervalo de *backoff* no qual o pacote com o menor *tag* de fim com melhores condições será associado ao menor intervalo de *backoff.*
7. Se ocorrerem colisões, então será escolhido um novo *backoff* da seguinte forma:
   a. o contador de *backoff* é incrementado em 1
   b. escolhe-se o novo *backoff* obedecendo:
      i. [$1,2^{\text{CollisionCounter}-1}$ x *CollisionWindow*], onde
      CollisionWindow é uma constante parametrizada.

O ambiente simulado é feito com *n* nós e *n/2* fluxos. O esquema proposto é bem mais justo que o esquema original do MAC IEEE 802.11. Entretanto este experimento não foi utilizado em redes *ad hoc multihop*, necessitando de aprimoramentos para esta topologia. Ele não resolve os problemas apresentados no capítulo 3.

C) Distributed Multi-Hop Sheduling na Medium Access with Delay and Throughput Constrains [41]

Aborda três assuntos fundamentais em escalonamento com qualidade de serviço em redes *ad hoc, s*endo eles: escalonamento por prioridades distribuído, acesso ao meio baseado em prioridade e gerenciamento de prioridade em ambiente *multihop*. Para isso desenvolve dois mecanismos de QoS para comunicação em

redes locais sem fio *multihop*. O primeiro é um **escalonamento por prioridade distribuído.** Através do monitoramento dos pacotes transmitidos, cada nó mantém uma tabela de escalonamento a qual é usada para estimar o nível de prioridades dos nós relativo aos outros nós. Este esquema foi baseado no esquema de prioridade de *backoff* do MAC IEEE 802.11 existente. O segundo mecanismo foi proposto baseado nos fatores de congestionamento, quebra de *link* e da natureza randômica de acesso ao meio que dificultam a exata realização de um escalonamento ideal. Baseado nisso, foi criado um esquema chamado **coordenação *multihop*** pois desta forma, nós *downstream* podem aumentar a prioridade relativa de um pacote para compensar o excessivo atraso ocorrido durante o *upstream.* Em seguida é proposto um modelo analítico simples que pudesse explorar quantitativamente estes mecanismos.

O esquema proposto altera a política de *backoff* original do MAC IEEE 802.11, criando um novo esquema de *backoff* distribuído. As simulações realizadas medem o atraso fim-a-fim, o *jitter* e o número de colisões ocorridas. O esquema proposto mostrou que o atraso fim-a-fim, *jitter* e número de colisões diminuíram em relação ao esquema original. Esta proposta não avaliou o *throughput*. Portanto não podemos garantir que este mecanismo resolve os problemas descritos no capítulo 3.

D) BTPS (*Busy Tone Priority Scheduling*) [28]

Apresenta um mecanismo de escalonamento com prioridades em redes *ad hoc multihop* chamado **BTPS** (*Busy Tone Priority Scheduling*). Este esquema baseia-se também na utilização de QoS. Segundo [28] existem dois padrões para redes em fio com cobertura para o nível MAC: o *European Telecomunication Standards Institute* (ETSI) *High Performance European Radio* LAN (HIPERLAN) [42] e o IEEE 802.11 WLAN [1]. O HIPERLAN suporta requisitos de QoS explicitamente em redes sem fio. O IEEE 802.11 pode transportar tráfego com requisito de tempo limitado usando PCF, onde necessita de um coordenador central. Entretanto nenhum dos dois pode oferecer efetivamente escalonamento com prioridade em redes *ad hoc.*

O mecanismo proposto usa o protocolo de duas faixas estreitas (*narrow-band*) com o BTPS, objetivando alcançar um escalonamento com prioridades. Ele assegura o acesso ao canal de pacotes de alta prioridade. Além disso, na ausência de pacotes de alta prioridade, fluxos de baixa prioridade podem fazer uso total da largura de banda disponível em BTPS. Os resultados da simulação demonstram a eficiência do protocolo BTPS em relação à entrega proporcional de pacotes de alta prioridade e alta agregação de *throughuput*.

E) Alteração do Algoritmo Black-Burst [21]

Propõe uma alteração no algoritmo *Black-Burst* (BB) [43], criando um esquema distribuído para o MAC para oferecer QoS em tempo real no acesso ao CSMA de redes *ad hoc* sem fio. Esquema é distribuído e baseia-se apenas em detecção de portadora. Ele concede acesso prioritário para tráfego de tempo real assegurando transmissões livres de colisões para este tipo de tráfego. Operando sobre uma rede *ad hoc* sem fio, ele garante ainda um atraso limitado para tráfego de tempo real. O BB é implementado a partir de uma modificação do CSMA original do padrão IEEE 802.11 apenas com pequenas modificações requeridas para tráfego de tempo real. O novo BB foi projetado para trabalhar de maneira geral em redes *ah doc* sem fio, desde que não ocorra terminal escondido, mas é o único no qual o canal pode ser reusado em toda a região de transmissão.

Esta proposta permite a utilização de tráfego em tempo real oferecendo QoS em redes *ah hoc multihop*. Entretanto uma condição necessária para o sucesso da solução é a não existência de terminais escondidos. Esta característica por ser indispensável para o algoritmo termina por não se aplicar em todos os caso, especialmente no cenário mostrado no capítulo 3 onde a presença de terminal escondido é muito provável. Além disso este trabalho não resolve os problemas descritos no capítulo 3.

4.3.2 - Abordagens Baseadas em Protocolos de Roteamento

As abordagens baseadas em protocolos de roteamento, concentram-se em níveis acima do MAC. De acordo com o problema descrito no capítulo 3, um dos protocolos que acusa tal problema é o TCP. Baseado no funcionamento do TCP especialmente o controle de conexões, muitas abordagens tentam trabalhar nesse nível tentando desta forma, não deixar a responsabilidade de detecção com o nível MAC. A idéia é atacar o problema evitando que ele ocorra. Veremos a seguir uma abordagem que trata este aspecto.

A) COPAS [44]

Considerando que as colisões na rede durante a disputa pelo acesso ao meio, degradam seriamente a performance do TCP, a idéia por traz dessa proposta é minimizar estas colisões oferecendo para isto dois caminhos para os pacotes TCP ACK e TCP DATA. Desta forma os dados seguiriam uma rota enquanto que reconhecimentos seguiriam por outra. Normalmente envio de dados e recepção de confirmação seguem o mesmo caminho que foi estabelecido no inicio da conexão. O algoritmo denominado **COPAS** foi criado a partir de alterações no protocolo de roteamento DSR original do padrão IEEE 802.11. O algoritmo garante a existência de uma rota X+Y+Z *forward* e outra Z+Y+X *reverse.* Desta forma uma será usada para pacotes TCP DATA e a outra para TCP ACK.

As simulações realizadas foram feitas com 1, 2 e 5 conexões TCP em cenários compostos de 50 e 100 nós. Para cada cenário foi verificado o *throughput*, *jitter* e o número médio de *backoff*. Foi comparada a performance do DSR original com a sua versão modificada, o COPAS. O resultado mostrou que com a modificação, o COPAS apresentou resultados superiores ao DSR original em todas as comparações, exceto no *jitter*, que no caso do COPAS foi ligeiramente maior. Esta mesma lógica poderia ser implementada utilizando também o protocolo de roteamento AODV.

Observe que é condição indispensável para que este algoritmo possa ser aplicado, uma topologia de rede que possibilite pelo menos duas alternativas de rotas para dados e reconhecimento. Entretanto no problema proposto no capítulo 3 onde temos nós alinhados, não oferecendo possibilidade de termos duas rotas, este algoritmo não pode ser usado, mas pode ser uma alternativa na busca de minimizar os problemas descritos.

B) Fair Sharing of MAC Under TCO in Wireless Ad Hoc Network [23]

Investiga a performance do TCP e MAC em redes *wireless multihop*. Foram usados vários protocolos de nível MAC dentre eles CSMA, FAMA (*Floor Acquisition Multiple Access*) [69] e 802.11. Foram estudadas várias topologias de rede, como nós alinhados, situações de terminal escondido, nós em anel e em formato de *grid*. Todas as topologias foram submetidas conexões com os protocolos CSMA, FAMA e 802.11, objetivando calcular o *throughput*, iniciando com o CSMA e finalmente progredindo tentando verificar a existência de colisões. A idéia central é criar um compartilhamento justo de acesso ao MAC com múltiplos fluxos TCP.

Os resultados mostram que tanto CSMA quanto FAMA são inadequados para redes *ad hoc multihop*, tanto na presença ou ausência de mobilidade. 802.11 é bem mais superior aos dois em termos de *throughput*. Entretanto, 802.11 mostra as evidencias dos efeitos de problemas de captura do canal. No caso de mobilidade CSMA e FAMA [69] entram em colapso, apenas 802.11 demonstra sinal de produtividade. O esquema proposto não resolve o problema de injustiça, já que ocorrem graves capturas de canal.

4.4 - Resumo das Soluções e Avaliações

Analisando as propostas acima, podemos verificar que de fato, algumas delas conseguiram melhorar o *throughput* do TCP e criar mecanismos mais justos de acesso ao meio, como por exemplo **[23]** e **[27]** além das propostas de escalonamento de fluxos por prioridade descrito em **[28]** onde encontramos a inserção de requisitos de QoS. Em **[44]** vimos que uma mudança no protocolo de roteamento causou a criação de duas rotas distintas, sendo uma para envio de pacotes de dados e outra para recepção de pacotes de reconhecimento. Esta alteração resultou em melhores taxas de *throughput.* Nenhuma dessas abordagens levou em consideração um cenário próximo ao descrito na Figura 3.9 A solução proposta sugere a criação de cenários mais justos, ou seja, cenários que proporcionem condições de disputa justas entre as estações que estão tentando acesso ao meio.

A avaliação feita indica que o problema descrito no capítulo 3 não foi resolvido ainda. Em quase todas as abordagens, de uma maneira ou de outra o problema foi amenizado. Entretanto nenhuma solução definitiva foi proposta, o que requer pesquisas mais aprofundadas.

Além desses trabalhos relacionados podemos encontrar outros em [45] [47] [48] [49] [50].

Capítulo 5

Conclusão, Contribuições e Trabalhos Futuros

5.1 – Conclusão

Nesta dissertação mostramos a tecnologia IEEE 802.11, com sua arquitetura e seus protocolos tanto de nível MAC quanto de nível de roteamento. Mostramos que o protocolo DFWMAC usado atualmente para simular redes 802.11 *ad hoc multihop* apresenta problema de **instabilidade** e **injustiça** no acesso ao meio. Estes problemas ficaram evidentes quando analisamos a performance do protocolo TCP nesse tipo de enlace. Nós vimos que colisões aliadas aos problemas de terminal escondido, terminal exposto e interferências entre as estações, podem gerar sérias capturas do meio beneficiando uma conexão e negando por completo o acesso a outra. O problema da instabilidade do TCP foi resolvido, apenas com uma mudança no parâmetro *window_* do TCP. Vimos ainda que, a medida que este número diminui, a instabilidade do TCP também diminui, estabilizando-se com tamanho igual a 4, no nosso caso. Já o problema de injustiça no acesso ao meio, aqui chamado de **injustiça de um salto,** não pode ser resolvido apenas com a mudança neste parâmetro. Neste segundo problema, a disputa pelo acesso ao meio torna-se injusta de maneira que uma determinada conexão não consegue mais acesso ao meio enquanto outra estiver ativa.

Através das simulações realizadas propomos uma alternativa que amenizasse o problema de injustiça criando topologias entre os nós de maneira que situações de terminal escondido e exposto não ocorressem. Para isso criamos condições para que a disputa pelo acesso não fosse injusta. Propomos topologias que não mantivessem as estações em um cenário totalmente alinhadas. Sabemos que esta proposta não resolve o problema de fato, uma vez que teríamos que prever todas as situações e que em razão da flexibilidade oferecida por uma rede local sem fio, poderia torna-se até inviável. Entretanto ela mostrou-se eficiente no aspecto de justiça entre as conexões, podendo ser aplicado em alguns casos.

Em todas as simulações realizadas onde verificamos tais problemas, ficou claro que a causa de todos eles está localizada sempre no mesmo lugar, o nível MAC.

Muitas pesquisas já foram feitas, como vimos no capítulo 4. Nenhuma delas apresenta uma solução definitiva para o problema. Uma outra componente que tende a agravar mais ainda o problema é a mobilidade, não tratada nesta dissertação.

Portanto, de acordo com a análise realizada e estudos de soluções existentes, concluímos que **o protocolo de nível MAC IEEE 802.11 não apresenta um bom desempenho em redes *ad hoc multihop.*** Uma solução eficaz e definitiva deve ser desenvolvida para atuar no nível MAC, onde o problema está localizado, estabelecendo critérios mais justos no acesso e compartilhamento do meio. Esta solução, acreditamos, que deva passar pela criação de mecanismos de prioridade, alteração da política do *backoff* e janela de contenção do MAC. Estas propostas juntas, devem culminar com requisitos de Qualidade de Serviço. Além disso, a solução deverá levar em consideração ajuste na faixa de interferência e sensibilidade.

5.2 - Contribuições

As contribuições geradas por este trabalho e que poderá auxiliar em outros trabalhos e pesquisas futuras são:

- **Levantamento do Estado da Arte:** Foi feito um levantamento da tecnologia IEEE 802.11, contemplando sua arquitetura, protocolos e serviços. Foi levantado o funcionamento dos níveis físico (PHY) e MAC definidos pelo padrão;

- **Apresentação do Problema:** O problema existente no MAC foi apresentado e mostrado as situações onde ele ocorre e como ocorre. Esta apresentação foi feita baseada em simulações que mostraram na prática a existência de tais problemas;

- **Solução Proposta:** Foi proposta uma solução que procurou amenizar o problema da injustiça no acesso ao meio, criando topologias de maneira a oferecer condições de disputa justas;

- **Análise de Soluções existentes:** Foi feito um levantamento e análise de soluções existentes que de alguma forma amenizam os problemas descritos;

- **Sugestão de Soluções:** Foi sugerido, baseado no problema e em experiências anteriores, formas de se atacar os problemas em busca de uma solução definitiva;

- **Sugestão de Trabalhos futuros:** Sugestão outros trabalhos que poderão surgir baseado neste como forma de dar continuidade na pesquisa do tema abordado.

5.3 - Trabalhos Futuros

Fica como sugestão de trabalhos futuros:

- Criação de um mecanismo de disputa justo no acesso ao meio levando-se em consideração problemas de terminal exposto e escondido;

- Utilização de requisitos de Qualidade de Serviço em busca de critérios de prioridades para cada fluxo e cada estação;

- Alterações na política de *backoff* do MAC IEEE 802.11, considerando problemas de terminal exposto e escondido;

- Modificação das faixas de interferência e sensibilidade das estações através do estudo das potências dos transmissores de maneira que estas faixas não interfiram outras estações.

- Estudo de tráfego com taxa constante como vídeo neste tipo de rede, para verificar a performance do TCP;

- Aprimoramento do protocolo MAC IEEE 802.11para trabalhar com redes *ad hoc multihop* móveis.

Capítulo 6

Bibliografia

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications:. IEEE Strandard 801.11, 1999.

[2] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: HighSpeed Physical Layer in the 5 Ghz Band IEEE 801.11a, 1999.

[3] J. Schindeler, *Mobile Comunication*. Addison-Wesley, 1st Ed., 2000 ISBN 0201398362.

[4] Rubinstein, Marcelo G. e Rezende, José Ferreira de, *"Qualidade de Serviço em Redes 802.11"*, 20º SBRC, Búzios/RJ, Maio 2002.

[5] Shugong Xu and T. Saadawi, *Does IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Network?*, in IEEE Communication, June 2001.

[6] Soares, Luiz Fernando Gomes et al: *Redes de Computadores: das Lans, Mans, Wans às Redes ATM*, Ed. Campus, 6ª edição, 1995.

[7] Tanenbaum, Andrew S.: *Redes de Computadores*. Ed. Campus, terceira edição, janeiro 2000.

[8] Prange, Cristian Ramos e Rochol, Juergen, *Redes Locais Sem Fio: Uma Análise Crítica*. UFRS.

[9] Bauchot, F.J. and Lanne, F. *IBM Wireless RF LAN Design and Architecture. IBM system Journal,* Armonk, EUA v34, n.3, pp. 390-408, Março 1995.

[10] Chayat, Naftali. *3 Mbit/s FH PHY Format Definition.* Documento IEEE p802.11-96/52, Março 1996.

[11] Chayat, Naftali. *Frequency Hopping Spread Spectrum PHY of the 802.11 Wireless LAN standard*, Março 1996.

[12] Trompower, Mike. *Direct Sequence Spread Spectrum Phisical Layer Specification for the 2.4 GHz ISM Band,* Março 1996.

[13] Hayes, Vic. *Tutorial on 802.11 to 802*, Março 1996.

[14] Haas, Z.J. Haas, S. Tabrizi, "*On Some Challenges and Design Choices in Ad-Hoc Communications*", MILCOM'98, Boston, MA, October 18-21, 1998

[15] S. Corson, J. Macker, "*Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*", Internet Draft, draft-ietf-manet-issues-02.txt, October 1998, expires in April 1999

[16] Patil, Abrieshek and Sahoo, Amit, *Routing Protocol For Ad-Hoc Wireless Network*, Novembro, 2001

[17] Bharghavan, Vaduvur et al. *MACAW: A Media Access Protocol For Wireless LAN's*, Agosto, 1994.

[18] Karn, P., *MACA – A New Channel Access Method For Packe Radio*, ARRL/CRRL Amateur Radio 9th Computer Networking Conference, Setembro, 1990.

[19] Biba, K., *A Hybrid Wireless MAC Protocol Supporting Asynchronous and Sunchronous MSDU Delivery Services*, IEEE 802.11 Working Group paper 802.11/91-92, Setembro, 1992.

[20] Fullmer, Chane L., and Garcia-Luna-Aceves, J.J, *Solutions to Hidden Terminal Problems inWireless Network*, Sigcomm, 1997

[21] Sobrinho, J. L. and Krishnakumar, A. S., *Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Network*, JSAC, 1999, vol 17, n. 8, pp, 1353-68.

[22] Bensasou, Brahim, and Wang, Yu, *Fair Medium Access in 802.11 Based Wireless Ad-Hoc Network*, Mobihoc 2000, agosto, 2000.

[23] Tang, K., and Gerla, M., Fair Sharing of MAC under TCP in Wireless Ad Hoc Network, IEEE MMT'99, outubro, 1999.

[24] Holland, G., and Vaidya, N., *Analisis of TCP performance in Wireless Multi-hop Network*, IEEE WMCSA'99, fevereiro, 1999.

[25] Wang, Yu, and Garcia-Luna-Aceves, J. J., *Colision Avoidance in Multi-Hop Ad Hoc Network*, IEEE/ACM MASCOTS 2002, dezembro, 2002

[26] Fang, Zuyan et all, *Performance Evalutation of Fair Backoff Algorithm for IEEE 802.11 DFWMAC*, IEEE/ACM/MOBHOC´02, Junho, 2002.

[27] Vaidya, Nitin H. et all, *Distributed Fair Scheduling in a Wireless LAN,* MOBICOM / ACM, agosto, 2000.

[28] Yang, Xue, and Vaidya, Nitin H., *Priority Scheduling in Wireless Ad Hoc Network,* MOBIHOC / ACM, junho, 2002.

[29] Wang, Yu and. Garcia-Luna-Aceves, J. J., *Performance of Collision Avoidance Protocols in Single-Channel ad Hoc Network*, novembro, 2002

[30] Jonson, D., et all, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), Internet Draft*, 2001.

[31] Wireless, http://www.networkdesigners.com.br/Artigos/wireless/wireless.html, acessada em 21/01/2003, às 15:35h.

[32] Rosa, Helio Fonseca, *Redes 802.11*, http://sites.uol.com.br/helyr/, acessada em 21/09/2002, às 15:45

[33] Padrão Para Redes Locais Sem Fio, http://www.dc.ufscar.br/~carvalho/WLAN/WLAN2.html, acessada em 21/01/2003 às 16:00h.

[34] Câmara, Daniel, *Roteamento em Redes Ad Hoc*, http://www.dcc.ufmg.br/~danielc/redes/roteamento.html, acessada em 22/01/2003 às 11:15h.

[35] Cunha, Daniel de Oliveira, *Roteamento em MANET's*, http://www.gta.ufrj.br/seminarios/semin2002_1/ Daniel/ , acessada em 22/01/2001 às 11:20h.

[36] Câmara, Daniel, *Princípios de Operação*. http://www.dcc.ufmg.br/~danielc/dissert_temp/node46.html#1072, acessada em 22/01/.2003 às 15:20h.

[37] The Network Simulation. http://www.isi.edu/nsnam/ns , accessada em 24/01/2003 às 14:10h.

[38] RFC 793, http://www.ietf.org/rfc/rfc0793.txt, setembro, 1981, acessada em 28/01/2003 às 09:09h.

[39] RFC 1323, http://www.ietf.org/rfc/rfc1323.txt, maio de 1992, acessada em 28/01/2003 às 09:11h.

[40] RFC 1332, http://deesse.univ-lemans.fr:8003/Connected/RFC/1332/8.html, acessada em 28/01/2003 às 09:58h.

[41] Kanodia, V. et all, Distributed Multi-Hop Scheduling an Medium Access with Delay and Trhroughput Constraints, MOBICOM/ACM, agosto, 2001

[42] ETSI TC-RES. Radio Equipament ad Systems (RES); High Performance Radio Local Area Network (HIPERLAN) type 1; Funcional Expecification. *European Telecomunication Standard ETS 300 652*, outubro, 1996.

[43] Sobrinho, J. L., and Krishnakumar, *A. S., Real-time Traffic over the IEEE 802.11 Medium Access Control Layer*, Bell Labs Tech J., vol 1 pp. 172-187

[44] Cordeiro, Carlos de M. et al, COPAS: Dynamic Contention-Balancing to Enhance the Performance of TCP over Multi-hop Wireless, outubro, 2002.

[45] Wang, Yu and Bensauo, Brahim, Priority Based Multiple Access for Service Differentiation in Wireless Ad Hoc Networks, maio, 2000.

[46] Wang, Yu and Garcia-Luna-Aceves, A new Hybrid Channel Access Scheme for Ad Hoc Networks, setembro, 2002.

[47] Balakrishnan, Hari et al, A Comparision of Mechanisms for Improving TCP Performance over Wireless Links, dezembro, 1997.

[48] Hang, Xiao Long and Bensaou, Brahim, On Max-min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analitical Framework and Implementation, outubro, 2001

[49] Vaidya, Nitin H. and Bahl, Paramvir, *Fair Scheduling in Broadcast Environments ,* agosto, 1999.

[50] Holland, Gavin and Vaidya, Nitin, Analysis of TCP Performance over Mobile Ad Hoc Networks, MOBICOM / ACM, agosto, 1999.

[51] Radiodifusão por Satélite, http://www.aminharadio.com/radio_satelite.html, acessada em 09/02/2003 às 11:08h.

[52] RFC 2108, http://www.ietf.org/rfc/rfc2108.txt, acessada em 09/02/2003 às 11:20h.

[53] Institute of Electrical and Electronics Engineers, http://www.ieee.org, acessada em 09/02/2003 às 11:30

[54] D. Clark, W. Fang, *Explicit Allocation of Best-Effort Packet Delivery Service*, IEEE/ACM Trans. on Networking, 6 (4), pp. 362-373, August 1998.

[55] IEEE Journal On, *Spread Spectrum Comunication II*, Vol 8, N. 05, juho, 1990

[56] IEEE Jornal Of Solid-States Circuits, vol. 37, N. 01, janeiro, 2002

[57] Institute of Electrical and Electronics Engineers http://grouper.ieee.org/groups/802/11/Tutorial/ds.pdf, acessada em 09/02/2003 às 16:20h.

[58] Inatel, http://grenoble.ime.usp.br/wcsf/slides/SessaoTecnica6/dqpsk.pdf, acessada em 09/03/2003 às 16:25h.

[59] Frequency Hopping Spread Spectrum vs. Direct Sequence Spread Spectrum, http://www.raylink.com/whitepaper/fhss_dsss.pdf, acessada em 09/02/2003 às 16:30.

[60] Sanchéz, Marvin et al, *CSMA/CA with Beam Forming Antennas in Multi-hop Packet Radio* http://www.s3.kth.se/radio/Publication/Pub2001/MarvinSanches2001_1presentation.pdf, acessada em 03/02/2003 às 16:45h.

[61] Nasipuri, Assis and Samir R., Multichannel CSMA with Signal Power-Based Channel Selection for Multihop Wireless Networks, setembro, 2000.

[62] RFC 2501, http://www.ietf.org/rfc/rfc2501.txt, acessada em 03/02/2003, às 17:05h

[63] IETF Home Page, http://www.ietf.org, acessada em 03/02/2003, às 17:08h

[64] Móbile Ad Hoc Networks (MANET), http://www.ietf.org/proceedings/98dec/43rd-ietf-98dec-91.html, acessada em 03/02/2003, às 17:17h.

[65] C. Cheng, R. Riley, S. Kumar, and J.J. Garcia-LunaAceves, *A Loop-Free Extended Bellman-Ford Routing Protocol without Bouncing E ect*. Computer Communications Review, Vol. 19(4):224-236, September 1989.

[66] RFC 0791, http://www.ietf.org/rfc/rfc0791.txt, acessada em 03/02/2003, às 17:59.

[67] Kamienski, C.A. & Sadok, D., *Chameleon: uma Arquitetura para Serviços Avançados Fim a Fim na Internet com QoS*, 19º SBRC, Florianópolis/SC, Maio 2001.

[68] X. Xiao and L. M. Ni. *Internet QoS: A big picture*. IEEE Network, 13 (2), March/April 1999.

[69] Fullmer, Chane L. and Garcia-Luna-Aceves, J. J., *Floor Acquisition Multiple Access (FAMA) for Packet-Radio Network,* MOBICOM, 1995.

[70] RFC 2001, http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2001.html acessada em 11/02/2003, às 13:51h

[71] WLANs com Infra-Estrutura, http://www.co.it.pt/seminarios/redes/semfios/sld005.htm, aceesada em 12/02/2003 às 10:28h

[72] WLANs Ad Hoc, http://www.co.it.pt/seminarios/redes/semfios/sld006.htm, acessada em 12/03/2003 às 10:30h.

[73] Posey, M. Brien, *Netgear 802.11a Cable/DSL High-Speed Wireless Router*, http://www.80211-planet.com/reviews/AP/article.php/1577961, acessada em 13/02/2003, às 13:57h.

Anexo A

Parte do *trace* da Simulação 1 com window_ = 8

```
s 28.839041039 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
D 28.839041706 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
s 28.840239039 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
D 28.840239706 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
s 28.841817039 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
D 28.841817706 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
r 28.842044372 _2_ MAC  --- 2689 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 2]
[897 0] 1 0
s 28.842054372 _2_ MAC  --- 0 MAC 38 [0 1 0 0]
r 28.842069372 _2_ RTR  --- 2689 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 2]
[897 0] 1 0
f 28.842069372 _2_ RTR  --- 2689 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 3]
[897 0] 1 0
r 28.842207039 _1_ MAC  --- 0 MAC 38 [0 1 0 0]
s 28.842543039 _3_ MAC  --- 0 MAC 44 [35f 2 3 0]
r 28.842719706 _2_ MAC  --- 0 MAC 44 [35f 2 3 0]
s 28.842729706 _2_ MAC  --- 0 MAC 38 [2bd 3 0 0]
r 28.842882372 _3_ MAC  --- 0 MAC 38 [2bd 3 0 0]
s 28.842932372 _3_ MAC  --- 2685 ack 132 [a2 2 3 800] ------- [5:0 1:0 32 2] [893
0] 2 0
r 28.843461039 _2_ MAC  --- 2685 ack 80 [a2 2 3 800] ------- [5:0 1:0 32 2] [893
0] 3 0
s 28.843471039 _2_ MAC  --- 0 MAC 38 [0 3 0 0]
r 28.843486039 _2_ RTR  --- 2685 ack 80 [a2 2 3 800] ------- [5:0 1:0 32 2] [893
0] 3 0
f 28.843486039 _2_ RTR  --- 2685 ack 80 [a2 2 3 800] ------- [5:0 1:0 32 1] [893
0] 3 0
r 28.843623706 _3_ MAC  --- 0 MAC 38 [0 3 0 0]
s 28.843693039 _2_ MAC  --- 0 MAC 44 [125e 3 2 0]
r 28.843869706 _3_ MAC  --- 0 MAC 44 [125e 3 2 0]
s 28.843879706 _3_ MAC  --- 0 MAC 38 [11bc 2 0 0]
r 28.844032372 _2_ MAC  --- 0 MAC 38 [11bc 2 0 0]
s 28.844082372 _2_ MAC  --- 2689 tcp 1092 [a2 3 2 800] ------- [1:0 5:0 32 3]
[897 0] 1 0
r 28.848451039 _3_ MAC  --- 2689 tcp 1040 [a2 3 2 800] ------- [1:0 5:0 32 3]
[897 0] 2 0
s 28.848461039 _3_ MAC  --- 0 MAC 38 [0 2 0 0]
r 28.848476039 _3_ RTR  --- 2689 tcp 1040 [a2 3 2 800] ------- [1:0 5:0 32 3]
[897 0] 2 0
f 28.848476039 _3_ RTR  --- 2689 tcp 1040 [a2 3 2 800] ------- [1:0 5:0 32 4]
[897 0] 2 0
r 28.848613706 _2_ MAC  --- 0 MAC 38 [0 2 0 0]
s 28.848763706 _2_ MAC  --- 0 MAC 44 [35f 1 2 0]
r 28.848940372 _1_ MAC  --- 0 MAC 44 [35f 1 2 0]
s 28.848950372 _1_ MAC  --- 0 MAC 38 [2bd 2 0 0]
r 28.849103039 _2_ MAC  --- 0 MAC 38 [2bd 2 0 0]
s 28.849153039 _2_ MAC  --- 2685 ack 132 [a2 1 2 800] ------- [5:0 1:0 32 1] [893
0] 3 0
r 28.849681706 _1_ MAC  --- 2685 ack 80 [a2 1 2 800] ------- [5:0 1:0 32 1] [893
0] 4 0
s 28.849691706 _1_ MAC  --- 0 MAC 38 [0 2 0 0]
r 28.849706706 _1_ RTR  --- 2685 ack 80 [a2 1 2 800] ------- [5:0 1:0 32 1] [893
0] 4 0
r 28.849706706 _1_ AGT  --- 2685 ack 80 [a2 1 2 800] ------- [5:0 1:0 32 1] [893
0] 4 0
s 28.849706706 _1_ AGT  --- 2691 tcp 1000 [0 0 0 0] ------- [1:0 5:0 32 0] [898
0] 0 0
```

```
r 28.849706706 _1_ RTR  --- 2691 tcp 1000 [0 0 0 0] ------- [1:0 5:0 32 0] [898
0] 0 0
s 28.849706706 _1_ AGT  --- 2692 tcp 1000 [0 0 0 0] ------- [1:0 5:0 32 0] [899
0] 0 0
r 28.849706706 _1_ RTR  --- 2692 tcp 1000 [0 0 0 0] ------- [1:0 5:0 32 0] [899
0] 0 0
s 28.849706706 _1_ RTR  --- 2691 tcp 1040 [0 0 0 0] ------- [1:0 5:0 32 2] [898
0] 0 0
s 28.849706706 _1_ RTR  --- 2692 tcp 1040 [0 0 0 0] ------- [1:0 5:0 32 2] [899
0] 0 0
r 28.849844372 _2_ MAC  --- 0 MAC 38 [0 2 0 0]
s 28.850213706 _1_ MAC  --- 0 MAC 44 [125e 2 1 0]
r 28.850390372 _2_ MAC  --- 0 MAC 44 [125e 2 1 0]
s 28.850400372 _2_ MAC  --- 0 MAC 38 [11bc 1 0 0]
r 28.850553039 _1_ MAC  --- 0 MAC 38 [11bc 1 0 0]
s 28.850603039 _1_ MAC  --- 2691 tcp 1092 [a2 2 1 800] ------- [1:0 5:0 32 2]
[898 0] 0 0
s 28.854851706 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
r 28.854971706 _2_ MAC  --- 2691 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 2]
[898 0] 1 0
s 28.854981706 _2_ MAC  --- 0 MAC 38 [0 1 0 0]
D 28.854982372 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
r 28.854996706 _2_ RTR  --- 2691 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 2]
[898 0] 1 0
f 28.854996706 _2_ RTR  --- 2691 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 3]
[898 0] 1 0
r 28.855134372 _1_ MAC  --- 0 MAC 38 [0 1 0 0]
s 28.855384372 _1_ MAC  --- 0 MAC 44 [125e 2 1 0]
r 28.855561039 _2_ MAC  --- 0 MAC 44 [125e 2 1 0]
s 28.855571039 _2_ MAC  --- 0 MAC 38 [11bc 1 0 0]
r 28.855723706 _1_ MAC  --- 0 MAC 38 [11bc 1 0 0]
s 28.855773706 _1_ MAC  --- 2692 tcp 1092 [a2 2 1 800] ------- [1:0 5:0 32 2]
[899 0] 0 0
s 28.857102372 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
D 28.857103039 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
s 28.858840372 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
D 28.858841039 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
s 28.859918372 _4_ MAC  --- 0 MAC 44 [35f 3 4 0]
D 28.859919039 _3_ MAC  COL 0 MAC 44 [35f 3 4 0]
r 28.860142372 _2_ MAC  --- 2692 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 2]
[899 0] 1 0
s 28.860152372 _2_ MAC  --- 0 MAC 38 [0 1 0 0]
r 28.860167372 _2_ RTR  --- 2692 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 2]
[899 0] 1 0
f 28.860167372 _2_ RTR  --- 2692 tcp 1040 [a2 2 1 800] ------- [1:0 5:0 32 3]
[899 0] 1 0
D 28.860266372 _4_ MAC  RET 0 MAC 44 [35f 3 4 0]
D 28.860266372 _4_ RTR NRTE 2690 ack 80 [a2 3 4 800] ------- [5:0 1:0 32 3] [896
0] 1 0
D 28.860266372 _4_ MAC  --- 2690 ack 80 [a2 3 4 800] ------- [5:0 1:0 32 3] [896
0] 1 0
```

Anexo B

Parte do *trace* da Simulação 2 com window_ = 1

```
s 30.064603792 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 30.064604459 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
s 30.065081792 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 30.065082459 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
s 30.067739792 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 30.067740459 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
r 30.068455792 _3_ MAC  --- 3047 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [4
0] 1 0
s 30.068465792 _3_ MAC  --- 0 MAC 38 [0 2 0 0]
r 30.068480792 _3_ RTR  --- 3047 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [4
0] 1 0
r 30.068480792 _3_ AGT  --- 3047 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [4
0] 1 0
s 30.068480792 _3_ AGT  --- 3048 ack 40 [0 0 0 0] ------- [3:0 2:0 32 0] [4 0] 0
0
r 30.068480792 _3_ RTR  --- 3048 ack 40 [0 0 0 0] ------- [3:0 2:0 32 0] [4 0] 0
0
s 30.068480792 _3_ RTR  --- 3048 ack 68 [0 0 0 0] ------- [3:0 2:0 32 2] [4 0] 0
0

s 30.070719125 _2_ MAC  --- 3049 tcp 1080 [a2 3 2 800] ------- [2:0 3:0 32 3] [5
0] 0 0
s 30.072227792 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 30.072228459 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
r 30.075039792 _3_ MAC  --- 3049 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [5
0] 1 0
s 30.075049792 _3_ MAC  --- 0 MAC 38 [0 2 0 0]
r 30.075064792 _3_ RTR  --- 3049 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [5
0] 1 0
r 30.075064792 _3_ AGT  --- 3049 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [5
0] 1 0
s 30.075064792 _3_ AGT  --- 3050 ack 40 [0 0 0 0] ------- [3:0 2:0 32 0] [5 0] 0
0
r 30.075064792 _3_ RTR  --- 3050 ack 40 [0 0 0 0] ------- [3:0 2:0 32 0] [5 0] 0
0
s 30.075064792 _3_ RTR  --- 3050 ack 68 [0 0 0 0] ------- [3:0 2:0 32 2] [5 0] 0
0
s 30.083627125 _2_ MAC  --- 3053 tcp 1080 [a2 3 2 800] ------- [2:0 3:0 32 3] [7
0] 0 0
s 30.085915792 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 30.085916459 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
r 30.087947792 _3_ MAC  --- 3053 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [7
0] 1 0
s 30.087957792 _3_ MAC  --- 0 MAC 38 [0 2 0 0]
r 30.087972792 _3_ RTR  --- 3053 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [7
0] 1 0
r 30.087972792 _3_ AGT  --- 3053 tcp 1028 [a2 3 2 800] ------- [2:0 3:0 32 3] [7
0] 1 0
s 30.087972792 _3_ AGT  --- 3054 ack 40 [0 0 0 0] ------- [3:0 2:0 32 0] [7 0] 0
0
r 30.087972792 _3_ RTR  --- 3054 ack 40 [0 0 0 0] ------- [3:0 2:0 32 0] [7 0] 0
0
s 30.087972792 _3_ RTR  --- 3054 ack 68 [0 0 0 0] ------- [3:0 2:0 32 2] [7 0] 0
0
s 30.090251125 _2_ MAC  --- 3055 tcp 1080 [a2 3 2 800] ------- [2:0 3:0 32 3] [8
0] 0 0
r 30.102700459 _3_ MAC  --- 0 MAC 38 [0 3 0 0]
s 30.103309792 _2_ MAC  --- 0 MAC 44 [122e 3 2 0]
```

```
r 30.103486459 _3_ MAC  --- 0 MAC 44 [122e 3 2 0]
s 30.103496459 _3_ MAC  --- 0 MAC 38 [118c 2 0 0]
r 30.103649125 _2_ MAC  --- 0 MAC 38 [118c 2 0 0]
s 30.103699125 _2_ MAC  --- 3059 tcp 1080 [a2 3 2 800] ------- [2:0 3:0 32 3] [10
0] 0 0
s 30.106247792 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 30.106248459 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
D 30.106595792 _5_ MAC  RET 0 MAC 44 [123e 4 5 0]
D 30.106595792 _5_ RTR NRTE 3044 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1516 0] 1 0
D 30.106595792 _5_ MAC  --- 3044 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1516 0] 1 0

.....
D 31.209081337 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
D 31.209252671 _5_ MAC  RET 0 MAC 44 [123e 4 5 0]
D 31.209252671 _5_ RTR NRTE 3384 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1517 0] 1 0
D 31.209252671 _5_ MAC  --- 3384 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1517 0] 1 0
s 31.209252671 _5_ RTR  --- 3402 DSR 36 [a2 4 5 800] ------- [5:255 6:255 255 6]
2 [0 0] [0 0 0 0->0] [1 1 6 5->4]

....
s 42.286178350 _5_ MAC  --- 0 MAC 44 [123e 4 5 0]
D 42.286179016 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
D 42.286526350 _5_ MAC  RET 0 MAC 44 [123e 4 5 0]
D 42.286526350 _5_ RTR NRTE 3686 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1517 0] 1 0

....
D 42.406964016 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
D 42.407311350 _5_ MAC  RET 0 MAC 44 [123e 4 5 0]
D 42.407311350 _5_ RTR NRTE 4084 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1517 0] 1 0
D 42.407311350 _5_ RTR NRTE 6699 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1517 0] 1 0
D 42.407311350 _5_ RTR NRTE 4922 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1517 0] 1 0

.....

D 80.157296201 _6_ RTR TOUT 9790 tcp 1020 [0 0 0 0] ------- [6:0 4:0 32 0] [1517
0] 0 0
D 99.401086744 _6_ RTR TOUT 16074 tcp 1020 [0 0 0 0] ------- [6:0 4:0 32 0] [1517
0] 0 0

.....

D 110.505130335 _4_ MAC  COL 0 MAC 44 [123e 4 5 0]
D 110.505477668 _5_ MAC  RET 0 MAC 44 [123e 4 5 0]
D 110.505477668 _5_ RTR NRTE 30078 tcp 1032 [a2 4 5 800] ------- [6:0 4:0 32 4]
[1522 0] 1 0
```