UNIVERSIDADE ESTADUAL DO MARANHÃO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO
CURSO DE MESTRADO PROFISSIONAL EM ENGENHARIA DE COMPUTAÇÃO E SISTEMAS

RAIMUNDO DE CARVALHO SILVA NETO

Dissertação de Mestrado

AVALIAÇÃO DE DESEMPENHO DO PROTOCOLO TCP EM REDES *WIRELESS* **LOCAL**: com e sem o uso de controladores.

São Luís
2013

RAIMUNDO DE CARVALHO SILVA NETO

AVALIAÇÃO DE DESEMPENHO DO PROTOCOLO TCP EM REDES *WIRELESS* **LOCAL**: com e sem o uso de controladores

Dissertação apresentada à Universidade Estadual do Maranhão, como parte dos requisitos para obtenção do Título de Mestre em Engenharia de Computação e Sistemas.

Orientador: Prof. Henrique Mariano Costa do Amaral

São Luís

2013

RAIMUNDO DE CARVALHO SILVA NETO

AVALIAÇÃO DE DESEMPENHO DO PROTOCOLO TCP EM REDES *WIRELESS*

LOCAL: com e sem o uso de controladores

Aprovado em: 14 / 08 / 2013

Prof. Henrique Mariano C. do Amaral (Orientador)
Coordenador do Curso de Mestrado Profissional em Engenharia de Computação e Sistemas

Prof. Fernando Cutrim Demétrio - UEMA

Prof. Luis Carlos C. Fonseca - UEMA

Prof. Sofiane Labidi - UFMA

AGRADECIMENTOS

Ao Senhor Deus do Universo: Onipotente, Onisciente e Onipresente, que ao longo de toda minha vida, tem demonstrado sua infinita misericórdia para comigo.

Ao meu orientador, o professor Henrique Mariano Costa do Amaral, pela atenção dispensada e neste convívio, a oportunidade de compartilhar um pouco de seus incontáveis conhecimentos.

Aos meus pais Sebastião e Maria, que souberam me educar no caminho da retidão e da verdade.

A minha esposa Cláudia, que sempre esteve ao meu lado, mesmo quando por força de compromissos profissionais e estudantis, não pude lhe dar a atenção que é merecedora.

Aos meus filhos Petra, Petrus e Saulo para que estes saibam que a educação é um processo contínuo e permanente ao longo de todas as nossas vidas.

Aos meus familiares e amigos de atividades profissionais pelo apoio e colaboração diárias que me são dispensados.

Quanto mais alta se torna uma construção, mais esta exigirá de suas fundações. Meus mais profundos agradecimentos àquela que sem dúvida, solidificou uma forte infraestrutura educacional em minha vida: dona Amélia Pinto Nogueira, avó e professora impar, a quem dedico grande reconhecimento e afeição.

RESUMO

As redes locais de computadores sem fio são extremamente suscetíveis à interferência e perda de potência dos rádios transmissores na área de cobertura. Há duas estruturas de redes locais sem fio: com uso de controladores ou não. Controladores de rede local sem fio são equipamentos proprietários das empresas de tecnologia, que gerenciam um grande quantitativo de rádios (pontos de acesso) simultaneamente, de maneira integrada e coordenada, presumidamente garantindo um melhor desempenho dos ativos das redes locais sem fio e consequentemente proporcionando tempos de transmissão dos segmentos TCP melhores que os apresentados por redes locais sem fio que não façam uso dos controladores. Para este trabalho, buscou-se por meio de experimento, efetuar um comparativo entre o desempenho de redes wireless com uso ou não de controladores. O uso de controladores em redes wireless é recomendado pela estabilidade maior do sinal Wi-Fi na célula, em virtude de recursos gerenciais dos sinais de transmissão, em relação às intempéries existentes no ambiente da propagação do sinal de rádio na frequência livre 2.4 GHz. Entretanto, o não uso de tais dispositivos, acaba por induzir os administradores de rede a servir a área de cobertura do sinal wireless, com a distribuição de AP's individuais que invariavelmente disputarão os hosts clientes, bem como poderão interferir nos sinais uns dos outros. Experimentalmente, foram obtidas amostras de transmissões de segmentos do protocolo TCP, para que estatisticamente fosse aferido qual o desempenho das transmissões em redes wireless com o uso de controladores e sem o uso destes em ambientes small-office. Os resultados foram avaliados utilizando-se ANOVA apoiada pelo Teste de Tukey.

Palavras-chave: Redes sem fio. AP. Controlador. ANOVA. Teste de Tukey.

ABSTRACT

Local networks of wireless computers are extremely susceptible to interference and loss of power radios within range. There are two structures of wireless local area networks: using controllers or not. Controllers Wireless LAN equipment owners are technology companies, managing a large quantity of radios (access points) simultaneously in an integrated and coordinated, presumably ensuring better performance of the assets of Wireless LANs and consequently providing time transmission of TCP segments better than those submitted by local wireless networks that do not use the controllers. For this study, we sought through experiment, making a comparison between the performance of wireless networks with or without use of controllers. The use of controllers in wireless networks is recommended by the increased stability of Wi-Fi signal in the cell, due to resource management of the transmitting signals in relation to the elements in the environment of propagation of the radio signal at the frequency 2.4 GHz free Meanwhile , the non-use of such devices, ultimately lead network administrators to serve the coverage area of the wireless signal, with the distribution of AP's individual invariably compete client hosts and can interfere with the signals from each other. Experimentally, samples were obtained broadcasts TCP segments, that were statistically assessed that the performance of transmissions in wireless networks with the use of controllers and without the use of these in small-office environments. The results were analyzed using ANOVA supported by the Tukey test.

Keywords: Wireless netorks. AP. Controller. ANOVA. Tukey Test.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledge
ACKc	Acknowledge confirm
ACKp	Acknowledge provisional
AP	Access Point
C++	Linguagem de programação multi-paradigma e de uso geral
CCNA	Cisco Certified Network Associate
chat	bate-papo
Cisco	Companhia multinacional sediada em San Jose, Califórnia, Estados Unidos da América
COFDM	Coded Orthogonal frequency-division multiplexing
CTS	Clear To Send
DNS	Domain Name System
DSSS	Direct Sequence Spread Spectrum
e-mail	Electronic Mail
FDM	Frequency-Division Multiplexing
FHSS	Frequency-hopping spread spectrum
FIN	Finish
FTP	File Transfer Protocol
GHz	giga-hertz - hertz equivale a ciclo por segundo
host	Qualquer máquina ou computador conectado a uma rede
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IEEE	Institute of Electrical and Electronics Engineers
IEEE 802.11	O padrão que estabelece normas para a criação e uso de redes sem fio
IEEE 802.11a	Padrão de rede sem fio que alcança 54 Mbps em 5.8 GHz
IEEE 802.11b	Padrão de rede sem fio que alcança 11 Mbps em 2.4 GHz
IEEE 802.11g	Padrão de rede sem fio que alcança 54 Mbps em 2.4 GHz
IM	Instant Messaging

IP	Internet Protocol
IRC	Internet Relay Chat
LAN	Local Area Network
MAC	Media Access Control
Mbps	megabits per second
MHz	mega-hertz - hertz equivale a ciclo por segundo
MSN	Microsoft Service Network
NAV	Network Allocation Vector
OFDM	Orthogonal frequency-division multiplexing
OSI	International Organization for Standardization
RADIUS	Remote Authentication Dial In User Service
RFC	Request for Comments
RIP	Routing Information Protocol
RTO	Recovery time objectve
RTO(w)	Recovery time objectve wait
RTS	Request To Send
RTT	Round Trip Time
RTT(w)	Round Trip Time Wait
SCCP	Skinny Call Control Protocol
SIP	Session Initiation Protocol
SMTP/POP	Simple Mail Transfer Protocol / Post Office Protocol
streaming	fluxo de mídia
SYN	Synchronize
TCP	Transmission Control Protocol
Telnet	TELecommunications NETwork
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
web	World Wide *Web*
WiFi	*Wireless* Fidelity

SUMÁRIO

1 INTRODUÇÃO

No Brasil, o índice de microempresas tende a se elevar, a partir da política econômica de baixa dos juros, incentivada pelo Governo Federal, bem como pela redução dos valores cobrados pelo consumo da energia elétrica, dentre outros incentivos. Com o crescimento do número de microempresas, a infraestrutura mínima de rede de computadores deverá prover as condições de competitividade dessas empresas.

As empresas e instituições públicas, de maneira geral, fazem uso de redes sem fio em suas instalações. Entretanto, essas redes sem fio, normalmente, são projetadas e montadas unicamente através da colocação de rádios (Pontos de Acesso) individuais como extensão da rede cabeada, sem que haja a avaliação mais aprofundada das condições técnicas para a instalação destes equipamentos.

Em termos gerais, as pessoas já fazem uso de algum acesso sem fio no âmbito de suas residências e creem que os mesmos aparelhos, que lhes provêm este acesso doméstico, serão capazes de administrar o tráfego de informações em nível empresarial. Os ativos de redes sem fio, que normalmente são usados por usuários residenciais, não possuem o nível crítico de estabilidade necessária para o uso em ambientes onde a manutenção ininterrupta dos serviços seja vital ao negócio da empresa.

Tais equipamentos são de baixo custo e possuem diminutos recursos de gerenciamento, a exemplo de *hardware* e *software* limitados destinados ao uso em pequenos ambientes tais como residências, *small-office*, etc.

Desta forma, justifica-se o presente projeto de dissertação com vistas a se avaliar o desempenho dos controladores em relação às redes *wireless* sem controladores, uma vez que tais equipamentos alardeiam maior disponibilidade, acessibilidade e mobilidade no ambiente das redes *wireless* LAN padrão IEEE 802.11b/g. Entretanto, não dispomos de um comparativo entre as duas estruturas de redes sem fio. Ou seja: com os recursos de gerenciamento oferecidos pelos equipamentos concentradores de pontos de acesso sem fio, entendem-se que eles devam prover maior ganho nas transmissões de segmentos TCP em redes sem fio, entretanto não se dispõe de quanto será a média de ganho que haverá ao fazer-se uso desses dispositivos. A necessidade de termos tal comparativo também irá abastecer os clientes finais na decisão de usar ou não controladores, uma vez que as redes sem fio montadas com controladores apresentam um custo financeiro de instalação bem mais elevado que a simples aquisição de pontos de acesso individuais.

1.1 Objetivo

O objetivo desta dissertação é realizar avaliação de desempenho de redes *wireless* local com controladores e sem, através das técnicas que possibilitem simular e medir. Para tanto foram desenvolvidos softwares em linguagem C++ que geraram carga de trabalho para amostras estatísticas.

Como produto final deste trabalho, obtivemos:

a) Dois programas (cliente/servidor) que transmitiram segmentos TCP, gerando amostras para conclusões estatísticas;
b) Verificação dos tempos de transmissão em *hosts*, apurando: as médias, desvios padrão, gráficos, realização de teste Anova e teste de Tukey com auxílio do software desenvolvido e outros.

1.2 Organização do trabalho

Esta pesquisa acadêmica está organizada em 5 (cinco) capítulos de acordo com o resumo que se segue.

No capítulo 2 trata-se da a camada de transporte do protocolo TCP, onde se revisam os principais conceitos sobre seu funcionamento e utilização. Este capítulo tem como finalidade subsidiar os leitores acerca da importância desta camada do modelo OSI, nas mais diversas redes de comunicação, bem como proporcionar os princípios que norteiam o comportamento técnico das conversações entre dispositivos ativos de rede e computadores durante o estabelecimento de conversações.

No capítulo 3 abordam-se as tecnologias de redes locais sem fio. Isto se deve ao fato da facilidade de aquisição, instalação e uso de equipamentos simples de pontos de acesso sem fio, nos mais diversos ambientes domésticos e empresariais, tornando-se uma tecnologia facilmente incorporada no dia-a-dia, mas que carece de uma compreensão aprofundada sobre seu funcionamento e sobre os problemas inerentes à própria tecnologia.

No capítulo 4 relatam-se os resultados das experiências realizadas na utilização do protocolo TCP sobre redes locais sem fio, usado-se pontos de acesso individuais e com o uso de controladores, com o objetivo de efetuar avaliação de desempenho para cada caso e confrontar-se os resultados em busca de uma média de ganho no tempo de transmissão na utilização de uma ou outra solução.

No capítulo 5, apresentam-se as conclusões finais do trabalho, juntamente com uma reflexão

sobre o grau de atendimento dos objetivos inicialmente propostos e os principais impactos dos resultados alcançados.

2 CAMADAS DE TRANSPORTE PROTOCOLO TCP

> Com a camada de rede, a camada de transporte é o núcleo da hierarquia de protocolos. A camada de rede oferece remessa de pacotes fim a fim usando datagramas ou circuitos virtuais. A camada de transporte se baseia na camada de rede para oferecer transporte de dados de um processo em uma máquina de origem a um processo em uma máquina de destino com um nível de confiabilidade desejado independente das redes físicas em uso no momento. (TANENBAUM, 2011, p.310).

Todas as estruturas de rede de computadores existentes, hoje em dia, oferecem suporte as variantes da rede de comunicação humana. Dados, voz, imagens e vídeos, onde quer que estejam os agentes de origem e destino da troca de informações, será necessário que o envio e recebimento dessas informações se deem de forma contínua e confiável, podendo os agentes estar na mesma localidade ou espalhados pelo mundo. Cada vez mais e mais dispositivos eletrônicos serão disponibilizados no mercado de consumo, onde invariavelmente tais equipamentos necessitarão de acesso à rede de computadores global para que possam realizar múltiplos serviços tais como: e-mail, *web*, mensagens instantâneas, etc.

Softwares aplicativos instalados nestes dispositivos fazem o papel de clientes de e-mail, *web brow*ser e permitindo ainda o envio e recepção de mensagens instantâneas. Para cada um dos dados gerados por essas aplicações, dar-se-á origem a segmentos, que no processo de encapsulamento culminarão nos pacotes IP; após a transmissão pela rede de computadores, urge que alcancem o servidor apropriado para tratar a solicitação enviada pelo usuário, quer seja resposta a seus e-mails, que seja página acessada, que sejam respostas às mensagens, etc.

Segundo Tanenbaum (2011, p.347), as informações contidas nos pacotes do protocolo IP não possuem nenhuma garantia de entrega ou recebimento por parte do destinatário dos pacotes, como também a velocidade de transmissão destes. É de responsabilidade do TCP o envio dos datagramas com velocidade suficiente para fazer uso da capacidade disponível sem, entretanto causar congestionamento. A função do IP é, pois, de realizar a melhor entrega possível, já a camada de transporte TCP é responsável pela transferência fim-a-fim dos dados que foram encapsulados pelas camadas superiores, fornecendo confiabilidade.

Além da confiabilidade, a camada de transporte TCP possui as seguintes funções:

a) Habilitar a comunicação de múltiplas aplicações na rede ao mesmo tempo em um único dispositivo;
b) Assegurar que, se necessário, todos os dados sejam recebidos confiavelmente e em ordem pela aplicação correta;

c) Empregar mecanismos de tratamento de erros.

2.1 Propósitos da camada de transporte

> O TCP é um protocolo full-duplex, orientado à conexão e altamente confiável. A arquitetura TCP é bastante complexa, o que acarreta um grande custo em termos de overhead. Como as redes de hoje são muito mais confiáveis do que as redes existentes quando o TCP foi criado, grande parte das características que garantem essa confiabilidade na transmissão poderia ser dispensada. (FILIPPETTI, 2008, p. 130).

A medida que os dados das camadas superiores são transmitidos da origem ao destino, a camada de transporte deverá prover a segmentação destes dados e o controle necessário para reagrupá-los no destino, pois, como a transmissão se dará ao longo de rotas diversas, os segmentos poderão chegar ao destinatário fora da ordem original de transmissão.

Os *hosts* conectados à rede possuem diversas aplicações disponíveis aos usuários. Tais aplicações poderão ser evocadas simultaneamente, cada uma acessando um serviço diverso: o usuário pode enviar um e-mail, enquanto abre uma página na Internet, troca mensagens via *chat* e faz uma chamada de voz via VoIP. Estas aplicações estão-se comunicando pela rede com outros *hosts* e servidores remotamente. Como as informações trafegam pela rede de forma não presumida, mister se faz que a camada de transporte mantenha e gerencie os múltiplos fluxos de comunicação entre as aplicações.

Quando iniciada a transmissão dos dados, o fluxo dos dados criado pelos diversos aplicativos existentes em um determinado *host* será dividido em pequenos fragmentos gerenciáveis chamados de segmentos. Os protocolos da camada de transporte descrevem todos os serviços que segmentam os dados oriundos da camada de aplicação.

A partir do momento em que o *host* de destino iniciar a recepção dos dados transmitidos, estes dados serão direcionados para a aplicação apropriada que dará o tratamento devido às solicitações do *host* de origem. Além disto, os segmentos de um aplicativo específico dificilmente chegarão na ordem em que foram enviados, pois os segmentos percorrerão caminhos diversos da origem ao destino. Desta forma, esses segmentos precisarão ser remontados na ordem em que foram originalmente transmitidos. A camada de transporte consegue fazer este processo de reagrupamento, utilizando as informações do cabeçalho da camada de transporte que foram criadas pelo *host* origem.

Tem-se com Comer (2006, p. 127) que, para que cada aplicação receba o fluxo de dados que lhe foi competente, a camada de aplicação deverá ter um mecanismo que possibilite identificar e separar o fluxo das diversas aplicações no *host* destino. Para tanto, no *host* de origem, quando da segmentação dos dados, será inserido um número identificador que

chamaremos de número da porta. Esta numeração da porta será inserida no cabeçalho da camada de transporte e estará associado a uma única aplicação a qual o segmento estará vinculado.

A camada de transporte se torna o elo de ligação entre a camada de aplicação e a camada mais inferior onde ocorre a transmissão efetiva dos dados em forma de bits. A camada de transporte promoverá diferentes trocas de dados e os submeterá às camadas inferiores, onde serão tratados como segmentos gerenciáveis que deverão ser transmitidos no meio físico a qual o *host* está conectado. As aplicações como e-mail, páginas *web*, transmissões VoIP, etc., não entrarão nos meandros da efetiva transmissão destes dados pela camada inferior. As aplicações gerarão o tráfego de dados que serão enviados da origem ao destino, mas que, no entanto, não se preocuparão com detalhes alheios a sua própria aplicação. Ou seja: não haverá relevância para as aplicações: a arquitetura do *host* de destino; se o meio de transmissão será par metálico, fibra ótica, meio sem fio, etc; qual a rota assumida pelos segmentos na transmissão; se os meios estão congestionados; ou qual a dimensão da rede. Por sua vez, as camadas inferiores no *host* origem, desconhecem as diversas aplicações que estão gerando os dados, pois sua função é estritamente acessar e transmitir estes dados pelo meio ao qual estão conectados e no *host* de destino, captar estes dados e elevá-los as camadas superiores para o tratamento devido.

A camada de transporte à medida que vai recebendo os dados oriundos das camadas inferiores vai organizando os segmentos e entregando a cada aplicação separadamente.

De maneira geral, a camada de transporte possui as competências anteriormente elencadas. Entretanto, as mais diversas aplicações existentes, necessitam de diferentes e particulares necessidades. Em virtude dessa realidade, estarão disponíveis do mercado múltiplos protocolos da camada de transporte.

Para algumas aplicações, os segmentos deverão chegar a uma sequência específica para serem processados com sucesso; é o caso das transmissões de voz sobre IP, onde estes dados são muito sensíveis a atrasos e à inversão na ordem de recepção dos segmentos.

Há casos, entretanto, que os dados precisarão ser recebidos por todos os *hosts* para poder ser usado; é o caso das transmissões em broadcast, que tem como fundamento a transmissão dos segmentos em uma rede local, para todos os *hosts* desta rede. Em outros casos ainda, uma aplicação poderá tolerar a perda de alguns segmentos por ocasião da transmissão; é o caso das videoconferências e teleconferências, onde mesmo ocorrendo a perda de alguns segmentos não impacta da transmissão geral da aplicação.

Antigamente tínhamos redes físicas diversas, uma para cada tipo de serviço: voz, vídeo, imagens e dados. No presente momento, com a realização da convergência das transmissões para um único meio físico, as aplicações com diferentes necessidades de transporte poderão comunicar-se na mesma rede. A diversidade de aplicações exige multiplicidade de protocolos que atuam na camada de transporte, pois estes exigirão regras diferentes para que os *hosts* de origem e destino possam lidar com essas singularidades. Por exemplo: no caso das transmissões de voz sobre IP, o protocolo da camada de transporte deverá ter apenas as funções básicas para efetuar a entrega dos segmentos da forma mais eficiente possível, pois a demora na entrega dos segmentos irá tornar a transmissão inaudível ou incompreensível e, portanto desprezível.

Outro caso que podemos citar é de uma transação bancária de crédito ou débito. Ela deverá ser cercada de todas as garantias possíveis de confiabilidade e inviolabilidade. Neste caso, precisamos de protocolos da camada de transporte que forneçam estas características adicionais para assegurar uma entrega confiável entre as aplicações. Com a agregação de tais recursos garantir-se-á uma comunicação robusta, entretanto, teremos sem dúvida uma sobrecarga a mais na rede, com consequente demanda por disponibilidade de acesso ao meio e largura de banda para transmissão.

Conforme já abordado sobre a separação de múltiplas comunicações simultâneas, deve-se salientar este importante conceito nas comunicações entre *hosts* fim-a-fim. Vamos usar como exemplo um equipamento de processamento de dados que esteja plugado a uma rede e através dela fazendo uso do envio e recebimento paralelo de e-mail, mensagens instantâneas, acessando *web*sites e fazendo uma ligação de voz via VoIP. Todas essas aplicações estarão solicitando ou respondendo a serviços em tempo compartilhado. Contudo, mesmo assim, não haverá interferência de uma transmissão em outra. As informações da chamada de voz não são recebidas pelo *web browser* e os dados do e-mail não serão visualizados pelos comunicadores instantâneos.

Precisamos também considerar que o conjunto de todos os dados transmitidos pela origem carece de ser recebido pelo destinatário integralmente para que estes dados passem alguma informação relevante. Este é o caso do envio dos e-mails e das páginas *web*. Atrasos medianos poderão ser tolerados para que estes dados em sua totalidade sejam recebidos, formatados e apresentados ao usuário final.

Entretanto, em alguns casos, perdas diminutas de parte dos dados poderão ser consideradas razoáveis, desde que não comprometam o montante das informações enviadas ou que possam ser reenviadas posteriormente. Como exemplo, podemos citar uma

conversação via VoIP onde a perda de parte da conversação poderá ser deduzida pelo usuário de destino, a partir do contexto maior da própria conversa ou ainda este usuário pode solicitar a retransmissão ao usuário de origem, sem gerencia ou controle dos protocolos da camada de transporte.

2.2 Controle das conversações

As principais funcionalidades presentes nos protocolos da camada de transporte incluem:

Segmentação e Reagrupamento – Imaginemos uma transmissão de videoconferência. Embora o fluxo de dados seja contínuo, este fluxo não monopolizará o meio de transmissão, pois caso o fizesse, não seria possível haver outros aplicativos executando suas funções concomitantemente. O uso de múltiplos aplicativos será possível, graças à segmentação dos dados que irão fluir da origem ao destino. Ora, se houver uma segmentação dos dados na origem, concluímos que deverá haver um reagrupamento destes dados no destino.

Multiplexação de Conversação – A ocorrência da segmentação possibilitará que um aplicativo não detenha o meio de transmissão unicamente para seu uso privado. A execução simultânea de vários aplicativos, a qual exija o uso da rede, fará gerar inúmeros segmentos dos mais variados aplicativos desembocando no meio de transmissão. Para gerar a diferenciação de quais aplicativos estão sendo requisitados, será incluído no cabeçalho da camada de transporte o número da porta associada ao serviço requisitado. A possibilidade de que um *host* de origem transmita diversas requisições simultaneamente pela rede, dá-se o nome de multiplexação da conversão.

Conversações orientadas à conexão – Para que a camada de transporte possa efetuar conversações orientadas a conexões, serão estabelecidas previamente sessões entre as aplicações. A finalidade dessas sessões criadas entre as aplicações será a gerência dos dados que estão sendo trocados entre os aplicativos.

Entrega Confiável – A efetiva transmissão dos segmentos pela rede poderá sofrer a corrupção de alguns segmentos ou mesmo a perda total destes. Isto se deve à custa de inúmeros fatores, como, por exemplo, interferência eletromagnética nos cabos metálicos usados como meios físicos de transmissão que corromperam o fluxo de bits. A camada de transporte deverá assegurar que todos os segmentos cheguem ao seu destino, através da

retransmissão dos segmentos pela origem, quando o destino não confirmar o recebimento destes.

Entrega na Mesma Ordem – A geração de tantos segmentos quantos os necessários para a transmissão da informação da origem ao destino, bem como o envio destes segmentos por todas as rotas alternativas que estejam ativas conectando os dois pontos da comunicação, muito corriqueiramente farão com que os segmentos, embora transmitidos sequencialmente, possam chegar ao destino de maneira desordenada e fora da sequência. A camada de transporte deverá receber estes segmentos desordenados, pô-los em sequência a assegurar a entrega à aplicação na ordem correta.

Controle de Fluxo – Qual a capacidade de processamento dos *hosts* de origem e destino? Quais softwares estão instalados em cada um deles. A quantos *hosts* origem, um *host* de destino está atendendo simultaneamente? Estas são perguntas que sinalizam que os *hosts* envolvidos na comunicação possuem limitação de recursos para atender as solicitações que lhes são feitas. Recursos como capacidade de processamento, memória e largura de banda poderão fazer com que um *host* origem com maior disponibilidade destes recursos acabe sobrecarregando um *host* de destino com menor capacidade. Esta sobrecarga no *host* de destino poderá ocasionar a perda de segmentos, o que resultaria na retransmissão destes segmentos pelo *host* de origem ampliando a sobrecarga no *host* de destino. Para os protocolos da camada de transporte, a melhor alternativa em situações como esta, é ao perceber esta sobrecarga no *host* de destino, informem e solicitem ao *host* de origem que inicie o processo de redução da taxa de fluxo de dados.

2.3 Suportes de comunicação confiável

Em termos de transmissões em redes de comunicação, oferecer confiabilidade ao tráfego, será garantir que o conjunto de segmentos, que compõe a informação a ser transmitida, chegue ao destino de tal maneira que representem a informação original. Os protocolos da camada de transporte poderão desenvolver um mecanismo para viabilizar a entrega de forma confiável, sendo que as três operações básicas de confiabilidade são: rastreamento de dados transmitidos, confirmação de dados recebidos pelo destino, retransmissão pela origem de quaisquer dados não confirmados.

Para que os protocolos da camada de transporte possam oferecer suporte de comunicação confiável às conversações na rede, haverá a necessidade de que a origem mantenha um sequenciamento e rastreamento contínuo dos segmentos transmitidos. Esse

rastreamento irá identificar os segmentos transmitidos, mas que não foram confirmados o recebimento pelo destino. Nesses casos, o protocolo da camada de transporte no *host* de origem, providenciará o reenvio de todos os segmentos não confirmados.

Saliente-se que o processo de confiabilidade, aqui relatado, em virtude da necessidade de o protocolo da camada de transporte ter que realizar confirmação, rastreamento e retransmissão, vai elevar uma demanda maior pelos recursos da rede, pois um maior quantitativo de dados de controle será enviado entre o *host* de origem e o de destino.

Verifica-se que há uma dicotomia existente entre o índice do que pode ser considerado confiável no fluxo dos dados e a necessidade de mais carga na rede a fim de garantir esta confiabilidade.

Tendo como base as necessidades das aplicações que serão executadas nos *hosts*, as empresas, que desenvolvem os softwares que farão uso dos recursos da rede, deverão ter em mente que tipo de protocolo da camada de transporte será mais apropriado para suas aplicações, pois na camada de transporte existem protocolos cujos métodos fornecerão entrega confiável ou entrega de melhor esforço possível. Os protocolos cuja entrega for garantida pautam-se nos processos de confirmação, rastreamento e retransmissão dos dados. Já os protocolos de melhor esforço possível não trabalham com esses processos, motivo pelo qual não há como garantir que os segmentos transmitidos a partir da origem serão recebidos pelo destino.

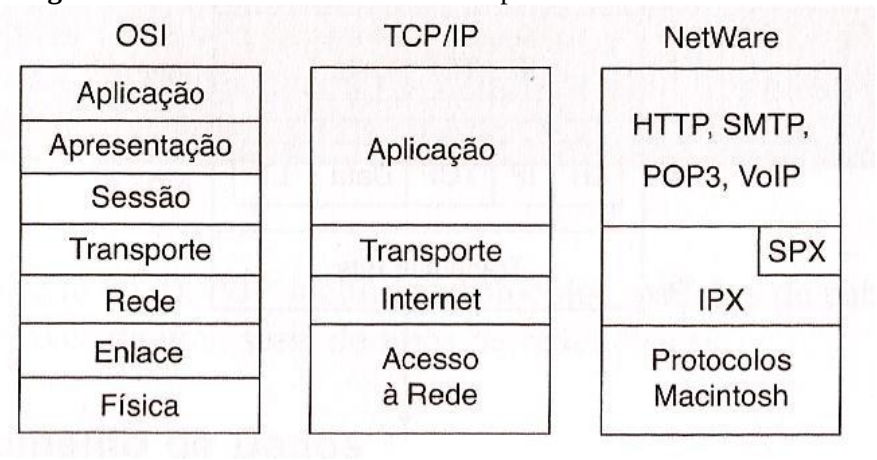
A definição se as aplicações necessitam de confiabilidade ou não vai depender dos objetivos e forma de trabalhar de cada uma destas aplicações. Em muitas delas, para que os dados transmitidos tenham relevância, estes dados precisam chegar até o destino perfeitamente igual ao dado que foi postado na rede e na ordem em que foram segmentados. São os casos das bases de dados, páginas *web* e e-mails, pois quaisquer perdas de dados, invariavelmente ocasionarão informações truncadas que poderão se tornar incompreensíveis. Desta forma, tais aplicações são desenvolvidas permeadas no uso de um protocolo da camada de transporte que forneça confiabilidade nas transmissões. Nestes casos o inconveniente da sobrecarga proveniente dos processos de confirmação, rastreamento e retransmissão a fim de gerar a confiabilidade, é caracterizado como aceitável em função da relação custo x benefício oferecido a tais aplicativos.

Entretanto, para alguns aplicativos, aceitar algumas perdas poderá ser um pressuposto de sua utilização sem que lhe inviabilize o uso. Tais aplicativos são mais tolerantes à perda de um quantitativo razoável dos dados que foram transmitidos. Tomemos como base um aplicativo de *web* conferência. Caso alguns segmentos transmitidos não

alcançarem por qualquer motivo o seu destino, o impacto causado poderá ser de uma breve interrupção, a distorção da imagem ou mesmo nem ser notado pelo usuário final.

Nesse caso, se quisermos garantir a entrega deste quantitativo ínfimo de segmentos perdidos, estaremos correndo o risco de inviabilizar o uso do aplicativo de *web* conferência, pois a identificação da perda de tais segmentos e a sua retransmissão pela origem estariam causando um retardo na execução das imagens sucessivas no destino. Colocando na balança, chega-se à conclusão, para o caso em questão, de que, havendo uma perda de dados aceitável, será melhor exibir as imagens que conseguirão chegar através dos segmentos transmitidos e não invocar a confiabilidade, para garantir a transmissão e deixar ao usuário a missão de interpretar a mensagem geral da *web* conferência, mesmo havendo algumas perdas pontuais na transmissão. Na figura 1, temos um comparativo entre os principais modelos de pilhas de protocolo onde verificamos a relação entre as camadas de transporte.

Figura 1: Protocolos da camada de transporte



Fonte: Odom (2008, p. 24)

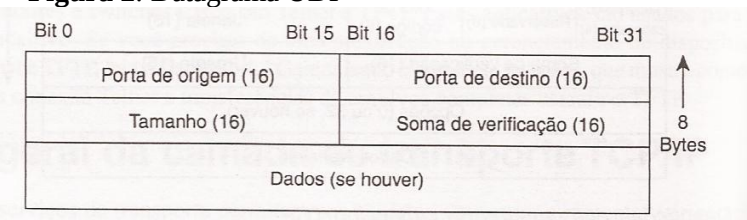
2.4 Protocolos TCP e UDP

Existem diversos protocolos que atuam na camada de transporte, entretanto os mais utilizados na atualidade são os que pertencem à pilha de protocolos TCP/IP, com destaque a dois protocolos desta pilha. São eles o TCP e o UDP. Estes dois protocolos têm como característica o gerenciamento de múltiplas aplicações simultaneamente e se diferenciam em suas funções à aplicabilidade de cada um deles.

O protocolo UDP não é orientado à conexão. Este é um protocolo simples que tem como grande diferencial a possibilidade de efetuar a transmissão e entrega dos segmentos, também conhecidos como datagramas, com baixa sobrecarga. Para obter essas

funcionalidades o protocolo UDP eliminará a confiabilidade na entrega dos datagramas, tornando-se então um protocolo da camada de transporte de melhor esforço possível. Como aplicações, que fazem uso do protocolo UDP, podemos citar DNS, Vídeo em Streaming, Voz Sobre IP (VOIP). O protocolo UDP é descrito na RFC 768. Na figura 2 temos os campos do datagrama UDP.

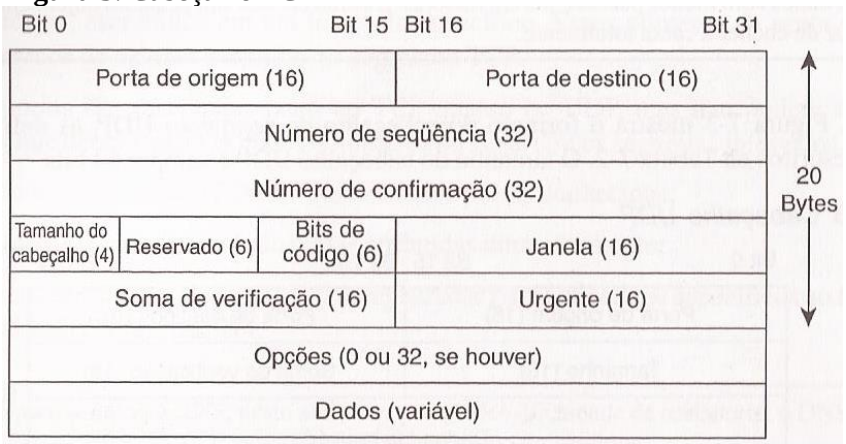
Figura 2: Datagrama UDP



Fonte: Mcquery (2002, p. 182).

O protocolo TCP é orientado à conexão. Para obter esta característica, o protocolo TCP gerará sobrecarga natural, objetivando prover as seguintes funções: entrega ordenada, entrega confiável e controle de fluxo. Como aplicações que fazem uso do protocolo TCP podemos citar navegadores *web*, e-mail, FTP. O protocolo TCP é descrito na RFC 793. Na figura 3 são expressos os campos do segmento TCP.

Figura 3: Cabeçalho TCP



Fonte: Mcquery (2002, p. 181)

2.5 Endereçamentos de porta

Os protocolos da camada de transporte quer estejam trabalhando com confiabilidade de entrega dos segmentos ou não, como o próprio nome desta camada já faz referência, eles têm como função primordial efetuar o transporte dos segmentos do *host* origem ao *host* de destino. Urge lembrar também que um *host* cliente poderá realizar inúmeras

solicitações a um *host* servidor ou a vários *hosts* servidores. Assim sendo, um *host* de origem possui a capacidade de ao mesmo tempo receber e enviar mensagens de e-mail, acessar várias páginas *web* e efetuar chamadas VoIP.

Para que isto ocorra sem que os dados de um aplicativo de e-mail sejam entregues a um aplicativo de página *web*, os protocolos TCP e UDP precisarão monitorar todas as aplicações com os quais estão fazendo uso dos recursos da rede. Este monitoramento, entre outras atribuições, consistirá em identificar os segmentos TCP, ou os datagramas UDP, para cada aplicativo por meio de um identificador no cabeçalho TCP e UDP chamado de número da porta.

No cabeçalho destes dois protocolos da camada de transporte, encontraremos duas portas: a porta de origem e a porta de destino dos aplicativos. Enquanto a porta de origem se referir ao número de porta relacionada com a aplicação cliente que efetuou a requisição dos serviços, a porta de destino estará associada ao número de porta da aplicação servidora que deverá tratar a requisição do cliente e lhe responder com os dados solicitados.

Como observado nas figuras 2 e 3, os protocolos UDP e TCP possuem datagramas e segmentos únicos e indistintos, quer sejam uma solicitação do cliente quer uma resposta do servidor. Também para este caso, o discernimento se for uma solicitação ou resposta, dar-se-á pelos números de portas. Temos diversas formas de atribuir os números de portas que serão influenciados caso sejam solicitações ou respostas. De maneira geral, os processos e serviços que são executados no *host* servidor, possuem número de porta padronizado e fixo, designados a eles pelo profissional desenvolvedor do aplicativo. Já os clientes escolhem de forma dinâmica o número de porta para cada solicitação ao servidor.

No momento em que uma aplicação cliente fizer uma requisição a um servidor, através do envio de uma solicitação do *host* origem ao host de destino, a porta de destino a ser preenchida no datagrama ou no segmento será a porta que foi designada no servidor para atender as requisições de uma determinada aplicação. Do ponto de vista do *host* cliente, este deve ter conhecimento de todos os números de portas, para os aplicativos disponíveis nos servidores. Neste caso o melhor a ser feito será padronizar as portas dos principais serviços que se disponibilize nas redes. Desta forma, de antemão os desenvolvedores de aplicativos servidores já sabem as portas padronizadas para e-mail, páginas *web*, mensagens instantâneas, etc.

Como exemplo, tomemos uma aplicação de navegador *web*. Ao se solicitar o acesso a partir de um *host* cliente, o aplicativo de navegação *web* usará o protocolo TCP cuja porta de destino nos segmentos será a porta 80, a não ser que este servidor de página *web*

tenha sido programado especificamente para usar outra porta. A porta 80 é utilizada por servidores *web*, para disponibilizar páginas em todo o mundo. Foi o padrão adotado, assim como os demais principais serviços na rede. A figura 4 mostra mais alguns exemplos de portas reservadas segundo Comer (2006, p.148).

Figura 4: Endereçamento de Porta

Decimal	Palavra-chave	Descrição
0		Reservado
7	ECHO	Eco
9	DISCARD	Descartar
11	USERS	Usuários ativos
13	DAYTIME	Hora do dia
15	netstat	Programa de status da rede
17	QUOTE	Citação do dia
19	CHARGEN	Gerador de caracteres
20	FTP-DATA	File Transfer Protocol (dados)
21	FTP	File Transfer Protocol
22	SSH	Secure Shell
23	TELNET	Conexão de terminal
25	SMTP	Simple Mail Transport Protocol
37	TIME	Hora
53	DOMAIN	Servidor de nome de domínio
67	BOOTPS	Servidor BOOTP ou DHCP
79	FINGER	Finger
80	WWW	World Wide Web server
88	KERBEROS	Serviço de segurança Kerberos
110	POP3	Post Office Protocol versão 3
111	SUNRPC	SUN Remote Procedure Call
119	NNTP	USENET News Transfer Protocol
123	NTP	Network Time Protocol
143	IMAP	Internet Message Access Protocol
161	SNMP	Simple Network Management Protocol
443	HTTPS	HTTP seguro

Fonte: Comer (2006, p. 149)

Já o preenchimento no número da porta cliente, no datagrama UDP ou no segmento TCP, será gerado de forma aleatória no momento da requisição dos serviços e este número a ser utilizado, não poderá estar em uso no *host* local, a fim de evitar que haja conflitos com portas em uso por outros aplicativos.

Em uma solicitação do cliente ao servidor, o número da porta cliente escolhido aleatoriamente será preenchido no datagrama ou segmento como porta de origem. A porta de destino, como já discorrido, será preenchida com o número do serviço a ser requisitado do host de destino. O *host* cliente faz acesso ao meio de transmissão e efetua o envio dos dados. Ao chegar ao *host* de destino, o datagrama ou segmento será capturado e tratado pelo aplicativo servidor associado a porta de destino no cabeçalho. O aplicativo servidor providenciará o processamento das informações requisitadas e precisa reenviar ao *host* cliente à resposta da solicitação recebida. Para garantir que o host solicitante receba as informações que o servidor irá enviar nas portas corretas, o servidor irá inverter as portas recebidas. O que era o número da porta de origem passará a ser o número da porta de destino e o que era o número da porta de destino, passará a ser o número da porta de origem.

Entretanto, somente os números de porta de origem e destino serão insuficientes para um endereçamento que garanta a entrega adequada dos dados. Há a necessidade da combinação do número de porta UDP ou TCP da camada de transporte, com o endereço IP da camada de rede; com isto, sim, poderemos garantir a entrega dos dados. À combinação do endereço da porta UDP ou TCP e o endereço IP damos o nome de soquete. O conjunto de dois soquetes, que compreende os endereços de IP e porta de origem e destino dos *hosts*, será único e identificará a conversa entre dois *hosts* de forma exclusiva.

Para Comer (2006, p.127), como um exemplo de soquete poderá ser citado um *host* de origem com endereço IP 128.10.2.3 que deseja acessar via *web browser* um *host* de destino, com o serviço http habilitado na porta padrão possuindo endereço IP 18.26.0.36. Como exemplo de socket de origem, seria 18.26.0.36:1069, que é formado pela justaposição do endereço IP 18.26.0.36 e uma porta TCP escolhia aleatoriamente de número 1069. E como socket de destino 128.10.2.3:80, que é formado pela concatenação do endereço IP de destino 128.10.2.3 e a porta padrão TCP para serviços http de número 80.

O controle e designação acerca da numeração das portas dos protocolos da camada de transporte, bem como da organização e padronização de endereçamento serão realizados através da IANA. Para o conjunto de protocolos TCP e UDP, existem três tipos de números de porta: portas conhecidas, portas registradas e dinâmicas ou privadas. A tabela 1 faz referência a tais portas.

Tabela 1: Portas TCP e UDP: conhecidas, registradas e dinâmicas

Protocolo	Números de Portas	Grupo de Portas	Exemplos
TCP	0 a 1023	Conhecidas	21 – FTP 23 – Telnet 25 – SMTP 80 – HTTP 110 – POP3 194 – IRC 443 – HTTPS
	1024 a 49151	Registradas	1863 – MSN Messenger 8008 – Alternar HTTP 8080 – Alternar HTTP
	49152 a 65535	Dinâmicas ou Privadas	-
UDP	0 a 1023	Conhecidas	69 – TFTP 520 - RIP
	1024	Registradas	1812 – RADIUS 2000 – SCCP (VoIP) 5004 – RTP (Voz e Vídeo) 5060 – SIP (Voz)
	49152 a 65535	Dinâmicas ou Privadas	-

Fonte: Elaborado pelo autor (2013).

As portas conhecidas têm uma numeração de 0 a 1023, onde este intervalo de numeração está reservado para serviços e aplicações tais como HTTP, POP3/SMTP, Telnet, etc. Normalmente, tais serviços vão estar disponibilizados para uma gama de usuários da rede, desta forma, o intervalo de numeração de 0 a 1023 é aplicado em *hosts* com função de servir a rede com os serviços acima descritos, tornando-se os servidores da rede. Uma vez que as portas são conhecidas será fácil desenvolver aplicativos que sejam executados nos *hosts* clientes e que façam referência a estas portas no cabeçalho TCP ou UDP no campo de portas de destino.

As portas registradas possuem numeração de 1024 a 49151, onde este intervalo de numeração estará designado para processos ou aplicações de usuário. Ou seja: aplicativos desenvolvidos por programadores para satisfazer necessidades específicas de um grupo de usuários recebem números de portas registradas em vez de portas conhecidas.

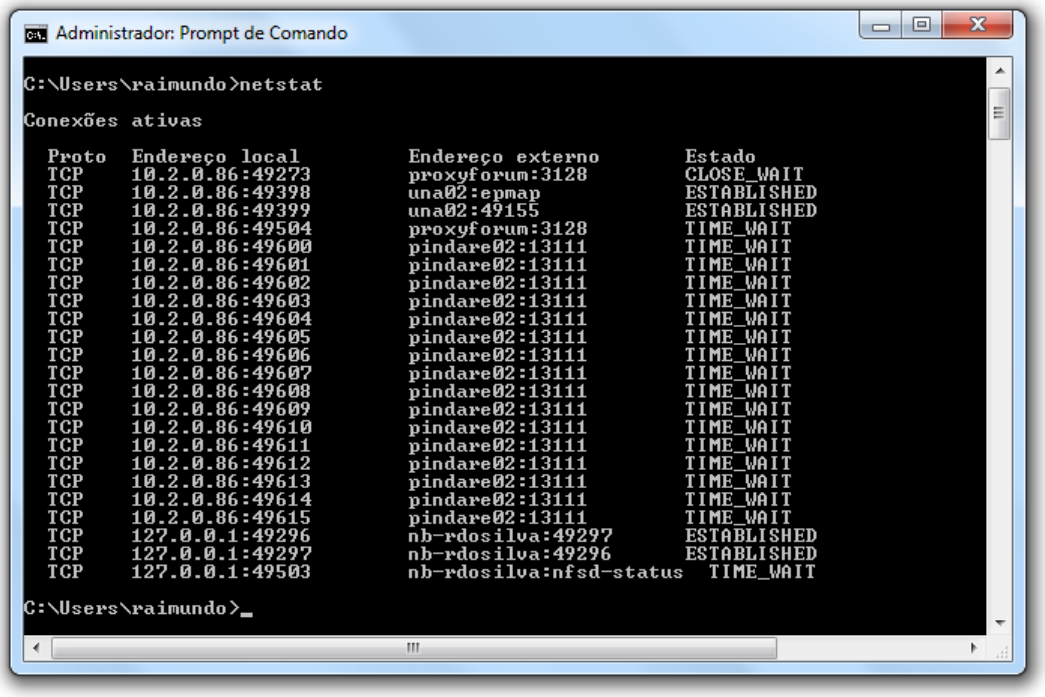
As portas dinâmicas, também conhecidas como portas privadas, recebem numeração que vai de 49152 até 65535. Quando um *host* cliente faz conexão com um *host* servidor para ter acesso ao serviço tipo HTTP, a porta de destino a ser utilizada no servidor será a porta de número 80. A porta de origem que o cliente utilizar nesses casos será uma porta dinâmica. Esta é a regra geral, ou seja, para conexões de cliente ao um servidor, a porta de origem será alocada de forma dinâmica e no servidor previamente conhecida.

Um utilitário de rede que possibilita evidenciar as conexões abertas em um *host* local é o NETSTAT. O nome deste utilitário é uma abreviação de Network Statistic, ou seja, estatística de rede, sendo comum aos sistemas operacionais Windows, Unix e Linux. Além de verificar as conexões abertas, este utilitário tem como finalidade obter informações acerca das conexões de rede, tabelas de roteamento, bem como outras informações sobre a utilização da interface na rede. O NETSTAT exerce suas funcionalidades tanto para o protocolo TCP, quanto para o protocolo UDP.

Como já foi relatado, com a utilização do NETSTAT podemos visualizar todas as conexões que estão em andamento naquele instante na máquina investigada. Informações como o protocolo em uso, o endereço local e o número de porta, o endereço de destino, o número de porta e o estado da conexão, poderão ser observados com o uso do NETSTAT.

Na figura 5, temos a tela de saída do comando NETSTAT, quando executado em um host aleatório no ambiente de uma rede local de computadores.

Figura 5: Tela de saída do utilitário NetStat



Fonte: Elaborada pelo o autor (2013)

2.6 Segmentação, reagrupamento e confiabilidade

Dependendo da aplicação a fazer uso dos recursos da rede, uma expressiva quantidade de informações poderá ser necessária a sua transmição como consequência da conexão existente entre o *host* de origem e o *host* de destino. Se a transmissão fosse efetuada por um único segmento extremamente grande, a possibilidade de que esta transmissão resultasse em insucesso seria muito elevada. Esta afirmativa se baseia no fato de que ativos de redes possuem restrição de memória para armazenar um segmento extremamente grande, enquanto o mesmo estivesse sendo transmitido por entre os ativos de rede, além do que um erro na transmissão que resultasse na corrupção deste único segmento culminaria na necessidade de sua retransmissão. Outros problemas advindos deste único segmento seriam a impossibilidade de enviar outros dados enquanto não fosse concluído o envio deste segmento; um segmento muito grande poderia alocar o recurso de acesso à rede por horas a fio.

Particionar os dados de uma aplicação em partes menores garantirá que os dados serão transmitidos, respeitando-se os recursos disponíveis de acesso ao meio e simultaneamente não alocar esse recurso de forma exclusiva por um único aplicativo. Desta forma, diversos aplicativos poderão fazer uso do meio pela intercalação de pequenos segmentos gerados individualmente por cada aplicativo.

Não há uma uniformidade em relação à segmentação efetuada pelos protocolos TCP e UDP. Dentre a totalidade de campos que fazem parte do cabeçalho TCP, encontramos o número sequencial que é responsável pelo reagrupamento dos segmentos recebidos pelo *host* de destino. Ou seja; a origem efetua a transmissão dos segmentos, inserindo no campo destinado ao número de sequencia, a numeração relativa a cada segmento que está sendo transmitido. Como poderão existir inúmeras rotas desde a origem até o destino, não haverá garantia de que os segmentos chegarão ao destino na mesma ordem em que foram transmitidos. Assim sendo, utilizando-se o número de sequência, o *host* de destino poderá reordenar os segmentos na ordem em que foram enviados pela origem.

No cabeçalho do protocolo UDP não existe o campo número sequencial, pois este protocolo não tem, entre suas funções, a preocupação de ordenar no destino os datagramas na ordem em que foram enviados. Também o UDP não oferecerá garantia de entrega de dados confiável. Com isto o protocolo UDP se tornará um protocolo mais simples com *overhead* menor que o protocolo TCP, tendo como consequência, a oferta de maior eficiência por ocasião das transmissões de dados, obtendo tempos menores. As aplicações que usarem UDP precisarão admitir que os dados poderão lhe ser entregues fora da ordem em que foram transmitidos. Serão estas aplicações que tratarão tais inconvenientes e não a camada de transporte usando o protocolo UDP.

A grande diferença entre os protocolos TCP e UDP está justamente sobre a questão da confiabilidade que o TCP possibilitará, embora com maior utilização dos recursos da rede em contraposição a maior velocidade de transmissão do UDP sem garantia de entrega. É por meio do uso de sessões orientadas à conexão que o TCP proporciona a confiabilidade nas transmissões. Para atingir este fim, teremos que antes do real envio dos segmentos virem a ser iniciados com o uso do protocolo TCP, existir uma conexão previamente estabelecida entre origem e destino, onde será através desta conexão que poderemos rastrear a sessão ou evidenciar a fluxo de transmissões entre os dispositivos finais da comunicação. Desta forma, no protocolo TCP, antes de haver o envio dos dados da origem ao destino, termos o estabelecimento de uma sessão entre os *hosts*.

Uma vez que houve o estabelecimento da sessão, o *host* de origem iniciará a transmissão efetiva dos dados, através dos segmentos TCP. Estes segmentos, quando recebidos pelo *host* de destino, fará com que o *host* de destino envie uma confirmação de recebimento ao host de origem. O recebimento dessas confirmações irá compor a base do sistema confiável proporcionado pela sessão TCP. Quando a origem receber uma confirmação sobre o envio de um conjunto de segmentos, a origem saberá que este conjunto foi entregue

adequadamente; então, a origem enviará o próximo grupo de segmentos e ficará no aguardo da confirmação para esse novo grupo de segmentos, enviados. Recebendo a confirmação para esse novo grupo de segmentos, o processo de transmissão será dado em continuidade. Caso algum grupo de segmentos não for confirmado dentro de um intervalo de tempo, o TCP de origem retransmitirá o conjunto de segmentos não confirmados à origem.

Observa-se que o tráfego, que é gerado na rede com fins de promover as confirmações e retransmissões, incrementará ao protocolo TCP um maior peso em seu desempenho geral, bem como um maior overhead nos *hosts* de origem e destino, pois exigem-se deles o controle de monitorar quais segmentos estão aguardando pela confirmação e ou pela retransmissão. Some-se a esse procedimento, a exigência das sessões que serão criadas antes do efetivo envio dos segmentos.

Observada a figura 3, temos o cabeçalho TCP com seus campos:

- Porta de origem e de destino (16 bits) – identifica as aplicações que utilizam o TCP.
- Número de sequência (32 bits) - determina o número de sequencia do segmento TCP.
- Número de confirmação (32 bits) - identifica o próximo número de sequência que o receptor espera receber.
- Tamanho do cabeçalho (4 bits) - identifica o tamanho do cabeçalho TCP.
- Reservado (6 bits) – reservado para uso futuro.
- Bits de código (6 bits), sendo eles:
  - URG: indica que o modo de urgência está ativado.
  - ACK: indica que o valor do número de sequência deve ser considerado.
  - PSH: indica que o receptor deve passar o segmento o mais rápido possível.
  - RST: utilizado para reiniciar uma conexão.
  - SYN: utilizado para indicar a sincronização de numeração no início da conexão.
  - FIN: transmissor indica fim da transmissão de dados.
- Janela (16 bits) - indica a quantidade de bytes que o receptor pode receber em um determinado momento.
- Soma de verificação (16 bits) - é utilizado pelo receptor para verificar a integridade de todo o segmento TCP.
- Urgente (16 bits) - válido somente com o modo de urgência ativado (flag

URG), aponta o último byte dos dados urgentes dentro do segmento.

- Opções (tamanho variável) – identifica informações opcionais.
- Dados (tamanho variável) – dados da camada de aplicação.

2.7 Estabelecimento, término de conexões e reagrupamentos TCP

Quando dois *hosts* efetuarem a comunicação, fazendo uso do protocolo TCP, uma conexão deverá ter sido estabelecida previamente ao envio efetivo dos segmentos de dados.

Após as transmissões terem sido concluídas, esta sessão será fechada e a conexão encerrada.

Com o intuito de estabelecer uma conexão inicial entre o *host* de origem e o *host* de destino visando criar uma sessão, será necessário que os *hosts* realizem o que chamamos de *handshake* triplo.

Segundo Tanenbaum (2011, p.323) que, em 1975, Tomlinson criou o *handshake* de três vias. Esse protocolo de estabelecimento não exige que ambos os lados comecem a enviar mensagens com o mesmo número de sequência. O processo do *handshake* triplo prevê que, ao se tentar efetuar uma conexão, o *host* de destino esteja ativo na rede, com o serviço do aplicativo que se deseja alcançar disponível e aceitando requisições no número de porta de destino almejado para o estabelecimento da sessão. Ou seja: o *handshake* triplo avisará o *host* de destino que há um *host* de origem, desejando efetuar conexão para transmissões de segmentos em um determinado número de porta.

No protocolo de transporte TCP é o *host* origem cliente quem iniciará a sessão para o *host* de destino servidor, onde podemos sintetizar as três fases no estabelecimento da sessão TCP como sendo:

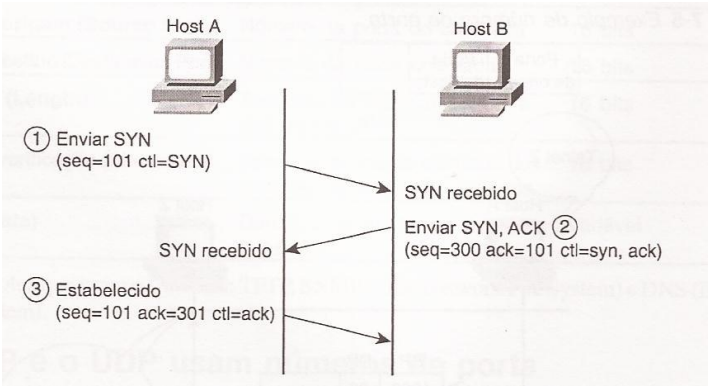
1º O *host* de origem faz a transmissão de um segmento que possui valor inicial de sequencia ao host de destino. No *host* de destino, este segmento será tratado como sendo uma solicitação do host de origem para que uma sessão seja iniciada.

2º O *host* de destino transmite ao *host* de origem um segmento que possui um valor de confirmação da solicitação de sessão, que será igual ao valor sequencial recebido, acrescido de uma unidade, além do seu próprio valor de sequencia de sincronização.

3º Para fechar o ciclo, o *host* de origem responde com uma confirmação cujo valor será também igual ao sequencial recebido acrescido de uma unidade. A figura

6 expõe um exemplo de *handshake* triplo.

Figura 6: Handshake Triplo



Fonte: Mcquery (2002, p. 184

Segundo MCQUERY (2002, p.184), analisando este exemplo, escolhemos:

1º segmento (SYN): indica que o nº de sequência inicial é 100. Este número é escolhido aleatoriamente e é usado para dar inicio ao rastreamento de dados entre o cliente e o servidor nesta sessão.

2º segmento (SYN + ACK): confirma a recepção do primeiro segmento e indica que o seu nº de sequência é 300, sendo o próximo octeto que está à esperar tem o nº de sequência 101. Enfatizamos que as conversações entre o cliente e o servidor são, na verdade, duas sessões unidirecionais: uma do cliente para o servidor; e outra do servidor para o cliente.

3º segmento (ACK): confirma a recepção do segundo segmento indicando que o seu nº de sequência é 101 e que o próximo octeto, que estará à espera de receber, terá o nº de sequência 301. Neste momento, as sessões estão estabelecidas do *host* cliente ao *host* servidor, e desta forma, todos os segmentos que vierem a ser transmitidos terão uma *flag* ACK definida, resultando que os segmentos oriundos da camada de aplicação possam fluir.

Após o *host* de origem enviar todas as solicitações e receber suas repostas, teremos necessidade de encerrar a conexão finalizando a sessão. Para que o protocolo TCP inicie o encerramento da sessão, a *flag* FIN no cabeçalho do segmento precisa ser vinculada. Como em uma sessão TCP, que na verdade temos duas sessões: uma da origem ao destino e outra do destino à origem, será necessário que utilizemos dois *handshake* duplos, compreendendo de um segmento FIN e outro segmento ACK, em cada sessão unidirecional.

Segundo Kurose (2006, p. 196), a finalização de uma sessão poderá ser solicitada pela origem, pelo destino ou por ambos. Para fins de exemplificação, foi adotado que o *host* cliente deu início à solicitação de encerramento à sessão:

1º Ao final das transmissões o *host* de origem envia um segmento com a *flag* FIN definida;

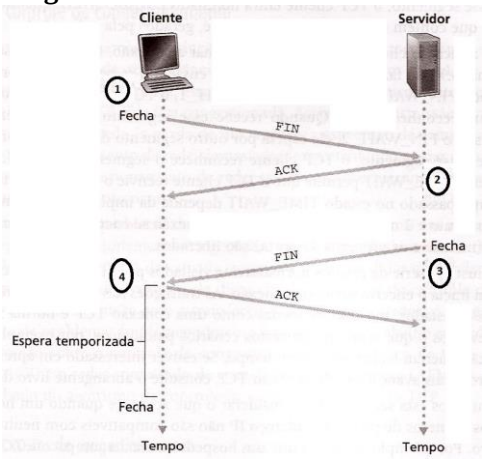
2º O *host* de destino recebe o segmento com a *flag* FIN definida e devolve um segmento com ACK, a fim de confirmar o recebimento do FIN e dar por encerrada a sessão do host origem para o host de destino.

3º O *host* de destino por sua vez, envia um FIN para o host origem, com vistas à encerrar a sessão do *host* de destino ao host de origem.

4º O *host* de origem recebe o segmento com a *flag* FIN definida e devolve um segmento com ACK, a fim de confirmar o recebimento do FIN e dar por encerrada a sessão do host destino para o host de origem.

Com fundamento na figura 7, podemos evidenciar o encerramento de uma sessão com os envios das *flags* FIN e recebimento de ACK, tanto da origem para o destino e vice-versa.

Figura 7: Término de conexões TCP



Fonte: Kurose (2006, p.197).

Mesmo com o estabelecimento das sessões previamente a qualquer transmissão, não haverá garantias de que os segmentos transmitidos cheguem ao destino na mesma ordem em que foram gerados na origem. Para que os dados transmitidos ao destino possam ser compreendidos pelo aplicativo, cumpre que estes dados sejam reagrupados e entregues na mesma ordem em que foram gerados. Fazendo uso do número de sequência é que o protocolo TCP poderá ordenar os segmentos que chegam ao destino fora de ordem.

O protocolo da camada de transporte TCP do *host* de destino depositará os dados recebidos dos segmentos em um *buffer*. Os segmentos ajustados no *buffer* são reagrupados à medida que os outros segmentos vão sendo recebidos e, uma vez finalizado o reagrupamento, estes segmentos serão repassados à camada de aplicação. Os segmentos que alcançarem o *host* de destino fora da ordem de transmissão, serão retidos no *buffer* para posterior processamento.

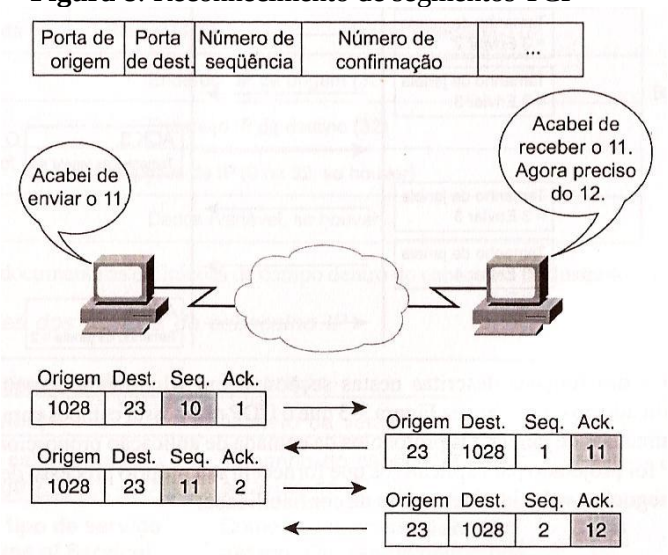
2.8 Confirmação, retransmissão e controle de congestionamento TCP

Como um protocolo da camada de transporte que garante a entrega sistematizada dos segmentos, o TCP recorrerá ao uso da confirmação de recebimento emitida pelo *host* de destino aos aplicativos que efetuaram a transmissão a partir do *host* de origem.

É através do campo do cabeçalho número de sequência e do número de confirmação do protocolo TCP, que de maneira conjunta regem a confirmação e recebimento dos dados que estão contidos do interior dos segmentos. Para calcular o número de sequência em uma sessão, efetuamos a adição da quantidade *bytes* que foram transmitidos e acrescentamos uma unidade. Ainda no decorrer da sessão que fora aberta, o protocolo TCP utiliza o número de confirmação em segmentos que serão transmitidos como retorno ao *host* de origem, com o objetivo de determinar o próximo *byte* que o destino estará aguardando, ou seja o número de confirmação.

O *host* de origem, ao efetuar a recepção do segmento de confirmação do *host* de destino, identificará que o *host* de destino recebeu perfeitamente todos os *bytes* que lhe foram enviados na ultima transmissão, exceto aquela unidade que fora acrescida para fins de cálculo de número de confirmação. A partir deste momento, o *host* de origem deverá enviar um novo segmento de dados ao *host* de destino, cujo número de sequência será igual ao número de confirmação. Embora estejamos reportando-nos à transmissão TCP de forma unidirecional, considerando uma origem transmitindo para um destino, na verdade as conexões ocorrem em duas sessões simultâneas de forma bidirecional.

Figura 8: Reconhecimento de segmentos TCP



Fonte: Mcquery (2002, p. 185)

Segundo MCQUERY (2002, p.185), com base na figura 8 acima, colhemos:

1º Enviado um segmento com 1 byte e um número de sequência igual a 10.

2º O destino recebe o segmento e determina que o número de sequência é 1 e possui 1 byte. O destino envia um segmento de volta para confirmar o recebimento com o número de confirmação igual a 11 como indicativo que o byte de dados que ele aguarda o recebimento nessa sessão é o byte 11.

3º No momento em que a origem receber a confirmação, ela fará o envio do próximo segmento iniciando com o byte número 11.

4º O destino recebe o segmento e determina que o número de sequência é 2 e que ele tem 1 *byte*. O destino envia um segmento de volta para confirmar o recebimento com o número de confirmação igual a 12 para sinalizar que o próximo *byte* de dados que ele aguarda receber nessa sessão é o *byte* 12.

No exemplo relatado, as transmissões estão sendo efetuadas e obtendo confirmação para cada segmento enviado. Na prática, esta situação geraria muito *overhead* nos meios de transmissão e maior processamento nos *host* de origem e destino. A fim de minimizar este problema, um conjunto de segmentos poderá ser enviado e confirmado como um único segmento contendo um número de confirmação resultante do total de *bytes* recebidos pelo host destino.

Desta forma, tendo um *host* de origem um número de sequência inicial igual

3.000 e 20 segmentos de 1.450 *bytes* fossem transmitidos, o *host* destino retornaria como número de confirmação 32.201.

Recebe o nome de tamanho da janela, a quantidade de segmentos que o *host* de origem tem êxito ao enviar ao *host* de destino, não havendo a necessidade do segmento que contém a confirmação ser recebido pelo host de origem. O tamanho da janela também é um campo no cabeçalho TCP que será utilizado para o monitoramento de dados transmitidos e controle de fluxo.

Mesmo que todas as regras que norteiam o planejamento e montagem de uma rede sejam obedecidas, invariavelmente teremos perdas de segmento nas transmissões entre os *hosts* origem e destino. Como o protocolo da camada transporte TCP foi desenvolvido para garantir a entrega dos segmentos, esse protocolo deverá ter mecanismos que identifiquem as perdas de segmentos e providencie a sua recuperação. Uma das formas que o TCP consegue recuperar os segmentos perdidos é efetuar a retransmissão de todos os segmentos que não obtiveram confirmação durante um intervalo de tempo.

Tomemos como exemplo os segmentos com números de sequência 3500 a 5000 e de 5400 a 5500 que foram recebidos pelo *host* de destino. Para este caso, o número de confirmação seria 5001, uma vez que a numeração de sequência dos segmentos entre 5001 a 5399 não foram recebidos e assim sendo, tais segmentos serão solicitados à sua transmissão.

Há possibilidade de o *host* de origem, depois um período de tempo estabelecido, não receber uma confirmação das transmissões dos últimos segmentos enviados. Nestas condições, o *host* de origem deverá efetuar a retransmissão baseado no último número de confirmação recebido. Ou seja: a partir do último número de confirmação recebido em diante o *host* de origem efetuará toda a retransmissão dos segmentos.

Para que haja a efetiva retransmissão a partir da camada de transporte, sem que haja a necessidade de solicitar a camada de aplicação os dados que não foram confirmados, o protocolo TCP efetuará a transmissão do segmento, mantendo uma cópia deste segmento no *buffer* e dará início a uma contagem de tempo. Se houver a confirmação de que o segmento foi recebido pelo *host* de destino, o TCP eliminará do *buffer* o segmento que já foi confirmado. Se a confirmação do recebimento não tiver sido recebida antes da contagem do tempo expirar, o segmento será retransmitido pela origem.

Existem outras formas de controle de segmentos que não foram confirmados, mas para o escopo deste trabalho, concentrar-se-á o excurso científico neste método aqui analisado.

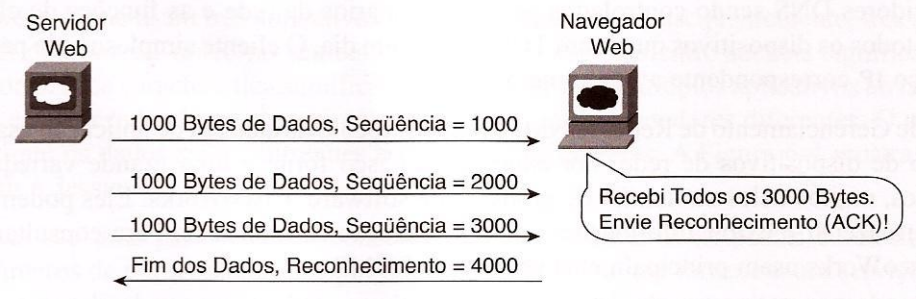
Além da garantia de entrega dos segmentos, através das retransmissões dos segmentos que não forem confirmados pelo destino, o TCP também oferecerá formas de controlar o fluxo de dados. Este controle do fluxo realizado pelo TCP colabora no sentido de ajustar a taxa de fluxo de dados efetiva em uma sessão aberta entre os *hosts* de origem e destino. Quando o host de origem recebe a confirmação que os dados enviados foram recebidos pelo *host* destino, o *host* de origem pode continuar a transmitir mais segmentos para esta sessão.

É através do campo de tamanho da janela no cabeçalho TCP, especificado quanto à quantidade de segmentos, que o *host* de origem irá transmitir antes que o *host* de destino necessite enviar uma confirmação de recebimento. Durante o processo do *handshake* triplo, no momento do estabelecimento inicial da sessão, é que o tamanho da janela inicial será definido.

O valor do tamanho da janela é flexível. O protocolo TCP constantemente ajusta a taxa efetiva de transmissão de segmentos, sempre procurando estabelecer o máximo que o conjunto da rede e do *host* de destino conseguirão processar sem que haja perda de segmentos, e com isto, tenhamos minimizadas as retransmissões de segmentos.

Expresso na figura 9, apresenta-se um exemplo de tamanho da janela e suas confirmações. Neste caso, foi definido que a janela inicial teria um valor de 3000 *bytes*. Isto fará com que o *host* de origem transmita 3000 *bytes*, aguarde pela confirmação de recepção dos segmentos pelo destino e envie mais 3000 *bytes*. Este procedimento será mantido até que o *host* de origem envie todas as informações desejadas ao *host* de destino.

Figura 9: Reconhecimento e tamanho da janela TCP



Fonte: Odom (2008, p. 102).

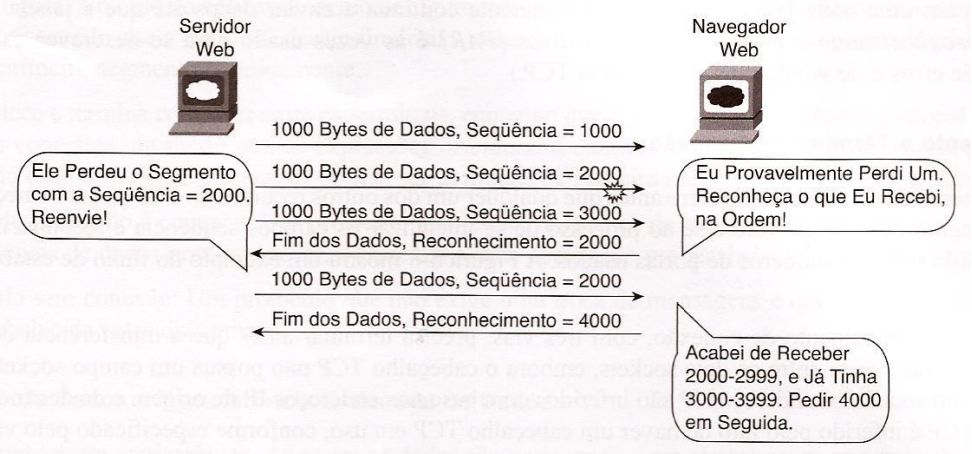
Depois que o *host* de origem fizer o envio dos segmentos baseado no tamanho da janela TCP, não haverá qualquer envio de novos segmentos até que o *host* de destino faça a confirmação da recepção dos segmentos já enviados. Devido ao congestionamento da rede, somado aos recursos do *host* de destino, que poderão ter ficados escassos por execução de

outros aplicativos, teremos atrasos no envio da confirmação de recepção por parte do *host* de destino, com consequente diminuição no envio de novos segmentos pelo *host* de origem. Embora fique mais lenta a transmissão de dados, isto acaba por reduzir a exigência dos recursos da rede e do host de destino, liberando-os para que não haja uma sobrecarga.

Outra técnica utilizada para controlar o fluxo de dados é fazer uso de tamanho de janelas variáveis ou dinâmicas. Quando a rede se tornar congestionada e seus recursos se mostrarem diminutos, o protocolo TCP buscará reduzir o tamanho da janela, forçando com que os segmentos transmitidos sejam monitorados mais intensamente, pois haverá maior quantidade de confirmações de recebimento, diminuindo, na prática, a velocidade de transmissão.

Como parte da definição do tamanho da janela, o *host* de destino envia ao *host* de origem, o número de *bytes* que estaria em condições de receber na sessão que se estabeleceu. Durante a transmissão desta sessão, outras requisições poderão chegar ao *host* de destino o que pode gerar concorrência sobre os seus recursos. Neste caso, o host de destino, poderá necessitar diminuir a taxa de transmissão que, para tanto, enviará ao *host* de origem, junto com o segmento de confirmação, um novo valor do tamanho de janela menor que espera receber.

Figura 10: Congestionamento TCP e controle de fluxo



Fonte: Odom (2008, p. 103).

Na figura 10, expressa-se a representação da ocorrência da interpretação de um congestionamento, a partir da perda dos 1000 *bytes* enviados na sequência 2000.

Conforme discorrido, o tamanho da janela é variável ou dinâmico e no exemplo anterior o tamanho da janela foi decrescido em virtude do congestionamento ocorrido. Entretanto, os problemas que ensejaram a diminuição do tamanho da janela podem-se esvair

e, neste caso, haveria necessidade de que o TCP retornasse o tamanho da janela para valores maiores. Este procedimento será realizado quando, após períodos de transmissão sem perda de segmentos ou com o *host* de destino possuindo mais disponibilidade de recursos, o destino começará a aumentar o tamanho da janela nos segmentos de confirmação. Tal procedimento irá tornar mais leve o uso dos recursos da rede, pois haverá menos confirmações a serem enviadas. O *host* de destino irá elevar continuamente o tamanho da janela, até que inicie a perda de segmentos, instante em que retornará o processo de diminuição do tamanho da janela.

Segundo Odom (2008, p.103), desta forma, pode-se evidenciar que o controle de congestionamento tem uma relação direta com tráfego que está fluindo na rede TCP. Se o fluxo de segmentos for muito elevado para a capacidade da rede, haverá perda de segmentos e o TCP iniciará a diminuição no envio de segmentos, com o fim de não sobrecarregar o tráfego mais ainda.

No momento que o TCP evidenciar o recebimento de vários ACK's sequenciais, será entendido que a rede foi aliviada e o TCP poderá agora elevar a janela, fazendo com que mais segmentos possam aproveitar a boa disponibilidade apresentada pela rede. Este ciclo de crescer e decrescer a janela de transmissão no envio dos segmentos permanecerá durante toda a sessão, enquanto ocorrerem congestionamentos na rede.

2.9 Opções de protocolos TCP sobre redes sem fio

> Os protocolos de transporte como o TCP, que implementam controle de congestionamento, devem ser independentes das tecnologias subjacentes das camadas de rede e enlace. Essa é uma boa teoria, mas na prática existem problemas com redes sem fio. O principal deles é que a perda de pacotes normalmente é usada como sinal de congestionamento, inclusive pelo TCP, conforme já discutimos. As redes sem fio perdem pacotes o tempo todo, devidos a erros de transmissão. (TANENBAUM, 2011, p. 338)

A possibilidade de perda de pacote nas transmissões sobre redes sem fio é muito mais elevada que nas redes fisicamente conectadas. Quando o TCP enviar segmentos e não receber a confirmação do recebimento destes, o TCP entenderá como *timeouts* por congestionamento dos meios de transmissão. O TCP não tem como diferenciar se está havendo *timeout* por congestionamento dos meios de transmissão ou perda dos segmentos por problemas nos meios de transmissão. Quando ocorrer congestionamento, o melhor será diminuir as quantidades de transmissão por intervalo de tempo, para aliviar o tráfego; já quando ocorrer perda de segmentos, o melhor seria reenviar o mais rápido possível. Como o

TCP não consegue diferenciar quando ocorre *timeout*, de quando ocorre perda de segmentos, vai sempre diminuir o ritmo das transmissões. Em uma rede sem fio com altas taxas de perdas de segmentos, o fenômeno físico fará com que o TCP diminua o ritmo das transmissões ocasionando piora na situação geral da rede.

Conforme será tratado no capítulo subsequente sobre as redes sem fio, serão evidenciadas mais detalhadamente quais características e problemas inerentes que são prejudiciais ao desempenho do protocolo TCP. Com o objetivo de adequar esta realidade da camada de enlace, aos protocolos da camada de transporte, existem propostas para a adequação do protocolo TCP que podemos classificar em três modalidades: protocolos de camadas inferiores ao TCP; protocolos com quebra de conexão e finalmente os protocolos fim-a-fim.

2.9.1 Os protocolos de camadas inferiores ao TCP

Tem como principal característica a criação de um *buffer* com o fim de armazenar os segmentos que passam por um determinado ponto de agregação das redes fixas e redes móveis, com o objetivo de gerenciar as confirmações e retransmissões de segmentos. São versões dos protocolos de camadas inferiores ao TCP:

a) Protocolo Snoop: Segundo VANGALA (2002) a proposta do protocolo *snoop*, concentra-se em efetuar análise de todos os pacotes que trafegam em ambos os sentidos entre um *host* fixo e outro móvel, armazenando em *cache* os segmentos, até que seja identificado o envio de um ACK com a confirmação do recebimento dos dados pelo host de destino. Na proposta deste protocolo, o AP em que o Host em mobilidade esteja conectado, é que fará o papel de ler os segmentos que passam e armazenar em seu *cache*, até que haja o recebimento do ACK de confirmação, pois se houver a necessidade de retransmissão de segmentos não confirmados, a retransmissão será realizada a partir do AP, e não pelo *host* fixo de origem das transmissões.

Um diferencial nesta proposta é que não haverá alteração no protocolo TCP, que é usado na rede fixa, uma vez que o controle das perdas no ambiente sem fio será realizado pelo próprio AP, além do que será mantida a conexão fim-a-fim do TCP existente entre o *host* de origem e o *host* de destino. Entretanto, esta solução, em virtude da necessidade de o AP estar sempre efetuando a leitura dos segmentos e, por conseguinte, dos endereçamentos constantes nos pacotes e quadros, a fim de viabilizar a retransmissão caso haja necessidade, esta proposta

não oferecerá suporte à criptografia, limitando-se a prover serviços a camada de aplicação.

b) **Protocolo Snoop Otimizado:** Santana (2003) nos diz que a proposta do protocolo *snoop* otimizado é uma implementação visando além de melhorar a performance em redes sem de fio, reduzir alguns problemas do *snoop* que se relacionam quando há uma grande perda de segmentos por ocasião de uma desconexão. Quando da utilização do *snoop* em uma rede sem fio, pode haver perdas no fluxo de segmentos da janela do receptor, e como o *snoop* retransmite apenas um segmento de cada vez, torna-se insuficiente para que, havendo perdas de grandes quantidades de segmentos, não venha a ocorrer um estouro do temporizador no host fixo. O que é proposto pelo *snoop* otimizado, visa desenvolver uma técnica capaz que o TCP receptor do *host* móvel identifique para o AP quais são os segmentos que não foram confirmados, quais estão em falta. Este procedimento será realizado através do uso de um ACK seletivo, fazendo com que o AP venha a reenviar todos os segmentos de uma vez. Desta forma, não haverá a retransmissão de um segmento e esperar pelo próximo ACK para retransmitir outro; serão enviados todos os segmentos faltantes de uma só vez.

2.9.2 Os protocolos com quebra de conexão

Tem como característica principal a quebra da conexão estabelecida entre o *host* de origem e o *host* de destino em duas, *send*o que o AP tornar-se-á o agregador das conexões. Ou seja: teremos duas sessões: uma sessão do *host* com acesso a rede por meio de conexão fixa até o AP e outra sessão do AP até o host com acesso a rede por meio de uma conexão sem fio. Embora com a divisão da conexão, algumas versões mantêm a lógica fim-afim do protocolo. São versões do TCP que empregam a quebra da conexão:

a) Mecanismo de Notificação Explícita: Segundo FLOYD (1994) Explicit Congestion Notification (ECN) é uma técnica que foi originada de outra que normalmente é aplicada por roteadores, a Random Early Detection (RED). Esta técnica possibilitará aos roteadores que, antes de ocorrer um congestionamento total, estes equipamentos sinalizarão seu estado de sobrecarga com a ação de descarte para alguns pacotes aleatoriamente, objetivando a prevenção do congestionamento. No exemplo desta técnica, aplicada e utilizada pelos

roteadores, o mecanismo de notificação explicita buscará não descartar os pacotes, mas realizar a sua marcação quando o estado da fila se encontrar em um nível que seja considerado de sobrecarga. Desta forma, na ocorrência de um ACK enviado ao transmissor, a marcação será enviada e o transmissor dá início ao processo de congestionamento do TCP, sem no entanto, haver tido perda efetiva de pacotes.

b) Técnicas de Reconhecimentos Múltiplos: Segundo PATEL (2001), como o AP é o ponto limite entre o ambiente da rede fixa e o ambiente da rede móvel, este AP deverá criar um ACK provisório (ACKp) para ser enviado ao transmissor, com vistas a marcar o recebimento até o ponto de acesso, bem como um outro ACK para confirmar o recebimento pelo host móvel (ACKc). É necessário, também, que os temporizadores do transmissor sejam modificados. Por isso, há quatro tipos de temporizador, que são os RTT, RTT(w), RTO, RTO(w), *send*o ativados em cada segmento transmitido. As funções destes temporizadores podemos evidenciar na tabela 2 abaixo.

Tabela 2: Temporizadores RTT, RTT(w), RTO, RTO(w)

Siglas	Descrição dos Temporizadores
RTT	Tempo de transmissão e recepção do ACKc vindo do receptor
RTT(w)	Tempo de espera pelo ACKp vindo do AP
RTO	Tempo máximo que o transmissor espera por ACKc vindo do receptor
RTO(w)	Tempo máximo que o AP espera por um ACKc vindo do host móvel

Fonte: Elaborado pelo autor (2013)

Então, quando um segmento for transmitido pelo *host* fixo, ele será armazenado pelo AP e imediatamente confirmado através de um ACKp. Ao ser confirmado pelo *host* móvel, um ACKc será enviado para o *host* fixo. Em caso de perda na rede sem fio, os temporizadores do *host* fixo já estarão ajustados através do ACKp que fora enviado pelo AP, evitando-se assim a retransmissão desnecessária.

a) TCP Indireto (Indirect TCP): Bakre (1994) afirma que a proposta do TCP Indireto é de seccionar a conexão entre um *host* fixo e um *host* móvel em duas partes distintas e isoladas, culminando com que não haja nenhuma relação entre as mesmas. Esta secção seria realizada no AP, que seria responsável pela coordenação das transmissões entre os *hosts* de origem e destino. Ou seja, o AP recebe o segmento oriundo do *host* fixo e faz a devolução de um ACK confirmando sua recepção. Em seguida este segmento é armazenado no AP e

transmitido para o *host* móvel, *send*o mantido no *cache* do AP, enquanto o *host* móvel não faça a confirmação do recebimento através de um ACK. Como nesta solução coexistem duas redes distintas, o protocolo de transporte a ser utilizado no segmente de rede sem fio não precisa ser o TCP. Mas em contrapartida as aplicações do *host* móvel necessitam ser recompiladas para acesso aos serviços disponíveis na rede. Uma das vantagens da adoção de uma solução com esta, reside no controle de fluxo independente, uma vez que as redes serão distintamente tratadas, possibilitando identificar as características individuais de cada rede separadamente. Saliente-se também, que as alterações necessárias serão implementadas apenas no AP e no *host* móvel, mantendo-se inalterado o *host* fixo. Como desvantagem desta proposta, podemos evidenciar a quebra da conexão origem e destino (fim-a-fim) do TCP, além de exigir um AP com capacidade de suportar a sobrecarga que será gerada.

2.9.3 Protocolos com conexão fim-a-fim

Os protocolos com conexão fim-a-fim se baseiam no princípio de que a transmissão e confirmação ocorrerão entre os dois pontos sem intermediação, ou seja, sem a necessidade de qualquer tipo de ponto coordenador. São protocolos com conexão fim-a-fim:

a) **Freeze TCP:** Segundo GOFF (2000), a proposta desta solução prende-se em não utilizar nenhum ponto intermediário na conexão, ficando o controle a ser realizado pelo receptor. O funcionamento da proposta baseia-se no envio de pacotes de confirmação ACK onde o tamanho da janela do receptor seja zerado, obrigando o transmissor a entrar em modo persistente. Uma vez em modo persistente, o transmissor verificará periodicamente o estado do receptor, confirmando a permanência da janela zerada - Zero Window Probe, ou não. Para dar recomeço a transmissão de forma mais eficiente, a solução fará a proposição de que sejam transmitidos pelo receptor três ACK consecutivos, com a informação do último segmento recebido. Este procedimento será interpretado pelo TCP padrão, como uma requisição para iniciar o modo de retransmissão rápida no host transmissor. O objetivo desta solução é minimizar os problemas decorrentes das frequentes desconexões e não em virtude dos erros comuns em redes sem fios. Um problema que poderá comprometer o uso efetivo desta solução é o momento em que se deseja a interrupção da

transmissão, pois a *host* pode tentar transmitir o pacote ZWA. Uma vez que não esteja mais conectado, o transmissor interpretaria como perda resultando na ativação do mecanismo de congestionamento do TCP.

b) **TCP WestWood:** Bandwidth Estimation for Enhanced Transport over *Wireless* Links, ou TCPW, segundo MASCOLO (2001) é uma proposta que emprega modificações no transmissor, no que se refere ao controle de congestionamento. Esta proposta se baseia no controle da janela de congestionamento por meio de estimativas da taxa de transmissão fim-a-fim. Desta forma, mantém-se a compatibilidade com qualquer tipo de rede e também com a versão do TCP no receptor. O tempo estimado de transmissão será obtido a partir do RTT, e é utilizado para se obter a configuração do tamanho da janela de congestionamento logo após a evidência de uma perda. Nesta solução, temos algumas vantagens em relação às propostas existentes, onde podemos referir que não haverá necessidade de pontos intermediários; mantem-se a semântica fim-a-fim do TCP; manteve um desempenho satisfatório, mesmo em redes heterogêneas. Uma grande desvantagem poder-se-ia mencionar, será a necessidade de modificações no TCP transmissor, fato que tornará inviável essa solução, haja vista a grande utilização em nível mundial da versão TCP Reno.

3 TECNOLOGIAS DE REDES LOCAIS SEM FIO

> Quase na mesma época que surgiram os notebooks, muitas pessoas sonhavam com o dia em que entrariam em um escritório e, como mágica, seus notebooks se conectariam à Internet. Em consequência disso, diversos grupos começaram a trabalhar para descobrir maneiras de alcançar esse objetivo. A abordagem mais prática foi equipar o escritório e os notebooks com transmissores e receptores de rádios de ondas curtas para permitir a comunicação entre eles. (TANENBAUM, 2011, p43).

Os sistemas tecnológicos que envolvem a transmissão sem fio oferecem condições para a transmissão de dados e informações entre dois ou mais dispositivos conectados, sem que para isso, tenha a necessidade do uso de qualquer condutor físico, quer sejam meios elétricos quer ópticos entre os *hosts* que se comunicam. Uma das principais vantagens para a utilização de um sistema sem fio em ambientes de redes locais é a facilidade proporcionada pela mobilidade ofertada aos usuários. A facilidade ocasionada pela ausência de condução do sinal de transmissão através de meios físicos elétricos ou óticos ocasionará um grande atrativo às redes locais sem fio, entretanto, observa-se que o sinal da transmissão em redes sem fio é extremamente mais suscetível à interferência e degradação em comparação as redes com o uso de cabos. Estes problemas inerentes às redes sem fio precisarão ser considerados, detectados e tratados pela Tecnologia pertinente para possibilitar um nível de confiança satisfatório, permitindo o uso das mais diversas aplicações.

Estando disponível ao mercado desde o ano de 1997, a utilização da tecnologia *wireless* em redes locais, vem permitindo aos usuários a mobilidade no ambiente de sua empresa ou mesmo na comodidade de sua residência. Também será possível, através do uso de antenas com ganho apropriado, expandir o alcance das redes atuais, principalmente em locais onde é muito difícil, se não impossível, a instalação de infraestrutura física de cabeamento. Como exemplo, temos os provedores de acesso à Internet sem fio, que, dispondo de um ponto cabeado de conexão com a rede global, poderão dispor deste ponto através de um rádio e antenas apropriadas para a cobertura de um bairro ou pequeno povoado. Com o aumento do alcance das redes e provendo a mobilidade, as redes *wireless* elevarão a agilidade da troca de informações, pois os dispositivos conectados podem sofrer a locomoção para qualquer lugar dentro da área de cobertura mantendo a possibilidade de acesso aos aplicativos e bases de dados em tempo real. Ressalte-se deste modo que o principal objetivo das redes locais sem fio não é o de substituir as redes cabeadas, mas almejar agregar valor as mesmas.

3.1 Aplicações para *Wireless* LAN suas vantagens e desvantagens

A mobilidade característica inerente às redes sem fio ofertam condições aos usuários para realizar um salto em sua produtividade. Em localidades tais como armazéns, lojas, hospitais, etc., o uso efetivo de coletores de dados *wireless* proporcionarão o trânsito das informações de forma rápida e precisa, consolidando dados de uma empresa e possibilitando a disponibilidade de informações valiosas, que poderão ser acessadas por qualquer usuário habilitado. Enquanto a outros ambientes tais como hotéis, aeroportos, universidades e escritórios, o uso de um notebook com uma placa *wireless* oferecerá ao usuário total flexibilidade. Tomemos alguns exemplos de aplicações em redes locais sem fio:

1) **Armazéns:** No instante da recepção ou na saída das mercadorias, coletores poderão ser usados no momento da descarga dos caminhões, executando a leitura das informações dos itens em trânsito.
2) **Lojas de varejo:** De forma similar aos armazéns, porém em uma proporção menor, as lojas poderão ter o gerenciamento de seus estoques também controlados através de coletores.
3) **Hospitais:** Periodicamente, pacientes internados recebem visitas dos médicos e estes fazem anotações que, mais tarde, serão passadas para outros médicos em períodos de virada de turno. Caso a coleta dessas informações seja feita por um dispositivo sem fio, os dados atualizados do paciente poderão ser vistos imediatamente por quaisquer outros profissionais deste estabelecimento.
4) **Aeroportos:** Redes sem fio instaladas em salas de espera possibilitarão aos passageiros o uso da tecnologia para acesso à Internet.
5) **Universidades:** Traz ao aluno a capacidade de trabalhar em seus projetos, movendo-se por qualquer ponto do campus.
6) **Escritórios:** Oferece aos funcionários a capacidade de movimentação por diferentes andares do prédio, acessando outras áreas, salas de reunião e demais dependências da empresa.

Como já discorrido, as redes locais sem fio têm como principal objetivo somar possibilidades às redes que utilizam meios físicos como cabos em suas conexões. Passemos a enumerar as principais vantagens e desvantagens das redes locais sem fio:

❖ **São vantagens das redes locais sem fio:**

1) Acesso às informações em qualquer local de cobertura do sinal.

2) Instalação em locais onde o cabeamento é oneroso ou proibido, tais como: baixo número populacional, prédios alugados ou tombados pelo patrimônio histórico.

3) Instalação em prédios antigos onde não se conhece a estrutura para quebrar paredes e inserir novos dutos.

❖ **São desvantagens das redes locais sem fio:**

- Com relação ao diagnóstico de problemas e a segurança, as redes locais sem fio são mais complexas.
- A manutenção da garantia de qualidade do serviço em redes sem fio é um desafio constante por causa das interferências.
- Não se tem conhecimento acerca dos efeitos que podem causar ao corpo humano a exposição prolongada da radiação em ambientes assistidos por redes locais sem fio.

3.2 Princípios da rádio frequência: ondas eletromagnéticas, modulação, principais tipos de modulação e modulações IEEE 802.11

Para um entendimento dos princípios da rádio frequência, colhe-se, inicialmente, abordar sobre as ondas, pois serão através das ondas de rádio que as informações serão enviadas da origem ao destino.

Onda é uma perturbação no meio que se propaga de um ponto a outro e que poderá variar de acordo com a sua origem, direção da oscilação e tipo da energia transmitida. Com relação à origem, uma onda poderá ser classificada como mecânica ou eletromagnética. As ondas mecânicas são as produzidas por uma perturbação num meio material como, por exemplo, uma onda na água, a vibração de uma corda de violão, a voz de uma pessoa propagada pelo ar ou o vento em uma bandeira.

Tais como as ondas de rádio, televisão e micro-ondas, as ondas eletromagnéticas são produzidas por variações do campo elétrico e do campo magnético. Ao contrário das ondas mecânicas, as ondas eletromagnéticas poderão trafegar tanto por meios materiais quanto pelo vácuo.

Em relação à direção da oscilação, uma onda poderá ser categorizada como transversal ou longitudinal. Para ilustrar este conceito, considere-se uma corda segura por duas pessoas nas extremidades, onde a pessoa na extremidade da esquerda levanta e abaixa a corda rapidamente. Forma-se um pulso de onda que estará se propagando na horizontal, da

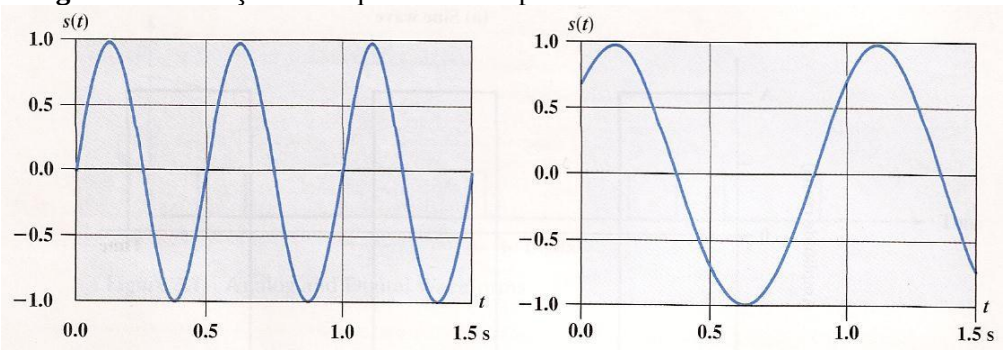
esquerda para a direita que é a direção da perturbação, enquanto os pontos da corda, os perturbados pelo pulso, estão oscilando para cima e para baixo que é a direção da propagação. Desta forma, a direção da oscilação vertical é perpendicular à direção de propagação horizontal. Neste caso, a onda será classificada como transversal.

Agora, se se considerar como exemplo uma pessoa falando, o som da voz dessa pessoa se propagará no espaço em todas as direções - direção da propagação, afastando-se da fonte - direção da perturbação. Neste caso, a onda será classificada como longitudinal. Por último, as ondas podem ser classificadas quanto ao tipo de energia transmitida: sonoras, luminosas, térmicas ou eletromagnéticas.

As ondas eletromagnéticas possuem dois pontos característicos principais, a saber:

1) Todas as ondas eletromagnéticas viajam na velocidade da luz.
2) Frequência e comprimento da onda são inversamente proporcionais. Deste modo, ao se elevar a frequência da onda, menor será o seu comprimento conforme mostrado na figura 11.

Figura 11 – Relação de frequência e comprimento de onda



Fonte: Stallings (2007, p. 70)

As ondas eletromagnéticas só terão utilidade para as comunicações de uma maneira geral, se puderem transportar informações. Tal qual um surfista que se aproveita da força e propagação da onda mecânica dos mares para surfar pelas praias, também precisamos inserir informações nas ondas eletromagnéticas para transmitir os dados. O processo de inserção de sinais em uma onda eletromagnética é chamado de modulação. Desta forma, modulação em sistemas sem fio é o processo através do qual o som, imagem ou dados, serão adicionados às ondas de rádio.

Existem diferentes métodos usados para a modulação, os quais serão determinantes para o melhor aproveitamento da largura de banda. O resultado alcançado por um método de modulação dependerá da qualidade do meio: quanto maior o ruído, menor será

a taxa de transmissão efetiva.

Nos sistemas sem fio, a técnica de modulação define o mecanismo de codificação dos dados de forma que eles sejam inseridos em ondas de rádio. Além da banda de frequência, as técnicas de modulação são importantes para a determinação da capacidade máxima de transmissão. Quanto mais sofisticada for a técnica, melhor será o aproveitamento da banda disponível.

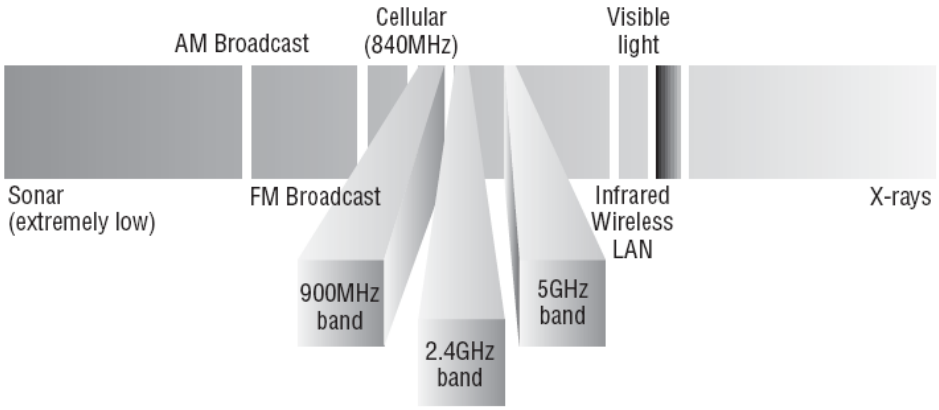
Adicionalmente, as técnicas de modulação precisam levar em consideração a susceptibilidade de um sistema à interferência, garantindo o melhor aproveitamento da transmissão.

Com o objetivo de prover a transmissão dos dados através das redes sem fio mais eficientes, há que se considerar os seguintes objetivos nos sistemas de conectividade: maior alcance, maior taxa de transmissão, menor susceptibilidade a interferências, melhor aproveitamento da banda disponível. Almejando atingir estes objetivos, dispomos de algumas variáveis que necessitam ser consideradas: potência do transmissor, tamanho da faixa de frequência e método de modulação.

A constante pesquisa por melhoria nos sistemas de transmissão sem fio objetiva atingir maior alcance com alta taxa de transmissão, *send*o o mais imune possível à interferência e tendo o melhor aproveitamento da largura de banda.

Lammle (2007, p.706) nos ensina que maior for a faixa de frequência (largura de banda) e quanto mais informação puder ser codificada nessa faixa de frequências (modulação), maior a eficiência da transmissão, fazendo com que um maior volume de dados seja transmitido. Dentre as frequências disponíveis, foram padronizadas algumas não licenciadas de uso livre conforme figura 12.

Figura 12: Frequências não licenciadas



Fonte: Lammle (2007, p. 706).

Para garantir que a informação codificada tenha alcance maior e de melhor qualidade, o transmissor das ondas deverá ter uma potência adequada para encaminhar os dados na distância estabelecida e um método de modulação sofisticado, no sentido de tornar a transmissão mais imune a interferências.

Dentre as formas de modulação existentes, podemos destacar as seguintes: Spread Spectrum, Frequency-Hopping Spread Spectrum, Direct Sequence Spread Spectrum e Orthogonal Frequency Divison Multiplexing.

Passemos a abordar cada uma destas formas de modulação:

1) **Spread Spectrum**: Esta forma de modulação nasceu de um esforço da 2ª Guerra Mundial, com o objetivo de teleguiar os torpedos em direção aos seus alvos corrigindo as suas rotas. Antes do surgimento desse tipo de transmissão de sinais de rádio, os torpedos eram guiados ou por fio ou por transmissões de rádio em uma frequência única. Os fios limitavam a distância dos torpedos e uma transmissão à rádio em frequência única era facilmente captada e anulada pelo inimigo. A solução foi criar um sistema de transmissão à rádio que não se fixasse em uma única frequência, mas que saltasse entre um grupo de frequências de forma pseudo-aleatória de forma síncrona entre o transmissor e o receptor. Desta forma, o inimigo poderia captar pequenas partes das transmissões que lhe pareceriam algum tipo de interferência. Este sistema foi patenteado em 1942, para uso militar e liberado para uso comercial em 1986.
2) **Frequency-Hopping Spread Spectrum**: A tecnologia FHSS - Frequency-Hopping Spread-Spectrum é um esquema de modulação *spread spectrum* que utiliza uma portadora de banda estreita alterando a frequência segundo um padrão conhecido pelo transmissor e receptor. Sincronizados adequadamente,

eles mantêm um único canal lógico. Para um dispositivo de recepção não desejado, o FHSS será interpretado como um pulso ruído de curta-duração. A tecnologia FHSS faz a utilização da largura de banda de forma ineficaz com o objetivo de garantir segurança. Por este motivo, os sistemas FHSS costumam apresentar velocidades de transmissão menores do que outros sistemas de modulação spread spectrum. Dispositivos de *Wireless* LAN têm um desempenho mais lento (1 Mbps) quando utilizam a modulação FHSS.

3) **Direct Sequence Spread Spectrum**: A modulação DSSS aloca uma faixa de frequência e a utiliza constantemente. O DSSS é um esquema de modulação *spread spectrum* que gerará um padrão redundante de *bit*s para cada *bit* transmitido. O padrão de *bits*, denominado *chip* ou código de *chip*, permitirá aos receptores filtrar sinais que não utilizam o mesmo padrão, incluindo-se ruídos ou interferências. O código de *chip* cumpre duas funções principais:

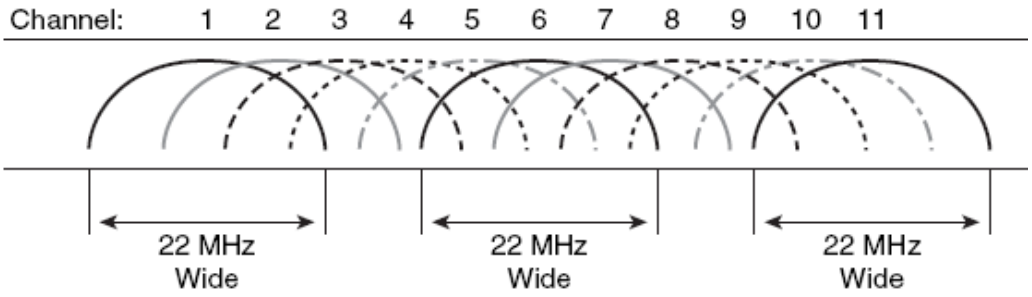
   - Identificar os dados para que o receptor possa reconhecê-los como pertencentes a um determinado transmissor; o transmissor faz a geração do código de chip e exclusivamente os receptores que conhecem o código são capazes de decifrar os dados.

   - Distribuir os dados pela largura de banda disponível; quanto maiores os *chips*, maior a largura de banda necessária, porém garantem maior probabilidade de recuperação dos dados originais; ainda que um ou mais *bits* do *chip* sejam corrompidos durante o processo de transmissão, a tecnologia agregada ao rádio efetua a recuperação os dados originais: usando técnicas estatísticas sem necessidade de retransmissão.

Os receptores não desejados em banda estreita ignoram os sinais de DSSS, considerando-os como ruídos de potência baixa em banda larga. As tecnologias de redes sem fio 802.11b usam a modulação DSSS e apresentam maior taxa de transmissão de dados do que a alternativa de modulação FHSS, devido à menor sobrecarga do protocolo DSSS.

Segundo CARROLL (2009, p.11), a modulação DSSS utiliza a banda de 2.4 GHz na faixa de frequência de 2.400 a 2.483 MHz, agrupado em 14 canais. Cada canal no DSSS tem uma largura de banda de 22 MHz alocada, onde estes canais têm sobreposição de frequência. O uso simultâneo de canais em uma mesma região poderá ocasionar perda de desempenho se a escolha dos canais for feita de forma incorreta. A fim de garantir que não haverá sobreposição de canais, deve-se utilizar os canais na mesma região física com espaçamento de cinco canais entre eles. A figura 13 demonstra os canais que não se sobrepõe.

Figura 13 - Sobreposição de Canais em DSSS



Fonte: Carroll (2009, p. 11)

Assim *send*o, as combinações possíveis, nos EUA, sem causar interferência seriam:

1, 6 e 11, 2 e 7, 3 e 8, 4 e 9, 5 e 10. Utilizando-se sempre estes canais como vizinhos, existe a garantia de não haver interferência gerada por outro dispositivo de seu controle.

4) **Orthogonal Frequency Divison Multiplexing**: A técnica de modulaçãoOFDM consiste na divisão da largura de banda disponível em pequenas faixas denominadas subcanais, tomando-a parecida com a modulação FDM (Frequency Division Multiplexing) amplamente empregada nas transmissões de TV a cabo. A FDM implica na utilização da banda de segurança entre canais adjacentes. Esta banda de segurança tem a função de evitar interferência entre os canais, que poderá ocorrer devido a simples variações na frequência central das respectivas portadoras. Esta banda de segurança, por não transportar informação útil (dados, voz, imagem), será "desperdiçada" para garantir a qualidade das informações transportadas. A OFDM emprega um conjunto de frequências de portadoras de subcanais ortogonais entre si. Isto significa que o valor máximo de uma determinada portadora coincidirá com o valor mínimo das demais. Desta forma, subcanais adjacentes podem ser alocados muito mais próximos uns dos outros sem estarem sujeitos a interferências dos canais adjacentes. De acordo com a banda necessária e sob demanda, poder-se-ão alocar vários subcanais de forma a atender a uma determinada necessidade específica, como, por exemplo, transmissão de voz e dados em subcanais distintos. Sob o ponto de vista de confiabilidade, pode-se empregar uma variante da OFDM, a COFDM (Coded

OFDM). A diferença reside na utilização de algoritmos de detecção e correção de erros antes da transmissão das informações, possibilitando haver a recuperação dos dados no destino.

O IEEE 802.11 foi o primeiro padrão de redes sem fio a operar numa faixa de frequência de 2,4 GHz. Este primeiro padrão definiu a comunicação sem fios com três opções de modulação distintas: infravermelho, DSSS e FHSS. A modulação baseada em infravermelho não chegou a ser implementada comercialmente. As outras duas formas baseadas em radio frequência foram adotadas nos produtos de redes sem fio.

Em 1999 surgiu o padrão IEEE 802.11b (uma variação do 802.11), que trouxe como melhoria a possibilidade de se operar em taxas de transmissão maiores, *send*o agora 5,5Mbps e 11Mbps. Este padrão utiliza a modulação DSSS.

O IEEE 802.11a também utiliza rádio frequência, mas opera em frequências mais altas: 5GHz, permitindo comunicação de até 54Mbps.

O IEEE 802.11g é uma evolução do IEEE 802.11b, já que opera na mesma faixa de frequência de 2,4GHz, e mantém a compatibilidade com esse padrão. Porém, ele poderá ser também visto como uma fusão dos dois padrões, o IEEE 802.11a e o IEEE 802.11b.

Ele usa o que cada um tem de melhor: a modulação do IEEE 802.11a (OFDM) e a faixa de frequência de 2,4GHz do IEEE 802.11b.

3.3 Terminologias em redes sem fio e Mobilidade

Uma das grandes vantagens no uso de redes sem fio é a mobilidade que esta poderá proporcionar aos *hosts* conectados. A fim de definir os tipos de mobilidade que poderão ocorrer no âm*bit*o das redes sem fio, passaremos a definir as terminologias mais utilizadas em redes locais sem fio:

1) **O Access Point (AP):** É uma combinação de hardware e *software* que faz o papel de integrador de clientes. O AP comuta o tráfego entre os clientes da mesma rede sem fio ou entre clientes da rede cabeada com clientes da rede sem fio. Sua função básica é monitorar quem está transmitindo e garantir que não haja colisão, ou seja, que dois clientes não transmitam ao mesmo tempo. Além disto, o AP também cuida da autenticação, da troca de chaves e até da priorização de tráfego no meio *wireless*.
2) **O Distribution System (DS):** É o segmento de rede que interliga os pontos de acesso. Este segmento de rede normalmente é cabeado e utiliza as tecnologia

Ethernet ou Fast Ethernet para esta conexão. Não existe restrição no padrão IEEE 802.11 em relação à tecnologia usada entre os AP. O DS é necessário não só para controle entre os pontos de acesso como também para permitir o roaming entre clientes conectados a diferentes AP.

3) ***Wireless* Distribution System (WDS):** Permite a interconexão de duas redes locais cabeadas através da tecnologia *wireless*. Usado entre prédios ou localidades distintas transformando-as em uma única rede. Desta forma tem-se uma integração entre os ambientes cabeados e sem fio. Para tanto, o ponto de acesso estará configurado no modo bridge, e não atenderá às requisições dos clientes *wireless*.
4) **Basic Service Area (BSA):** Nome dado à área de abrangência do sinal *wireless* entre os clientes.
5) **Extended Service Area (ESA):** É a área total de abrangência da rede, composta da união das áreas dos Basic Service Areas.
6) **Basic Service Set (BSS):** Nome dado a um conjunto de clientes aptos a se comunicarem. Existem dois tipos de BSS: Infrastructure BSS e Independent BSS.
7) **O lnfrastructure BSS:** Depende do AP para a comunicação entre os clientes. O AP terá como responsabilidade: a autenticação do usuário, a negociação de criptografia e a associação e desassociação do cliente. O AP será o mediador de toda a interação dos clientes *wireless* bem como a sua integração com os dispositivos da rede cabeada. Este sistema é robusto e normalmente implementado no mercado corporativo.
8) **Independent BSS:** Também conhecido como Ad-Hoc ou Peer-to-Peer, não dependerá de elemento central para funcionar. Como não existem intermediário, todos os clientes deverão estar dentro da área de alcance do sinal de todos os outros. Este tipo de BSS não possui a robustez do Infrastructure BSS, *send*o normalmente implementado em redes de residências e outros pequenos ambientes onde o sinal facilmente abrange todos os dispositivos.
9) **Extended Service Set (ESS):** Em diversas situações, a área de abrangência de um conjunto de clientes precisa ser maior do que a potência que um único Access Point pode oferecer. Neste caso, é possível interconectar múltiplos AP através de um meio de distribuição para permitir que usuários tenham serviço

em localidades maiores. Uma rede que seja provida de vários BSS, é denominada de Extended Service Set (ESS).

10) **Basic Service Set Identifier (BSSID):** É um campo de 48 *bit*s, representado em formato hexadecimal. É usado para identificar o BSS: em Infrastructure BSS, é o MAC da interface *wireless* do AP; em Independent BSS, é aleatório.
11) **Service Set Identifier (SSID):** A maioria dos administradores de rede prefere trabalhar com nomes ao invés de endereços em formato hexadecimal. O SSID é um campo alfanumérico que identifica o nome da rede. Este campo pode ter até 32 *bytes*. O SSID identificará todos os BSA que fazem parte de um ESA. Clientes podem ser configurados para utilizar uma rede pelo nome definido no SSID. Desta forma, o roaming do cliente é bastante simples, já que o cliente não precisa saber o BSSID dos AP que o atendem.

Três estados foram definidos pelo padrão IEEE 802.11 para definição da mobilidade do cliente. O último estado, apesar de definido, não é suportado de forma transparente ao usuário. O estado **Ausência de Transição** ocorre quando o cliente está fixo dentro de um BSA. O usuário poderá estar móvel, mas sempre dentro do alcance de um mesmo BSA e atendido pelo mesmo BSS. No estado **Transição entre BSS** o cliente se locomoverá entre BSA dentro do mesmo ESA. Embora esteja na área de cobertura da mesma rede, este cliente estará *send*o atendido por outro BSS e uma reassociação foi feita com o AP do novo BSS. Esse procedimento é transparente e poderá ser feito sem perda de conexão.

Já o estado **Transição entre ESS** ocorrerá quando o cliente se locomover para fora do alcance de qualquer AP desse ESS, e passará a ser atendido por outro ESS. Não existe procedimento transparente para essa mobilidade e o cliente deverá ser reconfigurado para suportar o novo ambiente.

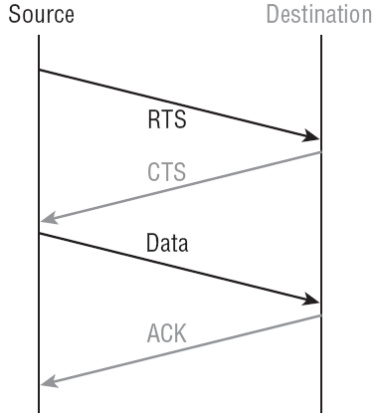
3.4 Métodos de acesso

> Com a Ethernet, uma estação só precisa esperar até o éter ficar inativo para começar a transmitir. Se não receber de volta uma rajada de sinal ruidoso enquanto transmite os primeiros 64 *bytes*, é quase certo que o quadro tenha sido entregue corretamente. No caso das LANs sem fios, esse mecanismo de detecção de colisão não funciona. E vez disso, o 802.11 tenta evitar colisões com o protocolo chamado CSMA com prevenção de colisão, ou CSMA/CA (CSMA with Collision Avoidance). (TANEMBAUM, 2011, p190).

A camada MAC (Media Access Control) é responsável pela padronização do método de acesso ao meio. No ambiente *wireless*, o envio de informação é complexo pela

característica "etérea" do meio de transmissão. Em função disto, existem dois problemas tratados pela camada MAC do IEEE 802.11: a interferência direta de outros dispositivos que transmitem na mesma frequência; a interferência de dispositivos sem fio que geram pacotes compreensíveis ao receptor, porém não destinados à sua rede. Ao contrário da maior parte do conjunto de frequências utilizáveis para se transmitir informações, as faixas utilizadas pelo padrão IEEE 802.11 não são licenciadas. Isto significa que outros dispositivos podem utilizar estas frequências. Como exemplo, pode ser citado o forno de micro-ondas que utiliza a mesma frequência do IEEE 802.11b. Além disto, por se tratar de um sistema de uso irrestrito, se duas empresas fizerem uso da mesma frequência em ambientes muito próximos será possível que uma rede cause interferência na outra. A confirmação do recebimento da informação assegura ao cliente origem que o destino recebeu a informação de forma íntegra. Toda operação no IEEE 802.11 gerará um pacote de confirmação denominado acknowledge (ACK). Isto garante ao transmissor que seu pacote foi recebido pelo destinatário. A figura 14 ilustra o acknowledge – ACK.

Figura 14: – Acknowledge - ACK



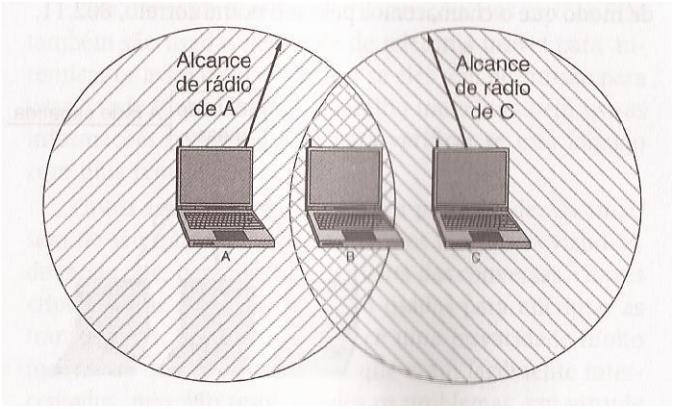
Fonte: Lammle (2007, p. 708)

Para Tanenbaum (2011, p. 44) um dos grandes desafios de um sistema de transmissão sem fio é a identificação de que o barramento já está em uso. A detecção física do uso do barramento nem sempre será suficiente, pois dois clientes podem estar posicionados de tal forma que seus raios de alcance cheguem ao AP, mas não entre si.

Quando um cliente desconhece a existência de outro cliente, dá-se o nome de Clientes Escondidos. Quando isto acontecer, a detecção física sempre acusará barramento ocioso. No exemplo, se os clientes transmitissem simultaneamente, o AP receberia informação incompreensível, uma vez que o sinal de um corrompe o do outro. Para evitar este problema, o padrão IEEE 802.11 faz a detecção da utilização do barramento de forma lógica,

gerando *frame*s de controle (RTS e CTS) que reservam o meio físico. Com esta reserva, um cliente sempre estará ciente do uso do meio físico, mesmo que a camada física não tenha esta visibilidade. A figura 15 ilustra o cliente escondido. O host A não é percebido pelo host C e vice-versa, uma vez que os raios de propagação dos sinais de ambos não estão cobrindo fisicamente um ao outro.

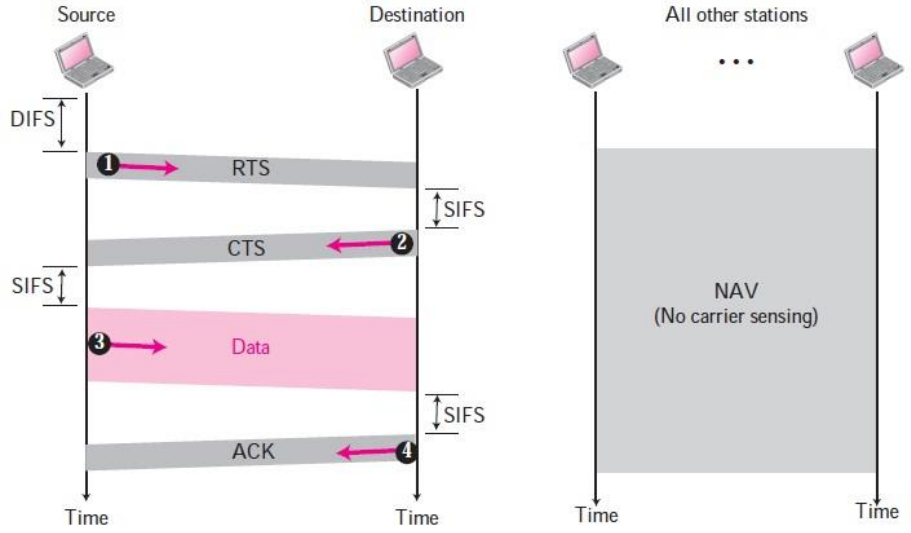
Figura 15 – Cliente Escondido



Fonte: Tanenbaum (2011, p. 44)

O IEEE 802.11 utiliza os sinais RTS (Request To *Send*) e CTS (Clear To *Send*) para evitar colisões e resolver o problema gerado pelos clientes escondidos. Um cliente *wireless* envia um sinal RTS antes de transmitir seus dados. Esse sinal é recebido por todos os elementos da área de alcance. O elemento destino transmitirá um CTS para indicar que recebeu o pacote. Nas Infrastructure BSS, o elemento destino será sempre o Access Point. No sinal CTS tem-se a informação de quem recebeu a autorização de uso do barramento e o tempo alocado. Nesse momento, todos os elementos da rede já saberão que o barramento está ocupado e permanecem em silêncio durante o período de reserva do barramento. O elemento que originou o RTS agora transmite seus *frame*s com a confiança de que outros elementos não interferirão no processo. A cada *frame* recebido, o elemento de destino transmite um ACK de confirmação. Esse processo consumirá recursos e tempo na rede. Por causa disto, é possível fazer um ajuste que define o tamanho mínimo do pacote de dados para justificar um RTS/CTS. Pacotes abaixo deste tamanho serão transmitidos normalmente, sem reserva prévia do barramento. Normalmente esse tamanho mínimo é de 2.347 *bytes*. A figura 16 ilustra o tópico evitando colisões.

Figura 16 – Evitando colisões



Fonte: Forouzan (2010, p. 62)

Segundo Forouzan (2010, p.62) a fim de verificar a disponibilidade do meio, a tecnologia de redes sem fio trabalha com duas funções principais que precisam interagir entre si para garantir a detecção de uso do barramento. São elas:

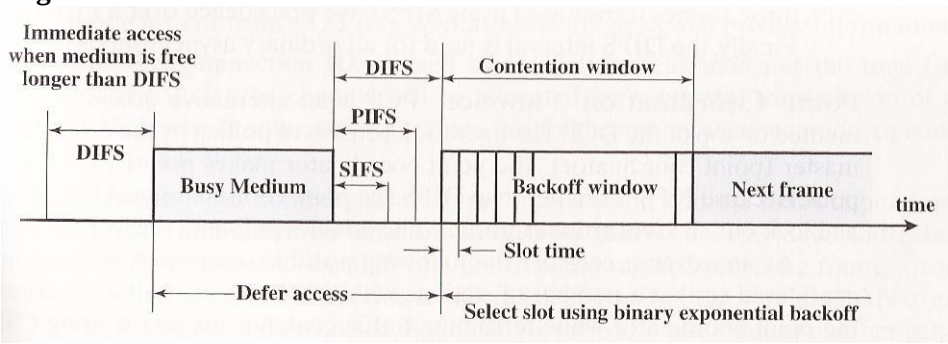
1) **Physical Carrier-Sensing**: É implementado na camada física e depende da mídia e da modulação utilizada. Este mecanismo detecta transmissões de rádio nos *transceivers*. Por causa do problema do Cliente Escondido, este mecanismo não é suficiente para tratar todas as situações.
2) **Virtual Carrier-Sensing:** É implementado na camada lógica através do NAV; os *frames* de controle (RTS, CTS, ACK) indicam o tempo de alocação do barramento no campo NAV destes pacotes. Qualquer cliente, ao receber o *frame* de controle, tem acesso ao valor do NAV e saberá por quanto tempo o barramento estará em uso.

Stallings (2007, p.539) nos ensina que, a fim de garantir que um quadro não se choque com outros e degrade a ambos, o padrão IEEE 802.11 possui diferentes formas de gerar intervalos entre os quadros, a saber:

1) **O DistributedA Inter-*Frame* Space (DIFS):** É o tempo mínimo necessário entre a transmissão de dois *frame*s. Antes de transmitir, qualquer elemento deverá aguardar a liberação do barramento e o intervalo de um DIFS para tentar alocar o barramento para si. Uma vez que intervalo tenha passado, os clientes competem pelo barramento utilizando o sistema aleatório de alocação denominado *contention window*.

2) **O Short Inter-*Frame* Space (SIFS):** É um intervalo de tempo menor que o DIFS e é usado para garantir a transmissão de pacotes em sequência**.** Por ser menor que um DIFS, um elemento que aguarda apenas um SlFS antes de transmitir não dá chance a outro (que está aguardando um DIFS) de interromper uma operação em andamento. Um SIFS também é usado entre transmissões de uma mesma operação**.** A figura 17 ilustra o intervalo entre *frames*.

Figura 17: Intervalo entre *Frame*s



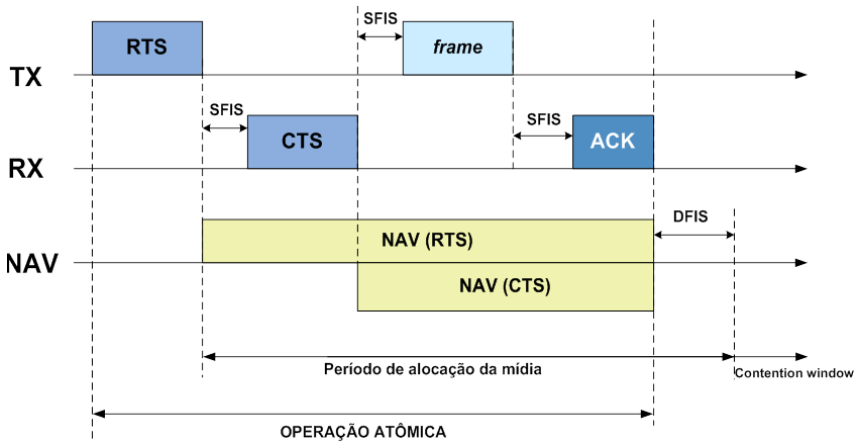
Fonte: Stallings (2007, p. 539)

Tanenbaum (2011, p.191) o NAV é um campo nos cabeçalhos dos *frame*s RTS, CTS e ACK usado para indicar o tempo de alocação do barramento. Este valor é uma estimativa do tempo necessário para se transmitir a informação. Qualquer cliente, que receba, o RTS, saberá que o barramento está em uso pelo período definido. Uma vez que o receptor da informação recebe o RTS, responde com um CTS. O intervalo entre o RTS e o CTS é um SIFS, para garantir que outro cliente não aloque o barramento para si. O valor do NAV, uma vez que tenha sido ajustado pela diferença de tempo entre a transmissão do RTS e do próprio CTS, é incluso neste pacote.

Qualquer cliente que estiver dentro do alcance do transmissor ou do receptor receberá um RTS ou um CTS (ou ambos) e saberá que o barramento está em uso pelo período do NAV. Após mais um SIFS, o transmissor enviará seu *frame*. Os *frames* são transmitidos durante o período de duração do NAV, garantindo que nenhum outro cliente causará colisão. Finalmente, após mais um SIFS, o destinatário da informação gera um ACK, garantindo a recepção do pacote. O NAV do ACK terá o valor igual a zero, sinalizando o término da transmissão. Um DIFS deverá passar-se antes do próximo cliente tentar alocar o barramento. O nome Operação Atômica é dado a uma operação ininterrupta que completa a transmissão de um bloco de informações e estará encerrada quando o elemento de destino confirma a

recepção do bloco. Para pacotes pequenos, a Operação Atômica consiste em apenas dois pacotes: *frame* e ACK. Em transmissões muito longas, onde se tem fragmentação do pacote em múltiplos *frame*s, a Operação Atômica consistirá de todos os *frame*s que compõem o pacote original e seus respectivos ACK. Se houver reserva do barramento, a Operação Atômica será formada inicialmente pelo conjunto de pacotes de controle (RTS e CTS) e mais o *frame* com seu respectivo ACK. Todos os *frame*s que compõem uma Operação Atômica são separados por SIFS (intervalos curtos). Os intervalos entre Operações Atômicas são DIFS. A figura 18 ilustra a detecção do canal virtual do CSMA/CA.

Figura 18 – Canal virtual com CSMA/CA

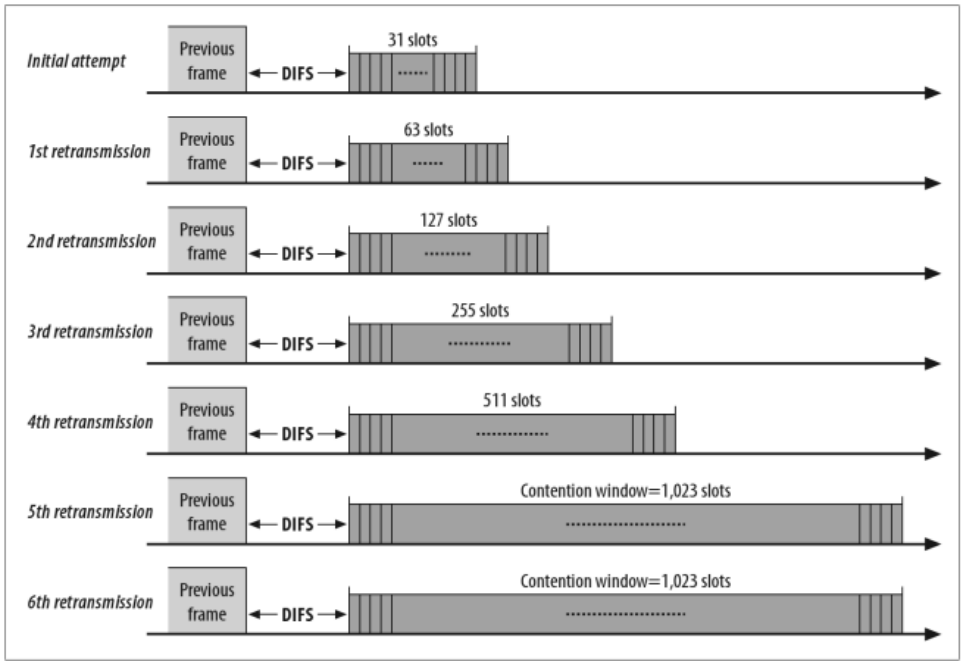


Fonte: Tanenbaum (2011, p. 192)

Após a finalização de um processo de transmissão e do intervalo DIFS, os clientes com dados a serem enviados precisarão iniciar suas transmissões. O período que sucede o DIFS é chamado contention ou *backoff window* e é dividido em partes iguais chamadas *slot*s. O tempo de duração de cada *slot* dependerá da mídia, onde quanto maior for a velocidade, menor o intervalo de tempo de cada *slot*. Cada cliente selecionará randomicamente um valor de *slot* e aguardará a expiração deste tempo para poder acessar o meio de transmissão. Como um *slot* é uma medida de tempo, os clientes aguardarão o intervalo aleatoriamente selecionado para tentar transmitir. Se seu tempo expirar e nenhum outro cliente estiver transmitindo, o cliente gerará um RTS e o processo de transmissão se iniciará. Inicialmente, o número de *slot*s é pequeno, para que os clientes não tenham que aguardar muito antes de transmitir. Entretanto, caso o número de colisões de RTS for muito elevado, a cada nova colisão o número de *slot*s dobrará e em poucas iterações o número de *slot*s se tomará muito grande. O número só voltará ao normal quando um *frame* for transmitido com sucesso ou se o número de tentativas de transmissão for excedido e o pacote descartado.

Afirma Matthew (2005, p.44) que utilizando os números da camada física DSSS (Direct-Sequence Spread-Spectrum) poder-se-á observar que a cada nova retransmissão o número de *slot*s dobrará até atingir o limite da DS (Distribution System), de 1023 *slot*s. A *contention window* permanecerá em seu valor máximo até um *frame* conseguir ser transmitido com sucesso ou quando um *frame* for descartado. A figura 19 ilustra a *Contention window*s/*Backoff Slot*s.

Figura 19: *Contention window*s/Backoff *Slot*s



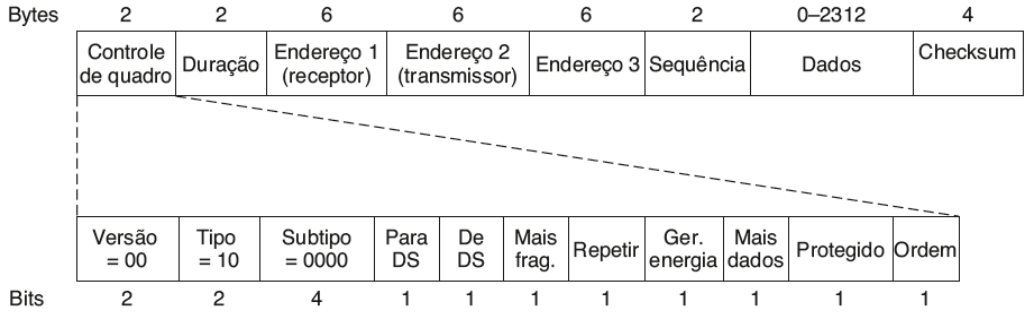
Fonte: Matthew (2005, p. 44).

A estrutura do *frame* padrão IEEE 802.11 possui características que lhe são peculiares em função dos desafios demandados pelas redes *wireless*. Dentre estas características está a existência de quatro campos de endereços no cabeçalho, a transferência do preâmbulo para a camada física e os campos tipo/comprimento a serem descritos na área de dados do *frame*. O primeiro campo de cada *frame* é denominado Controle do Quadro (*frame*) e possui dois *bytes* de tamanho. Os *bit*s deste campo possuem os seguintes significados:

1) **Versão:** Identifica a versão do protocolo IEEE 802.11 utilizada; atualmente existe somente uma versão e por isso o valor deste campo é sempre zero, outros valores serão utilizados quando novas versões forem desenvolvidas.
2) **Tipo e Subtipo:** Indicam o tipo do *frame* transmitido, que poderá ser classificado em três grupos: controle de *frame*, gerência de *frame* e dados do *frame*; para cada tipo de *frame* há os respectivos subtipos.

3) **Para DS e De DS:** O termo DS (Distribution System) identifica o meio de distribuição; como visto anteriormente, o DS poder ser cabeado ou ser um sistema sem fio. O *bit* Para DS será 0 se o *frame* é destinado à rede sem fio e 1 se for destinada a rede cabeada; já o *bit* DeDS será 0 se o *frame* vem de um cliente da *Wireless* LAN e 1 se vem de um cliente da rede cabeada.
4) **Mais Fragmentos:** Quando for necessário que um *frame* sofra fragmentação, o último fragmento será marcado como 0, indicando que não haverá mais fragmentos a serem recebidos; todos os outros fragmentos são transmitidos com este *bit* igual a 1.
5) **Repetir:** Indica que se trata da retransmissão de um *frame*.
6) **Gerência de Energia:** Este *bit* indica se o transmissor entrará em modo *power save* durante o tempo de transmissão do *frame*; o *bit* 1 indica que o transmissor entrará em *power save* e o *bit* 0 indica que não; esta função existe porque as redes sem fio normalmente são utilizadas por *notebooks* e outros dispositivos móveis, que normalmente podem trabalhar com a energia de baterias.
7) **Mais Dados:** Para atender a clientes no modo *power save*, os AP's poderão armazenar temporariamente os *frame*s recebidos do DS e destinados a esses clientes; assim, os AP marcam este *bit* como 1 para os *frame*s endereçados a clientes em *stand-by*.
8) **Protegido:** Os dados transmitidos em uma rede sem fio são naturalmente mais suscetíveis de serem capturados do que em redes cabeadas; para prover segurança na transmissão, um conjunto de métodos de criptografia denominados WEP (Wired Equivalent Privacy) foram criados; quando um *frame* é manipulado pelo WEP, este *bit* é marcado com 1. A figura 20 ilustra o formado do *frame* IEEE 802.11 em especial o campo Controle de Quadro.

Figura 20: Formato do *frame* IEEE 802.11



Fonte: Tanenbaum (2011, p. 194)

Tanenbaum (2011, p.194), o *frame* IEEE 802.11 poderá conter até três campos de endereços. Os campos de endereços são numerados e podem ter diferentes conteúdos que dependerão do tipo de *frame* e do caminho que o *frame* percorrerá entre o cliente origem e destino. Os conteúdos dos campos de endereços têm estreita relação com os campos Para DS e De DS do cabeçalho do *frame* IEEE 802.11. Estes campos têm o mesmo formato dos endereços MAC das redes locais tradicionais, com tamanho de 48 *bit*s. Vejamos a descrição dos possíveis conteúdos dos campos de endereços:

- **Origem:** endereço MAC do cliente que cria e envia o *frame*; apenas um cliente pode ser a origem do *frame*.

1) **Destino:** destino final do *frame*; endereço MAC do cliente para a qual o *frame* é destinado e onde é processado para ser entregue aos protocolos das camadas superiores.
2) **Transmissor:** endereço do elemento intermediário que transmite o *frame* para o seu destino final; em outras palavras, este é o endereço MAC do AP.
3) **Receptor:** Endereço do elemento da rede *wireless* que deve receber e processar o *frame*; caso este elemento seja um cliente, então o endereço do receptor coincide com o endereço destino (DA); para *frame*s cujo destino final é um cliente localizado na rede cabeada conectada à *Wireless* LAN através de um AP, este será o endereço MAC do AP.

- **BSSID (Basic Service Set Identifier):** Como diversos BSS podem se sobrepor numa mesma área de cobertura, é atribuído um identificador BSSID único para cada BSS; o BSSID é o endereço MAC do AP.

Caso um *frame* seja originado no cliente *wireless* e seja destinado ao cliente da rede cabeada, então os *bit*s Para DS e De DS do campo controle de quadro serão iguais a 1 e

0, respectivamente. *Send*o assim, os campos de endereço do cabeçalho do *frame* serão preenchidos da seguinte forma:

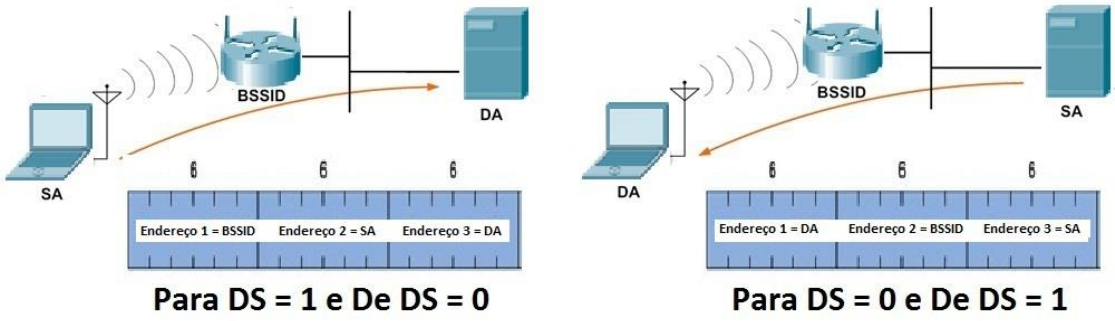
1) **Endereço 1:** endereço MAC do AP (BSSID);
2) **Endereço 2:** endereço MAC do cliente origem (SA);
3) **Endereço 3:** endereço MAC do cliente destino (DA);

Um *frame* originado no cliente da rede cabeada e destinado ao cliente *wireless*, os *bit*s Para DS e De DS do campo controle de quadro serão 0 e 1. *Send*o assim, os campos de endereço do cabeçalho do *frame* serão preenchidos da seguinte forma:

- **Endereço 1:** endereço MAC do cliente destino (DA).
- **Endereço 2:** endereço MAC do AP (BSSID).
- **Endereço 3:** endereço MAC do cliente origem (SA).

A figura 21 ilustra as duas situações de preenchimento dos campos de endereço, bem como a flags Para DS e De DS acima descritas para o envio de *frame*s.

Figura 21: Flags ToDS / FromDS do *frame* IEEE 802.11



Fonte: Elaborado pelo autor (2013).

4 EXPERIÊNCIAS DE LABORATÓRIO WIFI

> Embora a estatística seja importante, muitos trabalhos a deixam em segundo plano inferindo conclusões tomando como base apenas dados como média e desvio padrão. Esses dados, apesar de serem essenciais, não são suficientes para realizar a comparação, por exemplo, entre dois algoritmos. Nesse caso é necessário, no mínimo, a formulação de uma hipótese e sua verificação através de um teste-T. Ampliando esse escopo, quando a análise estende-se para 3 ou mais algoritmos deve-se utilizar a análise de variância (ANOVA) como ferramenta. (CORTÊS, 2011).

Tem-se como Cortês (2011) que, embora o teste realizado pelo ANOVA possa resultar na identificação das variações entre médias, culminando na determinação em que pese a existência ou não destas diferenças, não será possível afirmar onde a diferença será incidente. Nesses casos, a observação dessas diferenças poderá ser averiguada através do teste de Tukey com certo grau de significância.

Desta forma, o objetivo de todas as experiências laboratoriais foram prover amostras para análise estatística a fim de subsidiar as conclusões finais desta pesquisa. Para a captura das amostras, utilizou-se o *software* cliente-servidor desenvolvido, bem como o *Software* inSSIDer from Metageek version 3, para avaliação das redes sem fio e suas potencias, nas imediações dos experimentos. As amostras capturadas foram analisadas utilizando-se as ferramentas Excel da Microsoft e o *Software* ASSISTAT (ASSIS, 2013). Do *software* Excel, fez-se o uso da media, desvio padrão, gráfico e ANOVA. Do *software* Assistat, fez-se uso do teste de Tukey.

4.1 Plano de experimento

A fim de organizar os experimentos que foram realizados, optou-se por criar um plano de contas das experiências. Desta forma, ter-se-ia uma relação de todos os ensaios desejados, constando de quais equipamentos estariam envolvidos, o firmware, ganho e potência dos mesmos, além do ambiente com interferência direta ou não. A tabela 3 relata o Plano de Experimentos.

Tabela 3: Plano de Experimentos

CONTA	DESCRIÇÃO
1.0.0	**SEM Interferência Direta**
1.1.0	**Access Point – Router Linksys WRT54G V8**
1.1.1	Potência padrão do equipamento 18dBm
1.1.2	Potência aumentada via configuração para 20dBm
1.2.0	**Access Point – 3COM WX1200 + AP2750**
1.2.1	Potência padrão em único AP 18dBm
1.2.2	Potência aumentada em duplo AP 2 x 18dBm
2.0.0	**COM Interferência Direta**
2.1.0	**Access Point – Router Linksys WRT45G V8**
2.1.1	Potência padrão do equipamento 18dBm
2.1.2	Potência aumentada via configuração para 20dBm
2.2.0	**Access Point – 3COM WX1200 + AP2750**
2.2.1	Potência padrão em único AP 18dBm
2.2.2	Potência aumentada em duplo AP 2 x 18dBm

Fonte: Elaborado pelo autor (2013).

Tabela 4: Detalhamento do Plano de Experimentos

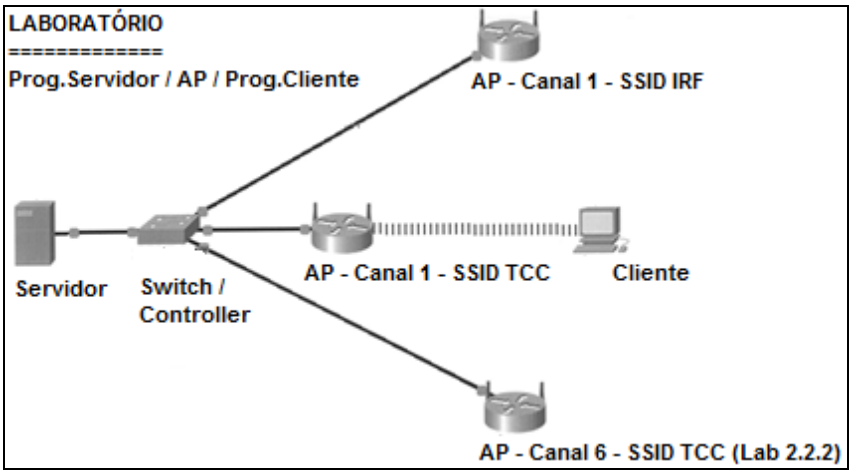
It.	Data	Hora	Conta	Equipamento	Transmissão		Firmware	Interferência Direta
					dBm	mW		
01	15.04.13	16h	1.1.1	Router Linksys WRT54G V8	18	64	dd-wrt.v24_micro_generic.bin	-
02	15.04.13	19h	1.1.2	Router Linksys WRT54G V8	20	100	dd-wrt.v24_micro_generic.bin	-
03	16.04.13	11h	1.2.1	3COM WX1200 + AP2750	18	64	wb060601.rel	-
04	16.04.13	19h	1.2.2	3COM WX1200 + 2x AP2750	2x 18	2x 64	wb060601.rel	-
05	17.04.13	08h	2.1.1	Router Linksys WRT54G V8	18	64	dd-wrt.v24_micro_generic.bin	0dBm, 1mW
06	17.04.13	11h	2.1.2	Router Linksys WRT54G V8	20	100	dd-wrt.v24_micro_generic.bin	0dBm, 1mW
07	18.04.13	12h	2.2.1	3COM WX1200 + AP2750	18	64	wb060601.rel	0dBm, 1mW
08	18.04.13	18h	2.2.2	3COM WX1200 + 2x AP2750	2x 18	2x 64	wb060601.rel	0dBm, 1mW

Fonte: Elaborado pelo autor (2013).

4.2 Topologia do laboratório e equipamentos

A figura 22 retrata a topologia que foi utilizada em laboratório para verificação dos tempos de transmissão dos segmentos TCP. Embora a topologia possuísse três AP disponíveis, o acesso à rede com cabo será efetuado exclusivamente por um AP, quando não se tratar dos experimentos 1.2.2 e 2.2.2. Fixou-se a topologia nesses moldes, para configurar a emissão de sinais de rádio na mesma frequência do AP cliente, quando os laboratórios exigiram que as transmissões tivessem sido realizadas com a inserção de interferência de rádio frequência (IRF). Já os laboratórios 1.2.2 e 2.2.2 utilizaram dois AP simultâneos para atendimento ao cliente, ficando o terceiro AP para a injeção de sinal de interferência, quando necessário.

Figura 22: Topologia do Laboratório



Fonte: Elaborado pelo autor (2013).

Na tabela 5, foram relacionados os equipamentos existentes na topologia, as quantidades, função, descrições e relação com os experimentos (Conta).

Tabela 5: Equipamentos utilizados nos laboratórios

It	Equipamento	Qtd	Descrição	Conta	Função
01	Notebook Server-PT	1	Intel® Pentium® Dual CPU T2330 @ 1.60GHz, com 2GB de RAM usando sistema operacional de 32 bits Win7	Todas	Executar o software servidor na rede cabeada
02	Switch Cisco 2960-24TT	1	24 portas 10/100 +2 1000GB + IOS Lan Base, 16Gbps, 6.5 mpps, Automatic QoS, IPv6, SSH, RADIUS, Layer 2	1.1.1 1.1.2 2.1.1 2.1.2	Possibilitar conexão do AP com a rede cabeada
03	Controlador 3COM WX1200	1	6 portas PoE 10/100 +2 uplink 10/100, gerencia até 12 AP	1.2.1 1.2.2 2.2.1 2.2.2	
04	Router Linksys WRT54G V8	1	Padrões: IEEE 802.11g, 802.11b, uma porta 10/100 potencia de transmissão: 802.11g: 0-23 dBm	1.1.1 1.1.2 2.1.1 2.1.2	Prover acesso sem fio a estrutura da rede cabeada
05	AP 3COM 2750	1	Padrões: IEEE 802.11g, 802.11b, 802.11a, uma porta 10/100, potência de transmissão 802.11g: 18dBm	1.2.1 1.2.2 2.2.1 2.2.2	
06	Notebook PC-PT Cliente	1	AMD E-350 processor 1.60GHz, com 4GB de RAM com S.O. Win7 64bits	Todas	Executar o software cliente na rede sem fio.

Fonte: Elaborado pelo autor (2013).

4.3 Algoritmos Desenvolvidos

Com base na tabela 6, expressam-se os algoritmos desenvolvidos e que serviram de base para o desenvolvimento em C++ das ferramentas cliente e servidor.

Segundo Comer (2006, p.240), um servidor é um programa de computador que irá proporcionar algum serviço no âm*bit*o da rede. Um servidor recebe e aceita uma requisição, processa um conjunto de informações e devolve uma resposta ao solicitante que damos o nome de cliente. Um programa será dito como cliente, quando envia uma solicitação para um servidor e aguarda por uma resposta do mesmo.

Os algoritmos desenvolvidos especificamente para apurar os tempos de transmissão dos segmentos TCP, foram implementados a partir de versão básica de comunicação cliente-servidor através da utilização de sockets. O algoritmo do servidor basicamente irá receber cada segmento enviado pelo cliente e como padrão do TCP, devolver ao cliente um ACK de confirmação pelo segmento recebido.

A base para E/S de rede, para Comer (2006, p.247), circunvizinha à abstração conhecida como socket, que é uma parte do kernel do sistema operacional. Socket vem a ser um mecanismo que oferece um ponto para comunicação externa ao host local. Um programa aplicativo cria um socket quando necessário e o sistema operacional retorna um valor inteiro, onde o aplicativo fará referência a este valor quando desejar acessar aos recursos do socket. No caso de os sockets corresponderem a uma conexão TCP, as duas extremidades da conexão: servidor e cliente precisarão ser delineadas.

Em nosso caso específico, todo o controle sobre a transmissão dos dados será realizada pelo algoritmo cliente, já que o algoritmo servidor providenciará o eco da solicitação cliente através da emissão do ACK natural ao recebimento e confirmação do TCP. O aplicativo cliente gerou os segmentos TCP com tamanho de 1.450 *bytes* preenchidos com 145 sequências de “xxxxxxxxx1”, que foram encapsulados em pacotes IP e enviados ao servidor que retornou o ACK. Ao receber o ACK do servidor, o cliente registrou os tempos de transmissão destes segmentos TCP para as grandezas de 1MB, 3MB e 5MB em um conjunto de 31 (trinta e uma) repetições por grandeza, com vistas a subsidiar as comparações estatísticas.

As grandezas de 1MB, 3MB e 5MB foram escolhidas nesses valores, pois representam uma estimativa do envio de dados pelo cliente, que utiliza dispositivos móveis para acessar um determinado servidor. Em três grandezas distintas, para que fosse idealizando três perfis de usuários.

Tabela 6: Algoritmos: Servidor e Cliente

	Servidor		Cliente
1	message ← buffer_size 1450	1	message ← buffer_size(145, "xxxxxxxxx1")
2	local_IP ← get_IP()	2	qtd_megas ← 0
3	local_port ← 49151	3	local_socket ← 0
4	local_socket ← local_IP + local_port	4	remote_socket ← 0
5	remote_socket ← 0	5	file ← open ("wifi.txt")
6	listen (local_socket, remote_socket)	6	for (laco0=1; laco0 ≤ 3; laco0++)
7	do message not= "fim"	7	do case (laco0)
8	accept (local_socket, remote_socket)	8	case 1: qtd_megas = 690
9	print "Aguardando conexão"	9	case 2: qtd_megas = 2069
10	receives (message)	10	case 3: qtd_megas = 3449
11	close_socket (remote_socket)	11	end case
12	end do	12	for (laco1=1; laco1 ≤ 31, laco1++)
13	close (local_socket, remote_socket)	13	t_ini ← clock()
		14	for (laco2=1; laco2 ≤ qtd_megas, laco2++
		15	remote_socket
		16	connect (remote_socket)
		17	send (remote_socket, message)
		18	next
		19	t_fim ← clock()
		20	write ("wifi.txt", t_fim – t_ini)
		21	next
		22	next
		23	send (remote_socket, "fim")
		24	close ("wifi.txt")

Fonte: Elaborado pelo autor (2013).

❖ **Algoritmo Servidor**

A principal função deste algoritmo é o *loop* existente (laço 7-12). Esta rotina ficou em execução recebendo os pacotes enviados pelo cliente e retornando o ACK de confirmação desta recepção. Este procedimento de confirmação é intrínseco a função *receives*() para o TCP e ficou em execução até que a variável message possuísse o valor "fim".

❖ **Algoritmo Cliente**

Este algoritmo fez o envio de grandezas na ordem de 1MB, 3MB e 5MB ao servidor (laço 6-22), em repetições de trinta e uma vezes cada (laço 12-21), fracionados em pacotes de 1.450 *bytes*, tantas às vezes quantas as necessárias até completar cada uma das três grandezas (laço 14-18). Antes de entrar nesse último laço, foi tomado o tempo inicial e, após sair deste laço, tomado o tempo final para calculo do tempo de transmissão que foi armazenado no arquivo wifi.txt. O uso de segmentos de tamanho fixo de 1.450 *bytes* garantiu que o TCP gerasse um único pacote IP para encapsulamento em meios que utilizam protocolos CSMA/CA e CSMA/CD.

5 CONCLUSÕES

Segundo TANENBAUM (2011, p.367), deve-se obter um tamanho de amostra grande o bastante, medindo o tempo necessário para o envio de segmentos, pois quando usamos uma grande amostra, torna-se possível reduzir a variação na medição da média e do desvio-padrão.

> Quando uma rede tem baixo desempenho, em geral os usuários reclamam com seus administradores, exigindo melhorias. Para melhorar o desempenho, os operadores devem primeiro descobrir exatamente o que está acontecendo. Para isso, os operadores precisarão fazer medições. O tipo mais elementar de medição consiste em ativar um timer ao iniciar um procedimento e usá-lo com a finalidade de verificar o tempo necessário para concluir essa atividade. A medição dos parâmetros e do desempenho das redes tem muitas armadilhas potenciais. Qualquer tentativa sistemática de medir o desempenho das redes deve ter o cuidado de evitar essas armadilhas. (TANENBAUM, 2011, p.367).

Seguindo o princípio acima descrito, para os experimentos realizados, a menor amostra em milissegundos capturada foi o tempo de transmissão de 1MB de informações em um enlace de rede sem fio de acordo com a Tabela B.1.1 do anexo II, experimento 111, repetido trinta e uma vezes.

Ainda com fundamento em Tanenbaum (2011, p.367), a totalidade das medições deve ser repetida em dias e horários diversos com o objetivo de verificar o efeito de diferentes cargas do sistema sobre a quantidade a ser medida.

Observando-se a recomendação supracitada, a totalidade dos experimentos foi realizada de acordo com a Tabela 4, onde as amostras foram medidas duas a duas, em quatro dias consecutivos em horários alternados.

Para Tanenbaum (2011, p.367), a repetição de uma medição poderá ter como retorno uma resposta extremamente rápida, caso os protocolos se utilizem de mecanismos de *caching*. Exemplos desta utilização poderão ser facilmente verificados para buscas realizadas em páginas da *WEB* ou pesquisa de nome em servidor DNS, pois, nestes casos, a primeira chamada envolverá a rede, depois a consulta será feita em *cache* local.

Nos experimentos realizados neste trabalho, foram desenvolvidos os *software*s cliente e servidor constantes na Tabela 6, onde se primou que o host cliente faça o envio do segmento TCP (CLIENTE linha 17, *send*) e aguarde pelo ACK de confirmação de recebimento (SERVIDOR linha 10, *receives*). Portanto, não se trata do envio de uma solicitação de consulta, que poderá ter a resposta de um *cache*, mas da transmissão de segmentos TCP, que exigem o retorno de uma confirmação ACK para cada segmento transmitido, forçando a efetiva verificação dos tempos de transmissão entre o cliente e o

servidor usando efetivamente os recursos da rede.

A medição será mais oportuna, se for realizada em uma rede ociosa, onde a carga de trabalho será toda gerarão pelo experimento. As redes sem fio geram um desafio a mais, pois não raramente ela estará sujeita a diversas fontes de interferências, quer sejam outras redes sem fio operando nas mesmas frequências, que sejam aparelhos domésticos quer se utilizam ou interferem na frequência 2.4GHz. (TANENBAUM, 2011, p.367).

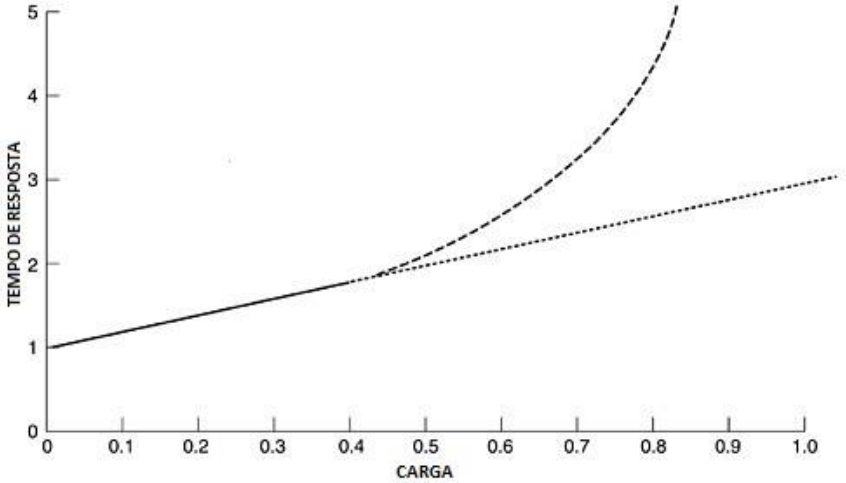
Esta consideração, foi observada da seguinte forma: o laboratório de rede sem fio, foi criado para o fim específico dos experimentos aqui realizados, conforme figura 22 da topologia do laboratório. Quanto ao monitoramento das redes sem fio nas imediações desta topologia, foi verificado através das tabelas A.1 a A.8 do Anexo A, que trata da inspeção do ambiente da rede sem fio. Nos períodos de realização das medições, manteve-se no raio de 100m, sob garantia sobre a não utilização de equipamentos que funcionassem ou interferissem na frequência 2.4GHz.

Segundo Tanenbaum (2011, p.368), quando se usar um *timer* para aferir um evento que dura menos de um milissegundo, deveremos ter muito cuidado nesta medição, pois os *clockss* por vezes nem sempre serão tão acurados quanto o retorno da precisão da medição realizada. Esta preocupação será superada se a medição for fruto da média de uma grande quantidade de repetições.

Esta ponderação foi observada, quando da elaboração do algoritmo constante na Tabela 6, pois o início da contagem do tempo de uma mediação ocorre na linha 13 com a variável “t_ini” recebendo o valor do *clocks*. Neste instante é iniciado um *loop* que transmitirá 1MB, 3MB ou 5MB (linhas de 14 a 18), para que ao final deste *loop* seja feita novamente a captura do *clocks* e armazenada na variável “t_fim”. Desta maneira, será evitada a leitura do *timer* em intervalos de tempos muito curtos.

Ainda com apóio em Tanenbaum (2011, p.368), deveremos ter cuidado com os efeitos de disputa que tendem a se tornar mais evidentes em carga alta. A figura 23, retrata um exemplo de que, em sobrecarga, à medida que a carga aumentar, o efetivo tempo de resposta tenderá a se mostrar mais similar ao gráfico da linha tracejada, do que o gráfico da linha pontilhada.

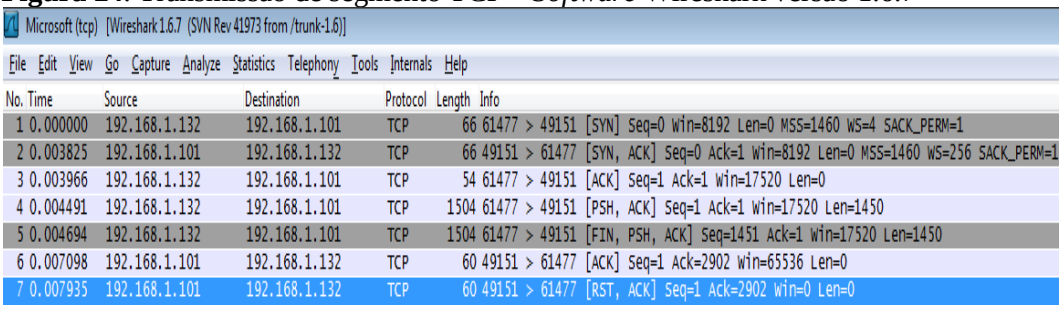
Figura 23: Resposta em função da carga



Fonte: Tanenbaum (2011, p. 368).

Em atenção a esta ponderação, o *software* cliente foi desenvolvido para que cada segmento tivesse um tamanho fixo de 1.450 *bytes* garantido o envio do mesmo em um único segmento TCP. Para cada segmento, primeiro teremos o estabelecimento da conexão, o envio efetivo do segmento e depois a finalização da conexão. Passemos a analisar a figura 24 que ilustra este procedimento: conforme discorrido, toda transmissão de segmentos TCP é precedida pelo *handshake* triplo demonstrado nas linhas 1, 2 e 3; após a origem e o destino concordarem com os termos da conexão, haverá o efetivo envio do segmento conforme a linha 4. Uma vez que o segmento foi transmitido, a conexão será finalizada pela linhas 5, 6 e 7. Tanto o *handshake* triplo quanto o término das conexões poderão ser graficamente revistos respectivamente nas figuras 6 e 7.

Figura 24: Transmissão de segmento TCP – *Software* Wireshark versão 1.6.7



Fonte: Elaborado pelo autor (2013).

Observadas as considerações para medição do desempenho da rede, passaremos às conclusões, tendo-se por base os experimentos realizados.

De acordo com o Anexo B, podemos verificar com base na estatística dos experimentos, onde o teste de Tukey aplicado às amostras capturadas encontrou as médias e

as agregou estatisticamente, que estas médias seguidas pela mesma letra não diferem entre si, ao nível de 5% de probabilidade.

Os experimentos foram associados em quatro grupos de forma que se obtiveram as conclusões deste trabalho:

- **Grupo I**: Experimentos SEM Interferência Direta

Tabela 7: Comparativo entre laboratórios sem interferência direta

Experimentos SEM Interferência Direta								
Bytes	Laboratórios: Média dos Tempos de Transmissão e Teste de Tukey							
	111		112		121		122	
	Média	Tukey	Média	Tukey	Média	Tukey	Média	Tukey
1MB	34.126	b	34.258	b	48.274	a	43.173	a
3MB	103.601	c	103.357	c	113.038	b	129.566	a
5MB	178.054	c	176.640	c	184.515	b	204.310	a
Observação: Média dos Tempos de transmissão em milissegundos								

Fonte: Anexo B

Resultados observados da tabela 7:

I. Os AP individuais apresentam melhor desempenho que os controladores;
II. Elevando-se a potência do AP individual, não melhorará o desempenho do AP.
III. O controlador com um único AP possui melhor desempenho em relação ao controlador com dois AP's, para transmissões igual ou superior a 3MB.

- **Grupo II: Experimentos COM Interferência Direta.**

Tabela 8: Comparativo entre laboratórios com interferência direta

Experimentos COM Interferência Direta								
Bytes	Laboratórios: Média dos Tempos de Transmissão e Teste de Tukey							
	211		212		221		222	
	Média	Tukey	Média	Tukey	Média	Tukey	Média	Tukey
1MB	51.117	a	33.217	b	36.878	b	35.965	b
3MB	143.997	a	101.105	b	110.646	b	107.997	b
5MB	187.460	a	170.237	b	183.695	a	184.367	a
Observação: Média dos Tempos de transmissão em milissegundos								

Fonte: (Anexo B).

Resultados observados da tabela 8:

I. Os controladores apresentam melhor desempenho que os AP's individuais sem elevação de potência para transmissões até 3MB;

II. Elevando-se a potência do AP individual, iguala-se seu desempenho ao controlador para transmissões até 3MB;

III. À proporção que a quantidade de dados transmitidos se elevar, o AP individual com elevação de potência tenderá a obter o melhor desempenho;

IV. O controlador com dois AP's possui melhor desempenho em relação ao controlador com um AP.

❖ **Grupo III: Experimentos AP (sem e com IRF)**

Tabela 9: Comparativo entre laboratórios AP (sem e com IRF)

Experimentos AP (sem e com Interferência Direta)								
Bytes	Laboratórios: Média dos Tempos de Transmissão e Teste de Tukey							
	111		112		211		212	
	Média	Tukey	Média	Tukey	Média	Tukey	Média	Tukey
1MB	34.126	b	34.258	b	51.117	a	33.217	b
3MB	103.601	b	103.357	b	143.997	a	101.105	b
5MB	178.054	b	176.640	b	187.460	a	170.237	c
Observação: Média dos Tempos de transmissão em milissegundos								

Fonte: Anexo B

Resultados observados da tabela 9:

I. Um AP individual com elevação de potencia em ambiente com interferência, iguala-se estatisticamente aos AP's individuais em ambiente sem interferência;

II. A medida que a quantidade de dados transmitidos se eleva, o AP individual com elevação de potencia, tende a obter o melhor desempenho.

❖ **Grupo IV: Experimentos Controlador (sem e com IRF)**

Tabela 10: Comparativo entre laboratórios Controlador (sem e com IRF)

Experimentos Controlador (sem e com Interferência Direta)								
Bytes	Laboratórios: Média dos Tempos de Transmissão e Teste de Tukey							
	121		122		221		222	
	Média	Tukey	Média	Tukey	Média	Tukey	Média	Tukey
1MB	48.274	a	43.173	ab	36.878	b	35.965	b
3MB	113.038	b	129.566	a	110.646	b	107.997	b
5MB	184.515	b	204.310	a	183.695	b	184.367	b
Observação: Média dos Tempos de transmissão em milissegundos								

Fonte: Anexo B

Resultados observados da tabela 10:

I. Para transmissões com até 1MB, os controladores demonstraram melhor desempenho em ambiente com interferência.

II. Para transmissões acima de 1MB, somente se obterá o recurso do 2º AP do controlador, em ambientes com interferência.

a. Recomendações para uso de equipamentos em redes sem fio

Como largamente demonstrado através dos experimentos realizados, para a utilização em redes sem fio, deve-se efetuar:

I. A utilização de AP individual com potência de 18dBm – 64mW, quando a área de cobertura possuir baixa ou desprezível interferência de outras redes sem fio.
II. A utilização de controladores gerenciando um mínimo de dois AP com potência de 18dBm – 64mW, quando a área de cobertura estiver sujeita à interferência não desprezível de outras redes sem fio.

Considerando-se que a frequência 2.4GHz é de uso livre, mesmo que tenhamos ambientes com baixa interferência de rádio frequência, nada impedirá que, em curto espaço de tempo, venha a se ter poluição na faixa de frequência em questão, decorrente da implantação de outras redes sem fio nas imediações da área de cobertura. Desta maneira, recomenda-se o uso de controladores, ainda que a área a ser atendida não possua considerável interferência de outras redes sem fio.

REFERENCIAS

ASSIS, F. S. S. **Software ASSISTAT versão 7.6 beta– DEAG-CTRN-UFCG**, Campina Grande-PB, 2013. Disponível em: <http://www.assistat.com>. Acesso em: 15 maio 2013.

BAKRE A. **Indirect TCP for Mobile Hosts** 1994. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.125.5511>. Acesso em: 08 mar. 2013.

CARROLL, B. J. **CCNA Wireless Official Exam Certification Guide.** 2. ed. Indianapolis, USA: Cisco Press, 2009.

COMER D. **Interligação de redes com TCP/IP**. v. 1, 5. ed. Rio de Janeiro: Elsevir, 2006.

CORTES, O.A.C. **Aplicando Algoritmos Genéticos Real-Coded no Refinamento de Funções de Pertinência Fuzzy**: uma analise estatística. Fortaleza, 2011. X Congresso Brasileiro de Inteligência Computacional (CBIC'2011).

FILIPPETTI, M. A. **CCNA 4.1 Guia Completo de Estudo**. 2. ed. Florianopolis: Visual Books, 2009.

FLOYD S. **TCP and Explicit Congestion Notification**. 1994. Disponível em: Disponível em: <http:// citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.216.7208>. Acesso em: 28 mar. 2013.

FOROUZAN B.A. **TCP/IP Protocol Suite**. 4. ed. New York, USA: McGraw-Hill, 2010.

GOFF, T.; MORONSKI J., PHATAK, D. S.; GUPTA, V. **Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments**, Trabalho apresentado ao IEEE Infocom, Março, 2000.

KUROSE, J.F. **Redes de computadores e a Internet: uma abordagem top-down.** 5. ed. São Paulo: Pearson Education do Brasil, 2006.

LAMMLE, T. **CCNA Cisco Certified Network Associate Study Guide**. 6. ed. Indianapolis, USA: Wiley Publishing, Inc. 2007.

MASCOLO S. **TCP Westwood**: Bandwidth estimation for enhanced transport over wireless links 2001. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.115.4540>. Acesso em: 26 fev. 2013

MATTHEW S.G. **802.11 Wireless Networks**: the definitive guide, 2. ed. USA: OREILLY & ASSOC, 2005.

MCQUERY, S. **Interconectando Cisco Network Devices**. 1 ed. São Paulo: Alta Books, 2002.

ODOM W. **CCENT/CCNA ICND 1 Guia Oficial de Certificação do Exame**. 2. ed. Rio de Janeiro: Alta Books, 2008.

PATEL M. **TCP over Wireless Networks: Issues, Challenges and Survey of Solutions**. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.6003>. Acesso em: 01 abr.2001.

SANTANA, A. A. **Proposta para Otimização de desempenho do protocolo TCP em redes wireless 802.11.** São Paulo, 2003. Tese de Mestrado apresentada na Escola Politécnica da Universidade de São Paulo.

STALLINGS, W. **Data and Computer Comunications.** 8. ed. New Jersey, USA: Pearson, 2007.

STALLINGS, W. **Data and Computer Comunications.** 8. ed. New Jersey, USA: Pearson, 2007

TANENBAUM, A. S. **Redes de computadores**. 5. ed. São Paulo: Pearson Education, 2011.

VANGALA S., LABRADOR M.A.**Performance of TCP over Wireless Networks with the Snoop Protocol** Disponível em: <http://citeseer.ist.psu.edu/viewdoc/summary ?doi=10.1.1.98.997>. Acesso em: 14 jan. 2013.

APENDICE A

Inspeção do Ambiente dos Experimentos

A.1 – Inspeção do ambiente: Wi-Fi 2.4GHz – AP - SEM IRF Direta

Redes 2.4GHz | **SSID TCC**

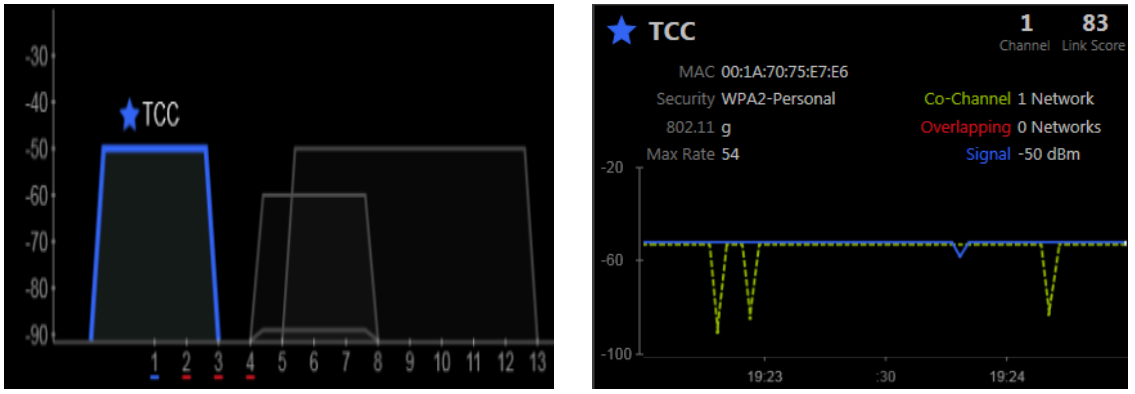


Tabela A.1 – Ambiente do experimento 1.1.1 - Software inSSIDer

Considerações: Há outro SSID operando no mesmo canal 1 do SSID TCC. Verifica-se instabilidade do sinal deste outro SSID, chegando a valores aproximados de -93 dBm. Já o sinal do SSID TCC mantém-se estável e constante na ordem de -50 dBm. Trataremos este ensaio com IRF desprezível.

Redes 2.4GHz | **SSID TCC**

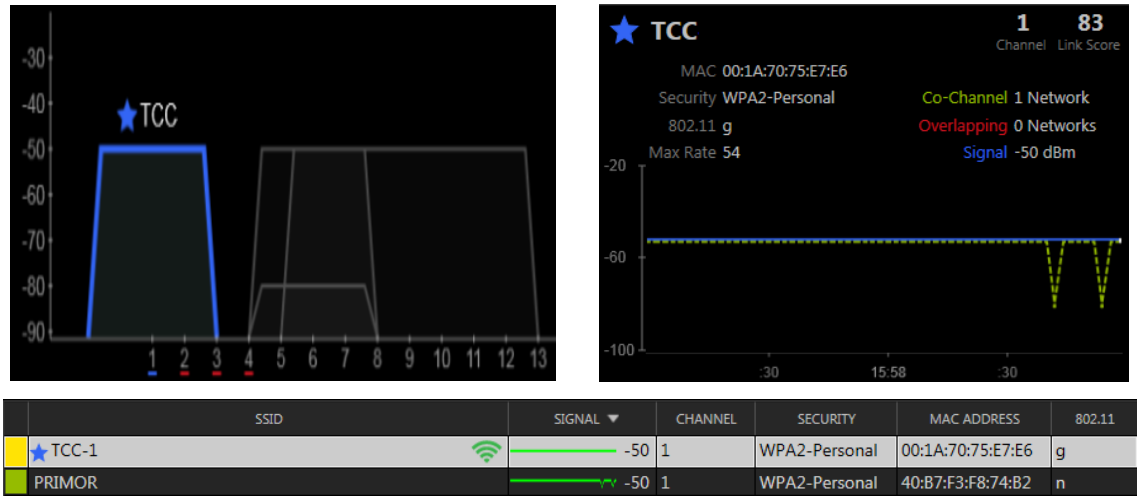
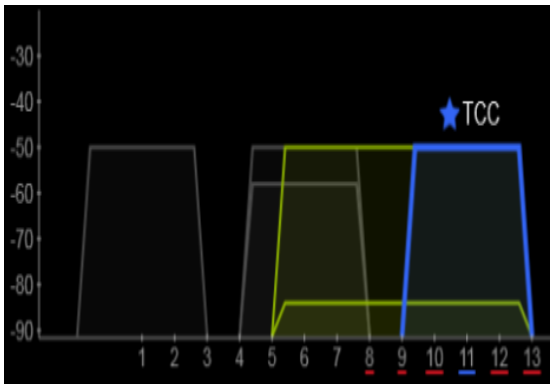


Tabela A.2 – Ambiente do experimento 1.1.2 - Software inSSIDer

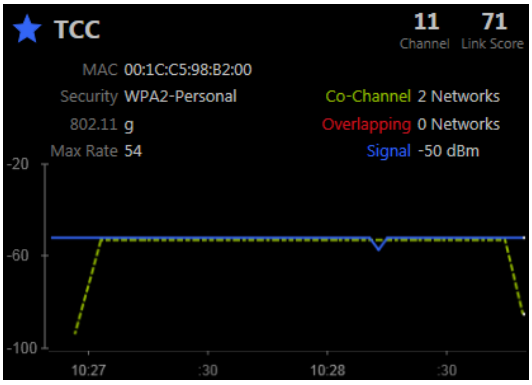
Considerações: Há outro SSID operando no mesmo canal 1 do SSID TCC. Verifica-se instabilidade do sinal deste outro SSID, chegando a valores aproximados de -85 dBm. Já o sinal do SSID TCC mantém-se estável e constante na ordem de -50 dBm. Trataremos este ensaio com IRF desprezível.

A.2 – Inspeção do ambiente: Wi-Fi 2.4GHz – AP Controller - SEM IRF Direta

Redes 2.4GHz



SSID TCC

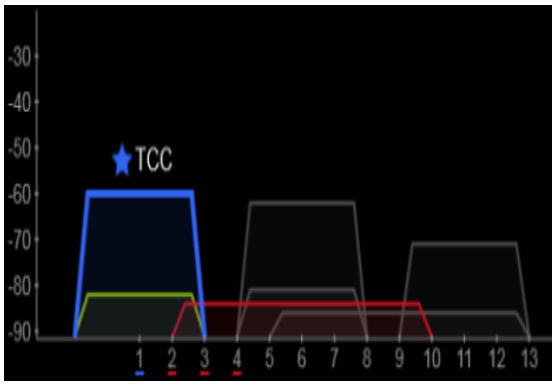


SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS	802.11
PRIMOR-TERREO	-50	11+7	WPA2-Personal	00:1A:3F:F3:01:74	n
TCC	-50	11	WPA2-Personal	00:1C:C5:98:B2:00	g

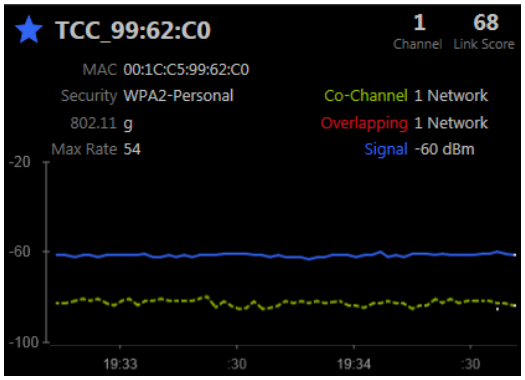
Tabela A.3 – Ambiente do experimento 1.2.1 - Software inSSIDer

Considerações: Há outro SSID operando no mesmo canal 11 do SSID TCC. Verifica-se instabilidade do sinal deste outro SSID, chegando a valores aproximados de -95 dBm. Já o sinal do SSID TCC mantém-se estável e constante na ordem de -50 dBm. Trataremos este ensaio com IRF desprezível.

Redes 2.4GHz



SSID TCC



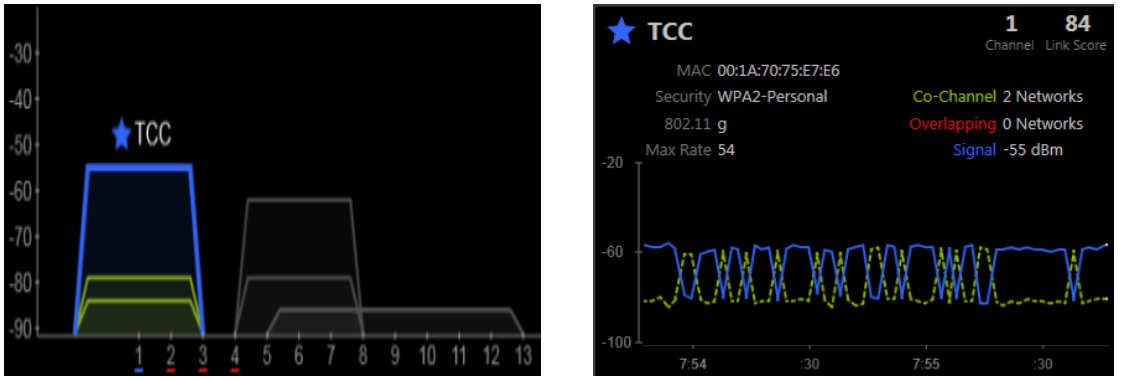
SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS	802.11
TCC	-60	1	WPA2-Personal	00:1C:C5:99:62:C0	g
TCC	-69	11	WPA2-Personal	00:1C:C5:98:B2:00	g
PRIMOR	-82	1	WPA2-Personal	40:B7:F3:F8:74:B2	n
PRIMOR-TERREO	-86	11+7	WPA2-Personal	00:1A:3F:F3:01:74	n

Tabela A.4 – Ambiente do experimento 1.2.2 - Software inSSIDer

Considerações: Há outro SSID operando no mesmo canal 1 e 11 do SSID TCC. Verifica-se estabilidade do sinal destes outros SSID, mantendo-se- entre -82 dBm e -86 dBm. Já o sinal do SSID TCC mantém-se estável e constante na ordem de –60 dBm a -69 dBm. Trataremos este ensaio com IRF desprezível.

A.3 – Inspeção do ambiente: Wi-Fi 2.4GHz – AP - COM IRF Direta

Redes 2.4GHz | **SSID TCC**

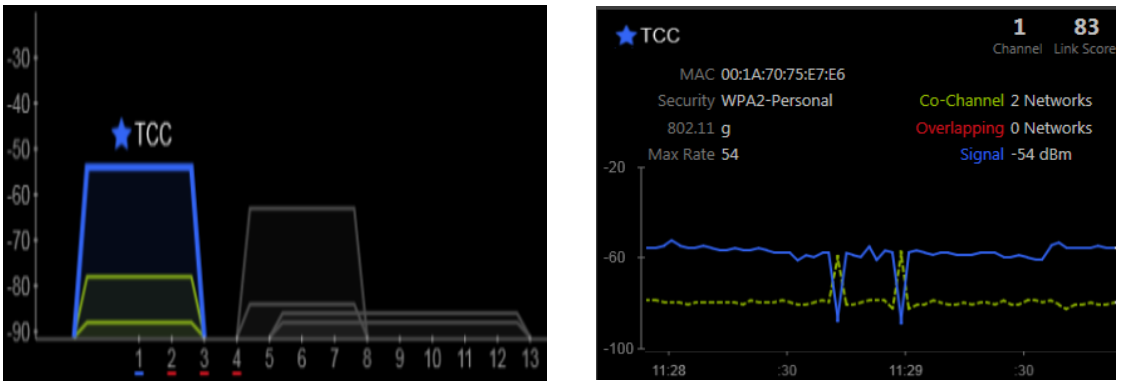


SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS	802.11
TCC	-55	1	WPA2-Personal	00:1A:70:75:E7:E6	g
IRF	-79	1	WPA2-Personal	00:1A:70:75:E7:E6	g

Tabela A.5 – Ambiente do experimento 2.1.1 - Software inSSIDer

Considerações: Instalado outro AP com SSID IRF com 0 dBm e 1mW, no mesmo canal 1 do SSID TCC. Verifica-se instabilidade dos sinais de TCC e IRF, oscilando fortemente entre -55 dBm a -79 dBm. Trataremos este ensaio com a presença de IRF.

Redes 2.4GHz | **SSID TCC**



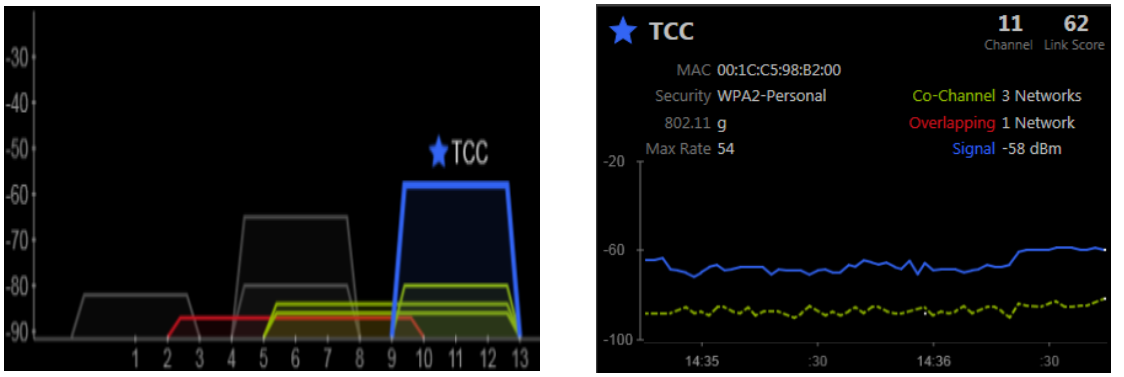
SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS	802.11
TCC	-54	1	WPA2-Personal	00:1A:70:75:E7:E6	g
IRF	-88	1	WPA2-Personal	00:1A:70:75:E7:E6	g

Tabela A.6 – Ambiente do experimento 2.1.2 - Software inSSIDer

Considerações: Instalado outro AP com SSID IRF com 0 dBm e 1mW, no mesmo canal 1 do SSID TCC. Verifica-se instabilidade dos sinais de TCC e IRF, oscilando fortemente entre -54 dBm a -88 dBm. Trataremos este ensaio com a presença de IRF.

A.4 – Inspeção do ambiente: Wi-Fi 2.4GHz – AP Controller - COM IRF Direta

Redes 2.4GHz | **SSID TCC**

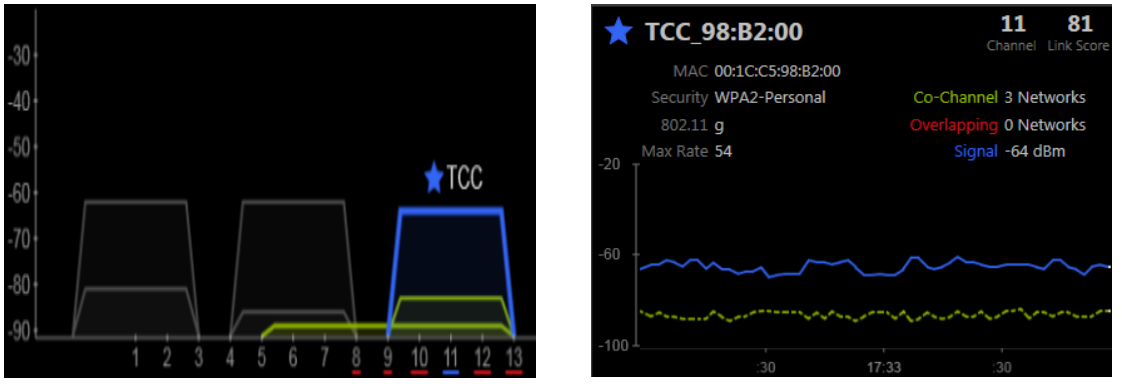


SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS	802.11
TCC	-58	11	WPA2-Personal	00:1C:C5:98:B2:00	g
IRF	-80	11	WPA2-Personal	00:1A:70:75:E7:E6	g

Tabela A.7 – Ambiente do experimento 2.2.1 - Software inSSIDer

Considerações: Instalado outro AP SSID IRF com 0 dBm e 1mW no mesmo canal 1 do SSID TCC. Automaticamente o canal SSID TCC foi alterado para o canal 11. Mudou-se o SSID IRF para o canal 11. Vê-se instabilidade dos sinais de TCC e IRF, com centro de -58 dBm e -80 dBm respectivamente. Trataremos este ensaio com a presença de IRF.

Redes 2.4GHz | **SSID TCC**



SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS	802.11
IRF	-83	11	WPA2-Personal	00:1A:70:75:E7:E6	g
TCC	-62	1	WPA2-Personal	00:1C:C5:99:62:C0	g
TCC	-64	11	WPA2-Personal	00:1C:C5:98:B2:00	g

Tabela A.8 – Ambiente do experimento 2.2.2 - Software inSSIDer

Considerações: Instalado outro AP com SSID IRF com 0 dBm e 1mW, em um dos canais do SSID TCC: canal 1. Automaticamente o host cliente é associado ao SSID TCC do canal 11. Vê-se instabilidade dos sinais de TCC e IRF, com centro de -62 dBm e -83 dBm respectivamente. Trataremos este ensaio com a presença de IRF.

APENDICE B

Estatística dos Experimentos

B.1 – Estatística dos Experimentos SEM Interferência Direta – 1MB

111	112	121	122	
33.930	35.054	47.564	43.103	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
34.086	34.210	33.899	52.276	
33.992	33.774	33.680	48.937	
34.008	33.931	33.852	36.863	
33.930	33.742	34.430	43.275	
34.039	33.868	33.930	40.528	
33.977	33.899	37.112	37.378	
33.993	33.868	36.769	33.743	
33.867	33.805	39.952	39.562	
33.821	37.207	37.144	42.837	
33.852	34.039	34.023	48.423	
33.977	33.992	33.759	39.593	
34.289	33.696	33.977	42.338	
33.618	33.930	33.821	51.449	
33.868	33.821	36.925	45.474	
33.758	33.774	39.842	36.551	
33.993	33.884	33.946	39.296	
33.821	34.195	34.039	42.323	
33.789	37.034	33.946	39.437	
33.883	34.008	36.910	49.015	
37.550	34.180	33.836	45.599	
33.618	34.133	37.144	45.537	
34.133	33.774	37.268	42.510	
33.711	33.790	40.279	42.198	
33.759	33.665	71.402	48.641	
33.821	33.836	110.588	36.348	
36.769	33.899	101.525	39.593	
33.790	33.946	83.351	39.530	
34.210	36.676	112.586	51.277	
34.024	34.320	65.052	45.506	
34.024	34.039	83.928	49.218	
34.126	**34.258**	**48.274**	**43.173**	**Média**
831	**941**	**24.451**	**4.981**	**D Padrão**

Tabela B.1.1 – 31 amostras capturadas de 1MB com Média e Desvio Padrão – Software MS-Excel

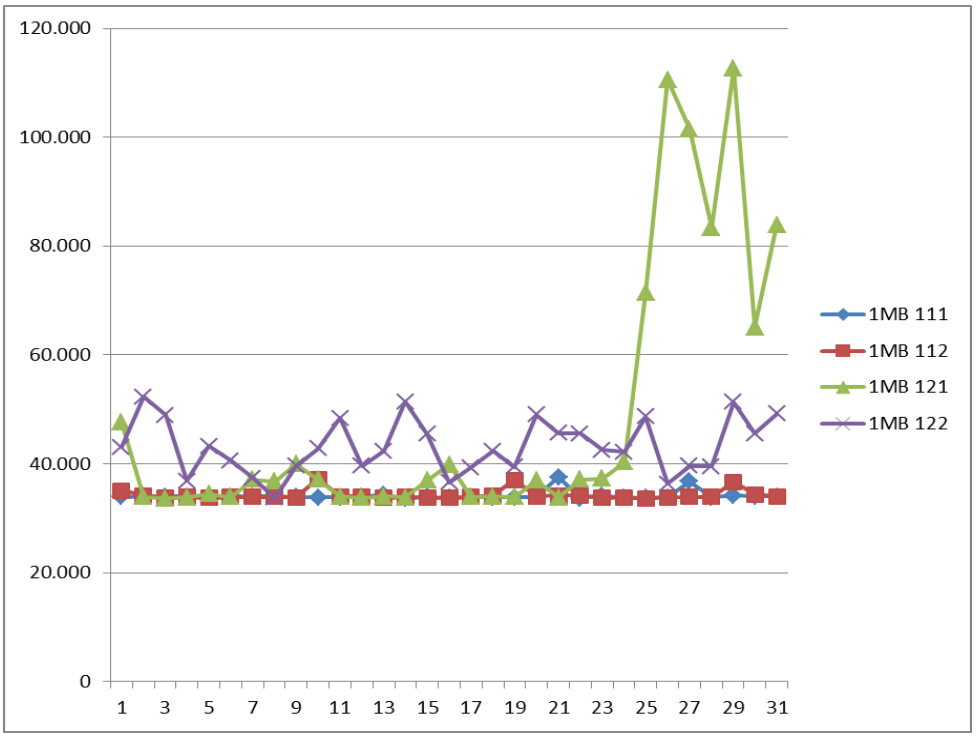


Gráfico B.1.1 – Comparativo entre amostras de 1MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
111	31	1057900	34125,80645	689771,428		
112	31	1061989	34257,70968	886253,8796		
121	31	1496479	48273,51613	597835872,7		
122	31	1338358	43172,83871	24807344,81		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	4525714386	3	1508571462	9,666933401	9,2E-06	2,68017
Dentro dos grupos	18726577283	120	156054810,7			
Total	23252291669	123				

Tabela B.1.2 – Teste ANOVA para 1MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	4525.71439	1508.57146	9.6669 **
Resíduo	120	18726.57728	156.05481	
Total	123	23252.29167		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	9.6669	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	34.12580 b	111
2	34.25771 b	112
3	48.27352 a	121
4	43.17284 a	122
dms =	8.27912	

MG = 39.95747 CV% = 31.26

Ponto médio = 73.10200

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES

FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.1.3 – Teste de Tukey para 1MB – Software ASSISTAT

B.1.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.1.1, os experimentos 111 e 112, mantêm as médias e desvio padrão aproximados entre si. O mesmo acontece com os experimentos 121 e 122, exceto pelo desvio padrão acentuado em 121. O gráfico B.1.1 retrata estas aproximações relatadas entre os experimentos. Com estas observações, verifica-se um possível melhor desempenho das transmissões ocorridas utilizando-se apenas AP individuais em contrapartida ao uso de controlador, em um ambiente com desprezível IRF.

B.1.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.1.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.1.3, foi encontrada igualdade estatística para as amostras 111 e 112, bem como as amostras 121 e 122.

Conclusão: Logo podemos afirmar que as transmissões de 1MB sem IRF (desprezível), com a utilização de AP individuais, obtiveram melhor desempenho em relação ao uso de controladores.

B.2 - Estatística dos Experimentos SEM Interferência Direta – 3MB

111	112	121	122	
104.707	101.182	163.192	128.123	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
102.071	101.665	105.487	113.443	
105.098	102.352	109.216	122.445	
103.630	104.879	106.938	136.469	
106.299	102.055	157.857	122.741	
105.675	101.354	172.848	136.235	
102.617	101.743	131.805	137.514	
102.492	104.754	103.943	128.544	
102.586	105.644	104.567	125.300	
102.975	102.273	101.166	124.909	
103.163	105.191	101.150	134.629	
106.517	105.987	101.151	131.118	
102.040	102.165	101.743	131.446	
102.633	102.351	104.583	131.133	
102.960	101.931	110.401	145.814	
103.225	101.728	102.056	140.665	
102.820	105.035	101.774	144.051	
105.799	102.539	102.586	123.178	
102.290	105.503	151.305	116.360	
102.648	102.960	104.754	125.565	
106.392	102.367	105.472	126.157	
105.909	105.784	108.482	140.791	
102.991	102.352	102.196	122.694	
102.664	103.163	102.211	128.622	
102.242	105.581	102.352	135.611	
102.368	102.351	107.999	137.686	
103.381	102.539	102.929	127.811	
103.366	105.815	107.859	132.397	
103.132	102.664	105.503	122.710	
103.412	105.566	105.675	113.553	
103.522	102.585	114.987	128.840	
103.601	**103.357**	**113.038**	**129.566**	**Média**
1.412	**1.631**	**19.944**	**8.208**	**D Padrão**

Tabela B.2.1 – 31 amostras capturadas de 3MB com Média e Desvio Padrão – Software MS-Excel

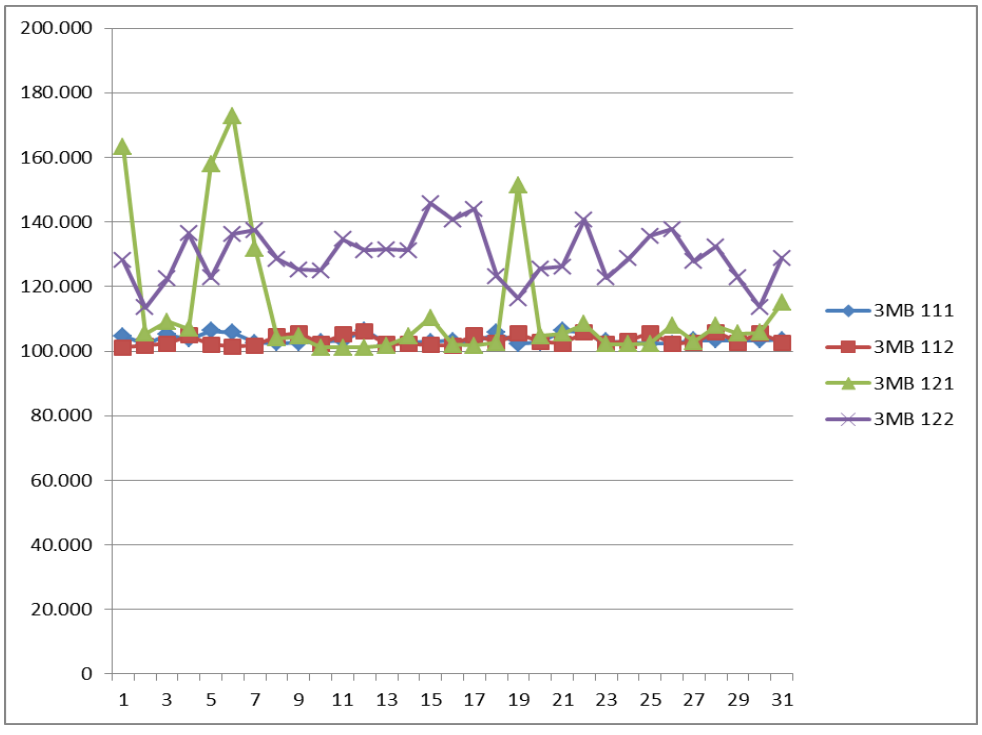


Gráfico B.2.1 – Comparativo entre amostras de 3MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
111	31	3211624	103600,7742	1993174,381		
112	31	3204058	103356,7097	2659351,613		
121	31	3504187	113038,2903	397745860,6		
122	31	4016554	129566,2581	67368456,53		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	14083143287	3	4694381096	39,97200879	5,6E-18	2,68017
Dentro dos grupos	14093005294	120	117441710,8			
Total	28176148581	123				

Tabela B.2.2 – Teste ANOVA para 3MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	14083.14329	4694.38110	39.9720 **
Resíduo	120	14093.00529	117.44171	
Total	123	28176.14858		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	39.972	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	103.60080 c	111
2	103.35670 c	112
3	113.03830 b	121
4	129.56630 a	122
dms =	7.18219	

MG = 112.39051 CV% = 9.64

Ponto médio = 136.99900

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.2.3 – Teste de Tukey para 3MB – Software ASSISTAT

B.2.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.2.1, os experimentos 111 e 112, mantêm as médias e desvio padrão aproximados entre si. O mesmo acontece com os experimentos 121 e 122, incluindo um desvio padrão mais acentuado. O gráfico B.2.1 retrata estas aproximações relatadas entre os experimentos. Com estas observações, verifica-se um possível melhor desempenho das transmissões ocorridas utilizando-se apenas AP individuais em contrapartida ao uso de controlador, em um ambiente com desprezível IRF.

B.2.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.2.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.2.3, foi encontrada igualdade estatística para as amostras 111 e 112. As demais amostras não ensejaram igualdades estatísticas entre si.

Conclusão: Logo podemos afirmar que as transmissões de 3MB sem IRF (desprezível), com a utilização de AP individuais, obtiveram melhor desempenho em relação ao uso de controladores.

B.3 – Estatística dos Experimentos SEM Interferência Direta – 5MB

111	112	121	122	
175.641	171.975	179.697	209.509	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
172.817	178.838	179.385	206.029	
178.745	175.891	174.393	172.412	
172.786	175.594	175.079	172.770	
179.837	172.988	178.823	172.521	
172.490	172.022	174.658	172.583	
172.957	174.236	180.414	183.098	
172.505	175.516	185.063	190.507	
176.296	177.045	181.631	221.676	
176.780	172.568	198.495	216.654	
173.191	173.394	187.154	230.396	
173.535	173.004	197.153	201.210	
176.109	176.343	185.734	219.461	
179.915	176.733	177.434	194.314	
174.439	173.129	185.500	212.612	
174.892	176.717	196.997	211.334	
177.809	174.798	173.410	209.711	
180.617	174.518	179.572	195.344	
178.870	174.938	182.817	223.579	
178.137	177.887	194.797	222.363	
175.001	174.830	183.129	189.447	
184.268	180.445	181.319	216.435	
178.745	177.529	183.644	212.566	
181.256	177.606	189.353	204.422	
177.185	181.350	187.231	211.771	
184.221	181.460	184.720	215.498	
183.191	180.071	193.441	193.300	
181.335	182.489	185.921	232.628	
181.257	182.068	185.110	206.716	
189.758	179.042	182.364	202.910	
185.079	180.804	195.515	209.835	
178.054	**176.640**	**184.515**	**204.310**	**Média**
4.359	**3.155**	**7.010**	**17.007**	**D Padrão**

Tabela B.3.1 – 31 amostras capturadas de 5MB com Média e Desvio Padrão – Software MS-Excel

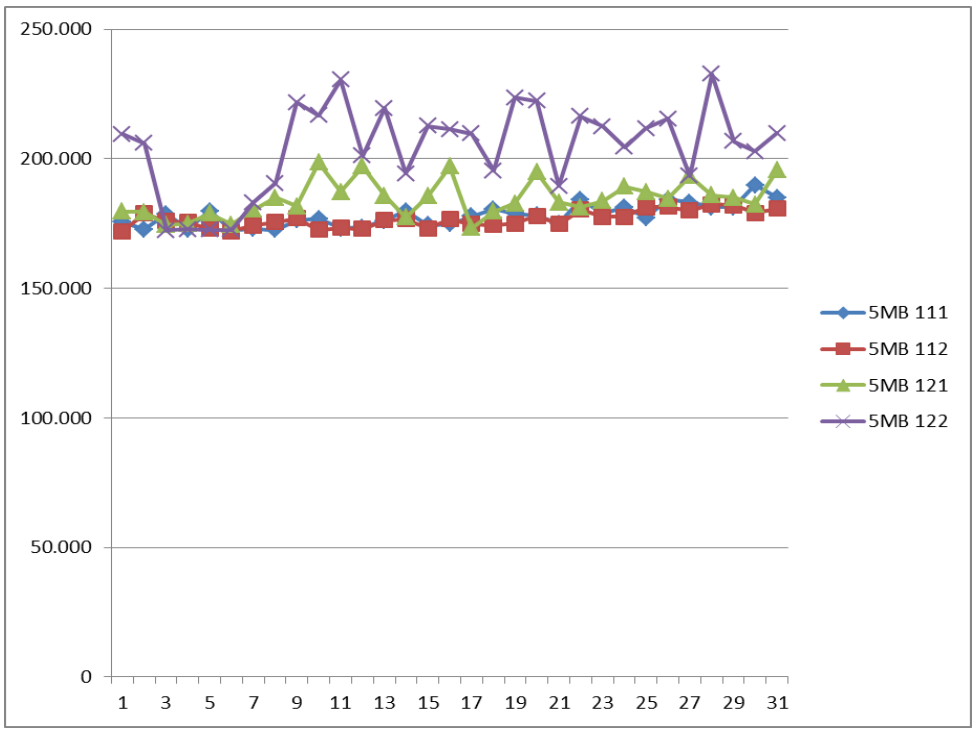


Gráfico B.3.1 – Comparativo entre amostras de 5MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
111	31	5519664	178053,6774	18999449,29		
112	31	5475828	176639,6129	9956076,645		
121	31	5719953	184514,6129	49141540,91		
122	31	6333611	204310,0323	289224426,8		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	15133159911	3	5044386637	54,93157056	2E-22	2,68017
Dentro dos grupos	11019644810	120	91830373,42			
Total	26152804721	123				

Tabela B.3.2 – Teste ANOVA para 5MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	15133.15991	5044.38664	54.9316 **
Resíduo	120	11019.64481	91.83037	
Total	123	26152.80472		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	54.9316	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	178.05370 c	111
2	176.63960 c	112
3	184.51460 b	121
4	204.31000 a	122
dms =	6.35095	

MG = 185.87948 CV% = 5.16

Ponto médio = 202.30150

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.3.3 – Teste de Tukey para 5MB – Software ASSISTAT

B.3.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.3.1, os experimentos 111 e 112, mantêm as médias e desvio padrão aproximados entre si. O mesmo acontece com os experimentos 121 e 122, incluindo um desvio padrão mais acentuado. O gráfico B.3.1 retrata estas aproximações relatadas entre os experimentos. Com estas observações, verifica-se um possível melhor desempenho das transmissões ocorridas utilizando-se apenas AP individuais em contrapartida ao uso de controlador, em um ambiente com desprezível IRF.

B.3.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.3.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.3.3, foi encontrada igualdade estatística para as amostras 111 e 112. As demais amostras não ensejaram igualdades estatísticas entre si.

Conclusão: Logo podemos afirmar que as transmissões de 5MB sem IRF (desprezível), com a utilização de AP individuais, obtiveram melhor desempenho em relação ao uso de controladores.

B.4 – Estatística dos Experimentos COM Interferência Direta – 1MB

211	212	221	222	
46.301	33.104	52.572	38.610	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
68.484	32.947	36.473	33.743	
35.490	32.526	33.977	33.524	
35.646	32.635	39.562	36.520	
42.089	32.417	34.086	36.847	
45.443	32.526	36.613	36.613	
50.638	32.464	33.681	33.696	
45.536	35.412	40.497	39.437	
47.658	32.354	36.551	33.665	
85.644	32.401	33.431	33.462	
94.568	32.402	36.332	39.687	
84.396	35.490	39.250	36.706	
50.825	32.276	33.540	33.556	
45.411	35.412	36.535	33.571	
41.684	32.355	39.515	39.702	
39.171	33.041	36.770	39.547	
42.589	32.432	33.337	36.660	
41.542	32.401	33.571	33.508	
38.205	32.480	36.769	45.553	
39.094	32.479	36.676	33.243	
42.229	32.386	36.800	36.130	
49.998	35.474	36.770	36.395	
35.521	38.454	42.603	33.306	
40.155	32.511	39.609	36.395	
58.001	32.744	36.598	33.321	
72.898	32.386	33.618	36.738	
42.121	32.401	36.597	33.946	
49.311	32.479	33.088	33.820	
62.447	33.587	41.137	33.634	
68.640	34.164	33.025	36.645	
42.900	33.587	33.650	36.738	
51.117	**33.217**	**36.878**	**35.965**	**Média**
15.754	**1.420**	**3.923**	**2.811**	**D Padrão**

Tabela B.4.1 – 31 amostras capturadas de1MB com Média e Desvio Padrão – Software MS-Excel

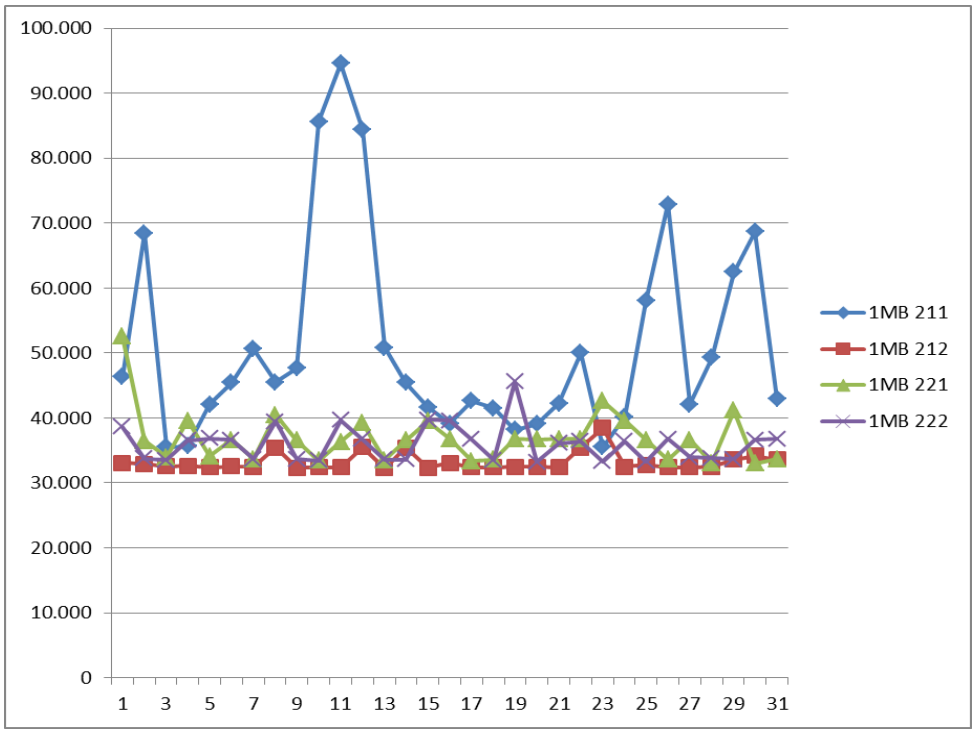


Gráfico B.4.1 – Comparativo entre amostras de 1MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*
211	31	1584635	51117,25806	2,5E+08
212	31	1029727	33217	2016593
221	31	1143233	36878,48387	1,5E+07
222	31	1114918	35965,09677	7899131

ANOVA

Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	6002705902	3	2000901967	29,2643	2,85068E-14	2,68017
Dentro dos grupos	8204805106	120	68373375,89			
Total	14207511009	123				

Tabela B.4.2 – Teste ANOVA para 1MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F	
Tratamentos	3	6002.70590	2000.90197	29.2643	**
Resíduo	120	8204.80511	68.37338		
Total	123	14207.51101			

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	29.2643	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	51.11726 a	211
2	33.21700 b	212
3	36.87848 b	221
4	35.96510 b	222
dms =	5.48011	

MG = 39.29446 CV% = 21.04
Ponto médio = 63.42200

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.4.3 – Teste de Tukey para 1MB – Software ASSISTAT

B.4.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.4.1, os experimentos 212, 221 e 222 mantiveram as médias e desvio padrão aproximados entre si. Apenas o experimento 211 não acompanhou a média dos demais. O gráfico B.4.1 retrata estas aproximações relatadas entre os experimentos. Com estas observações, verifica-se que com o advento da IRF, o controlador obtém uma boa regularidade das transmissões. O AP individualizado só se igualou estatisticamente ao controlador com a elevação da potencia de transmissão peculiar do experimento 212. Destaque-se que este último experimento, obteve a menor média e menor desvio padrão.

B.4.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.4.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.4.3, foi encontrada igualdade estatística para as amostras 212, 221 e 222.

Conclusão: Logo podemos afirmar que as transmissões de 1MB em ambiente com IRF utilizando controlador, obtiveram melhor desempenho em relação ao uso de AP individual. Para igualar-se ao controlador, o AP individual teve sua potencia de transmissão elevada no experimento 212.

7.5 – Estatística dos Experimentos COM Interferência Direta – 3MB

211	212	221	222	
167.919	112.726	108.248	101.493	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
305.573	107.468	103.865	103.538	
197.169	109.122	116.782	109.886	
205.468	110.074	138.575	109.513	
156.219	100.340	119.434	106.875	
123.989	97.515	154.237	101.666	
167.029	97.485	146.032	107.578	
157.638	97.313	100.636	104.130	
160.275	97.329	112.929	103.600	
201.147	100.417	104.395	101.603	
178.636	97.687	110.604	104.333	
172.302	100.308	108.030	113.443	
118.763	110.371	107.672	104.084	
147.998	100.963	104.270	104.114	
134.160	100.807	110.261	113.397	
128.279	103.772	105.129	107.203	
123.115	100.792	101.728	107.422	
114.130	101.634	101.416	110.636	
140.541	97.594	104.707	106.704	
120.276	97.593	107.047	116.485	
128.029	98.421	103.849	101.432	
110.838	101.431	104.380	104.910	
113.880	97.750	104.598	123.224	
107.313	101.104	107.266	116.845	
116.127	98.108	105.144	116.438	
110.838	97.719	110.261	108.358	
122.678	97.937	107.359	104.941	
109.591	100.698	101.478	104.723	
106.626	104.629	107.453	116.345	
107.531	97.282	104.770	101.775	
109.824	97.875	107.484	111.212	
143.997	**101.105**	**110.646**	**107.997**	**Média**
42.426	**4.446**	**12.724**	**5.603**	**D Padrão**

Tabela B.5.1 – 31 amostras capturadas de 3MB com Média e Desvio Padrão – Software MS-Excel

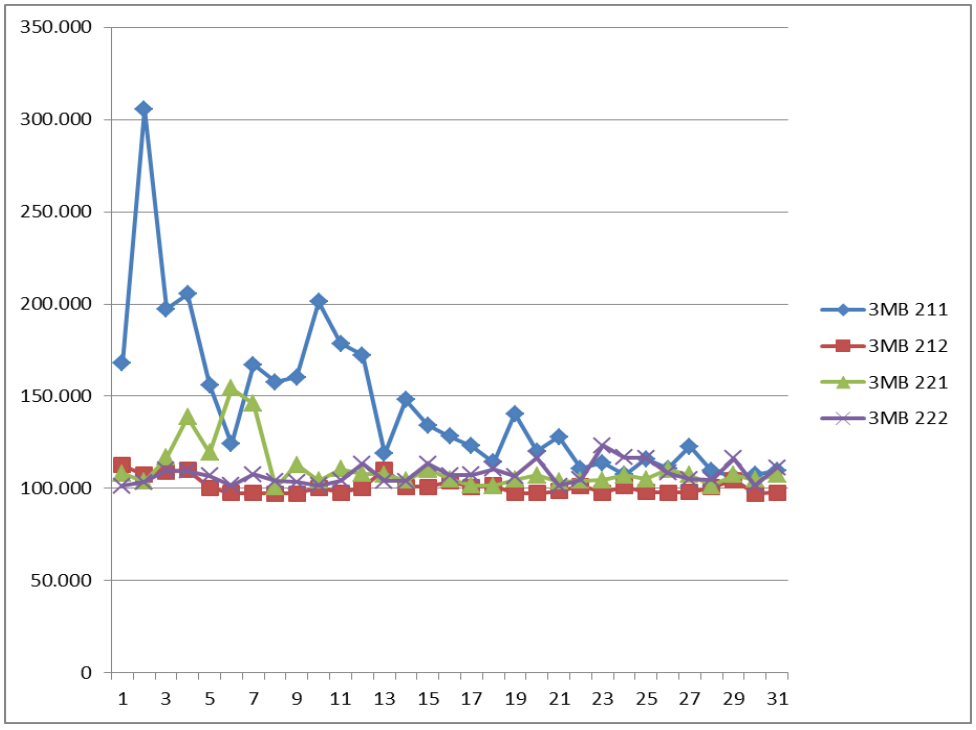


Gráfico B.5.1 – Comparativo entre amostras de 3MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
211	31	4463901	143996,8065	1799954242		
212	31	3134264	101105,2903	19768620,95		
221	31	3430039	110646,4194	161893720		
222	31	3347906	107996,9677	31398977,1		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	34049367833	3	11349789278	22,55280983	1,2E-11	2,68017
Dentro dos grupos	60390466802	120	503253890			
Total	94439834635	123				

Tabela B.5.2 – Teste ANOVA para 3MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	34049.36783	11349.78928	22.5528 **
Resíduo	120	60390.46680	503.25389	
Total	123	94439.83463		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	22.5528	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	143.99680 a	211
2	101.10530 b	212
3	110.64640 b	221
4	107.99700 b	222
dms =	14.86754	

MG = 115.93637 CV% = 19.35
Ponto médio = 201.42750

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.5.3 – Teste de Tukey para 3MB – Software ASSISTAT

B.5.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.5.1, os experimentos 212, 221 e 222 mantiveram as médias e desvio padrão aproximados entre si. Apenas o experimento 2.1.1 não acompanhou a média dos demais. O gráfico B.5.1 retrata estas aproximações relatadas entre os experimentos. Com estas observações, verifica-se que com o advento da IRF, o controlador obtém um boa regularidade das transmissões. O AP individualizado só se igualou ao controlador com a elevação da potencia de transmissão peculiar do experimento 212. Destaque-se que este último experimento, obteve a menor média e menor desvio padrão.

B.5.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.5.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.5.3, foi encontrada igualdade estatística para as amostras 212, 221 e 222.

Conclusão: Logo podemos afirmar que as transmissões de 3MB em ambiente com IRF utilizando controlador, obtiveram melhor desempenho em relação ao uso de AP individual. Para igualar-se ao controlador, o AP individual teve sua potência de transmissão elevada no experimento 212.

B.6 – Estatística dos Experimentos COM Interferência Direta – 5MB

211	212	221	222	
190.710	175.718	175.454	181.850	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
179.229	166.921	181.225	182.770	
187.543	164.518	178.511	170.929	
175.251	170.492	175.282	170.337	
200.117	164.144	178.574	184.985	
209.493	169.994	181.725	194.423	
193.971	164.970	203.065	176.639	
178.807	167.701	175.968	172.911	
188.027	168.527	185.142	178.777	
197.637	173.222	191.365	182.036	
194.174	170.524	182.988	189.213	
195.359	167.311	188.106	181.834	
177.981	168.074	194.641	173.457	
175.937	169.370	169.588	170.165	
168.152	165.750	173.238	173.098	
172.973	166.811	179.822	173.659	
165.985	167.232	182.676	176.109	
169.104	166.796	179.884	174.517	
168.387	167.185	192.130	180.820	
168.808	167.170	180.055	178.901	
169.978	170.415	180.212	183.691	
193.206	168.059	177.575	176.436	
194.688	171.647	180.477	188.386	
195.281	171.632	192.941	174.533	
198.355	171.444	184.330	185.515	
203.627	173.098	185.687	186.702	
222.051	174.096	183.254	183.269	
193.752	175.766	181.646	200.398	
190.664	176.732	186.343	253.375	
184.845	181.943	190.335	196.670	
207.153	180.087	202.302	218.962	
187.460	**170.237**	**183.695**	**184.367**	**Média**
14.130	**4.407**	**7.772**	**16.398**	**D Padrão**

Tabela B.6.1 – 31 amostras capturadas de 5MB com Média e Desvio Padrão – Software MS-Excel

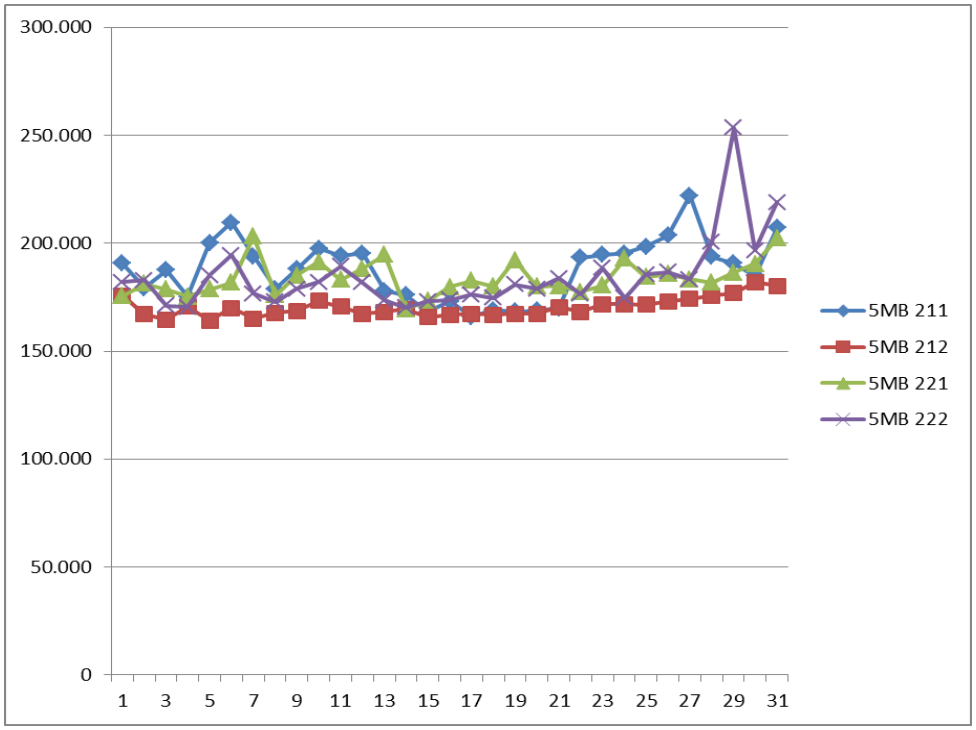


Gráfico B.6.1 – Comparativo entre amostras de 5MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
211	31	5811245	187459,5161	199664257,3		
212	31	5277349	170237,0645	19419586,8		
221	31	5694541	183694,871	60406798,58		
222	31	5715367	184366,6774	268895836,3		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	5437096845	3	1812365615	13,21962291	1,6E-07	2,68017
Dentro dos grupos	16451594368	120	137096619,7			
Total	21888691213	123				

Tabela B.6.2 – Teste ANOVA para 5MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	5437.09685	1812.36562	13.2196 **
Resíduo	120	16451.59437	137.09662	
Total	123	21888.69121		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	13.2196	<0.001

MÉDIAS E MEDIDAS

	Médias de tratamento	Experimento
1	187.45950 a	211
2	170.23710 b	212
3	183.69490 a	221
4	184.36670 a	222
dms =	7.75995	

MG = 181.43953 CV% = 6.45
Ponto médio = 208.75950

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.6.3 – Teste de Tukey para 5MB – Software ASSISTAT

B.6.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.6.1, os experimentos 211, 221 e 222 mantiveram as médias e desvio padrão aproximados entre si. Apenas o experimento 212 destacou-se perante as demais médias. O gráfico B.6.1 retrata estas aproximações relatadas entre os experimentos. Com estas observações, verifica-se que com o advento da IRF, o controlador obtém uma boa regularidade das transmissões. Entretanto, o AP individualizado, com a elevação da potencia de transmissão peculiar do experimento 212, apresentou as melhores média e desvio padrão.

B.6.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.6.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.6.3, foi encontrada igualdade estatística para as amostras 211, 221 e 222.

Conclusão: Logo podemos afirmar que as transmissões de 5MB em ambiente com IRF utilizando AP individualizado com potência elevada acima dos demais equipamentos, obtive melhor desempenho em relação ao uso dos demais dispositivos.

B.7 – Estatística dos Experimentos AP (sem e com IRF) – 1MB

111	112	211	212	
33.930	35.054	46.301	33.104	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
34.086	34.210	68.484	32.947	
33.992	33.774	35.490	32.526	
34.008	33.931	35.646	32.635	
33.930	33.742	42.089	32.417	
34.039	33.868	45.443	32.526	
33.977	33.899	50.638	32.464	
33.993	33.868	45.536	35.412	
33.867	33.805	47.658	32.354	
33.821	37.207	85.644	32.401	
33.852	34.039	94.568	32.402	
33.977	33.992	84.396	35.490	
34.289	33.696	50.825	32.276	
33.618	33.930	45.411	35.412	
33.868	33.821	41.684	32.355	
33.758	33.774	39.171	33.041	
33.993	33.884	42.589	32.432	
33.821	34.195	41.542	32.401	
33.789	37.034	38.205	32.480	
33.883	34.008	39.094	32.479	
37.550	34.180	42.229	32.386	
33.618	34.133	49.998	35.474	
34.133	33.774	35.521	38.454	
33.711	33.790	40.155	32.511	
33.759	33.665	58.001	32.744	
33.821	33.836	72.898	32.386	
36.769	33.899	42.121	32.401	
33.790	33.946	49.311	32.479	
34.210	36.676	62.447	33.587	
34.024	34.320	68.640	34.164	
34.024	34.039	42.900	33.587	
34.126	**34.258**	**51.117**	**33.217**	**Média**
831	**941**	**15.754**	**1.420**	**D Padrão**

Tabela B.7.1 – 31 amostras capturadas de 1MB com Média e Desvio Padrão – Software MS-Excel

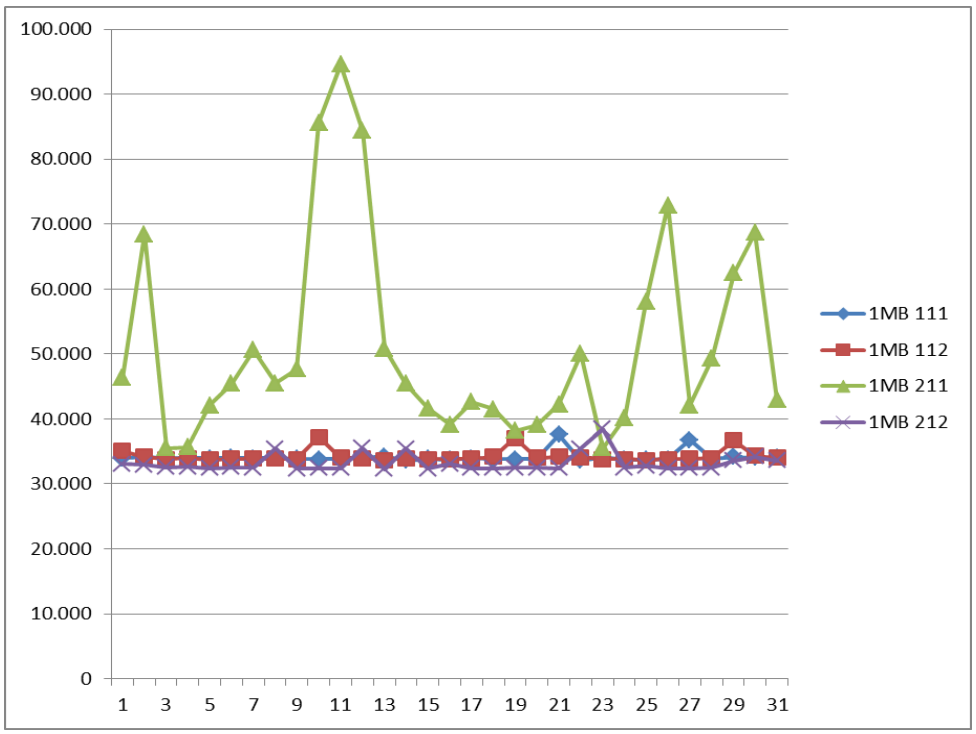


Gráfico B.7.1 – Comparativo entre amostras de 1MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
111	31	1057900	34125,80645	689771,428		
112	31	1061989	34257,70968	886253,8796		
211	31	1584635	51117,25806	248186432,2		
212	31	1029727	33217	2016593,4		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	6938570681	3	2312856894	36,74423087	6,44143E-17	2,68016757
Dentro dos grupos	7553371527	120	62944762,73			
Total	14491942209	123				

Tabela B.7.2 – Teste ANOVA para 1MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	6938.57068	2312.85689	36.7442 **
Resíduo	120	7553.37153	62.94476	
Total	123	14491.94221		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	36.7442	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	34.12580 b	111
2	34.25771 b	112
3	51.11726 a	211
4	33.21700 b	212
dms =	5.25806	

MG = 38.17944 CV% = 20.78

Ponto médio = 63.42200

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES

FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.7.3 – Teste de Tukey para 1MB – Software ASSISTAT

B.7.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.7.1, os experimentos 111, 112 e 212 mantiveram as médias e desvio padrão aproximados entre si, sendo que o experimento 212, obteve a melhor média. Apenas o experimento 211 não acompanhou a média e desvio padrão dos demais. O gráfico B.7.1 retrata estas aproximações relatadas entre os experimentos. Desta forma, as médias e desvio padrão dos AP individuais no ambiente sem IRF, foi similar as médias e desvio padrão do AP individual no ambiente com IRF, quando este AP teve sua potência elevada.

B.7.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.7.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.7.3, foi encontrada igualdade estatística para as amostras 111, 112 e 212.

Conclusão: Logo podemos afirmar que as transmissões de 1MB em ambientes sem IRF, não sofrem alterações nos tempos de transmissão quando se eleva a potencia do rádio: experimento 111 e 112. Mas quando estas transmissões ocorrem em ambientes com IRF, a elevação da potência é fundamental para a manutenção da média e desvio padrão conseguido no ambiente sem IRF: experimento 212.

B.8 – Estatística dos Experimentos AP (sem e com IRF) – 3MB

111	112	211	212	
104.707	101.182	167.919	112.726	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
102.071	101.665	305.573	107.468	
105.098	102.352	197.169	109.122	
103.630	104.879	205.468	110.074	
106.299	102.055	156.219	100.340	
105.675	101.354	123.989	97.515	
102.617	101.743	167.029	97.485	
102.492	104.754	157.638	97.313	
102.586	105.644	160.275	97.329	
102.975	102.273	201.147	100.417	
103.163	105.191	178.636	97.687	
106.517	105.987	172.302	100.308	
102.040	102.165	118.763	110.371	
102.633	102.351	147.998	100.963	
102.960	101.931	134.160	100.807	
103.225	101.728	128.279	103.772	
102.820	105.035	123.115	100.792	
105.799	102.539	114.130	101.634	
102.290	105.503	140.541	97.594	
102.648	102.960	120.276	97.593	
106.392	102.367	128.029	98.421	
105.909	105.784	110.838	101.431	
102.991	102.352	113.880	97.750	
102.664	103.163	107.313	101.104	
102.242	105.581	116.127	98.108	
102.368	102.351	110.838	97.719	
103.381	102.539	122.678	97.937	
103.366	105.815	109.591	100.698	
103.132	102.664	106.626	104.629	
103.412	105.566	107.531	97.282	
103.522	102.585	109.824	97.875	
103.601	**103.357**	**143.997**	**101.105**	**Média**
1.412	**1.631**	**42.426**	**4.446**	**D Padrão**

Tabela B.8.1 – 31 amostras capturadas de 3MB com Média e Desvio Padrão – Software MS-Excel

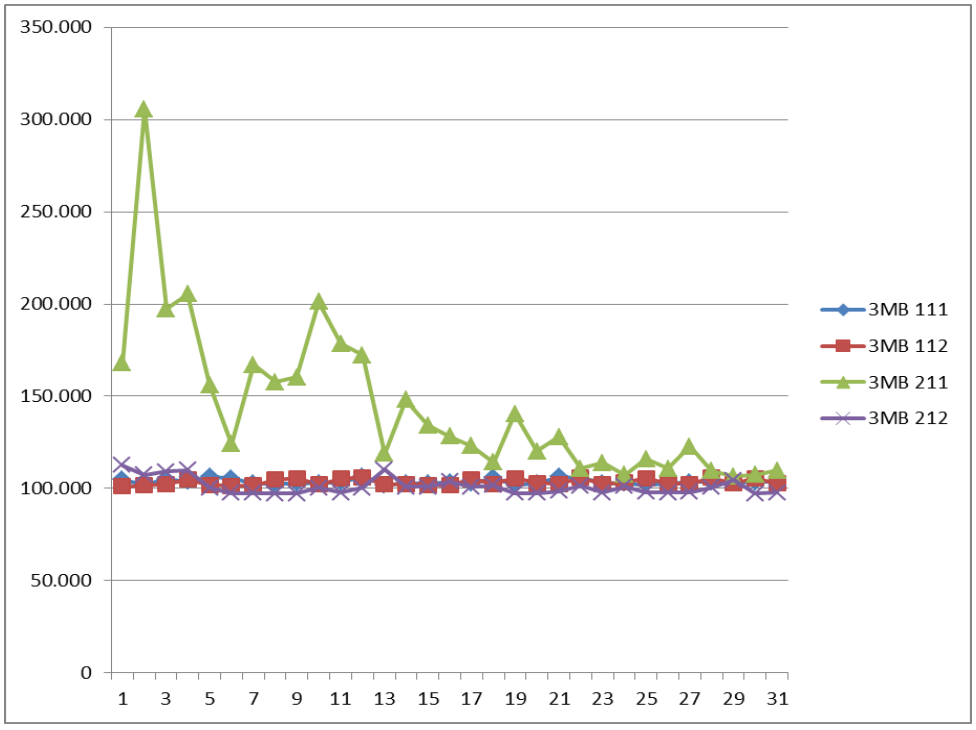


Gráfico B.8.1 – Comparativo entre amostras de 3MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
111	31	3211624	103600,7742	1993174,381		
112	31	3204058	103356,7097	2659351,613		
211	31	4463901	143996,8065	1799954242		
212	31	3134264	101105,2903	19768620,95		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	39792335787	3	13264111929	29,08197953	3,33342E-14	2,68016757
Dentro dos grupos	54731261669	120	456093847,2			
Total	94523597456	123				

Tabela B.8.2 – Teste ANOVA para 3MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	39792.33579	13264.11193	29.0820 **
Resíduo	120	54731.26167	456.09385	
Total	123	94523.59746		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	29.082	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	103.60080 b	111
2	103.35670 b	112
3	143.99680 a	211
4	101.10530 b	212
dms =	14.15379	

MG = 113.01490 CV% = 18.90

Ponto médio = 201.42750

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.8.3 – Teste de Tukey para 3MB – Software ASSISTAT

B.8.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.8.1, os experimentos 111, 112 e 212 mantiveram as médias e desvio padrão aproximados entre si, sendo que o experimento 212, obteve a melhor média. Apenas o experimento 211 não acompanhou a média e desvio padrão dos demais. O gráfico B.8.1 retrata estas aproximações relatadas entre os experimentos. Desta forma, as médias e desvio padrão dos AP individuais no ambiente sem IRF, foi similar as médias e desvio padrão do AP individual no ambiente com IRF, quando este AP teve sua potência elevada.

B.8.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.8.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.8.3, foi encontrada igualdade estatística para as amostras 111, 112 e 212.

Conclusão: Logo podemos afirmar que as transmissões de 3MB em ambientes sem IRF, não sofrem alterações nos tempos de transmissão quando se eleva a potencia do rádio: experimento 111 e 112. Mas quando estas transmissões ocorrem em ambientes com IRF, a elevação da potência é fundamental para a manutenção da média e desvio padrão conseguido no ambiente sem IRF: experimento 212.

B.9 – Estatística dos Experimentos AP (sem e com IRF) – 5MB

111	112	211	212	
175.641	171.975	190.710	175.718	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
172.817	178.838	179.229	166.921	
178.745	175.891	187.543	164.518	
172.786	175.594	175.251	170.492	
179.837	172.988	200.117	164.144	
172.490	172.022	209.493	169.994	
172.957	174.236	193.971	164.970	
172.505	175.516	178.807	167.701	
176.296	177.045	188.027	168.527	
176.780	172.568	197.637	173.222	
173.191	173.394	194.174	170.524	
173.535	173.004	195.359	167.311	
176.109	176.343	177.981	168.074	
179.915	176.733	175.937	169.370	
174.439	173.129	168.152	165.750	
174.892	176.717	172.973	166.811	
177.809	174.798	165.985	167.232	
180.617	174.518	169.104	166.796	
178.870	174.938	168.387	167.185	
178.137	177.887	168.808	167.170	
175.001	174.830	169.978	170.415	
184.268	180.445	193.206	168.059	
178.745	177.529	194.688	171.647	
181.256	177.606	195.281	171.632	
177.185	181.350	198.355	171.444	
184.221	181.460	203.627	173.098	
183.191	180.071	222.051	174.096	
181.335	182.489	193.752	175.766	
181.257	182.068	190.664	176.732	
189.758	179.042	184.845	181.943	
185.079	180.804	207.153	180.087	
178.054	**176.640**	**187.460**	**170.237**	**Média**
4.359	**3.155**	**14.130**	**4.407**	**D Padrão**

Tabela B.9.1 – 31 amostras capturadas de 5MB com Média e Desvio Padrão – Software MS-Excel

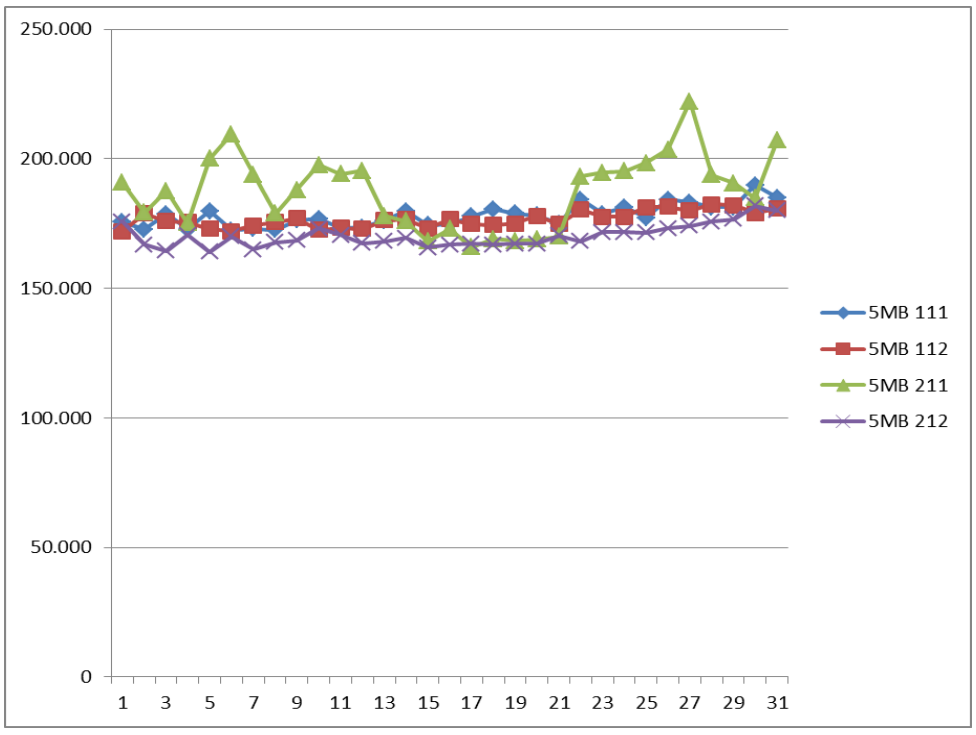


Gráfico B.9.1 – Comparativo entre amostras de 5MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
111	31	5519664	178053,6774	18999449,29		
112	31	5475828	176639,6129	9956076,645		
211	31	5811245	187459,5161	199664257,3		
212	31	5277349	170237,0645	19419586,8		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	4698395563	3	1566131854	25,25618178	9,77093E-13	2,68016757
Dentro dos grupos	7441181100	120	62009842,5			
Total	12139576663	123				

Tabela B.9.2 – Teste ANOVA para 5MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	4698.39556	1566.13185	25.2562 **
Resíduo	120	7441.18110	62.00984	
Total	123	12139.57666		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	25.2562	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	178.05370 b	111
2	176.63960 b	112
3	187.45950 a	211
4	170.23710 c	212
dms =	5.21886	

MG = 178.09747 CV% = 4.42

Ponto médio = 193.09750

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.9.3 – Teste de Tukey para 5MB – Software ASSISTAT

B.9.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.9.1, os experimentos 111 e 112 mantiveram as médias e desvio padrão aproximados entre si. Os experimentos 211 e 212, não guardaram médias e desvio padrão aproximados. O gráfico B.9.1 retrata estas aproximações relatadas entre os experimentos. Desta forma, as médias e desvio padrão dos AP individuais no ambiente sem IRF, foi similar as médias e desvio padrão do AP individual no ambiente com IRF, quando este AP teve sua potência elevada.

B.9.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.9.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.9.3, foi encontrada igualdade estatística para as amostras 111 e 112. Entretanto o melhor caso foi o experimento 212, que se destacou estatisticamente dentre os demais.

Conclusão: Logo podemos afirmar que as transmissões de 5MB em ambientes sem IRF, não sofrem alterações nos tempos de transmissão quando se eleva a potencia do rádio: experimento 111 e 112. Mas quando estas transmissões ocorrem em ambientes com IRF, a elevação da potência é fundamental para a manutenção da média e desvio padrão conseguido no ambiente sem IRF: experimento 212. Neste caso com 5MB, o uso de AP individualizado com sua potencia tendo sido elevada, foi o melhor tempo obtido estatisticamente.

B.10 – Estatística dos Experimentos Controlador (sem e com IRF) – 1MB

121	122	221	222	
47.564	43.103	52.572	38.610	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
33.899	52.276	36.473	33.743	
33.680	48.937	33.977	33.524	
33.852	36.863	39.562	36.520	
34.430	43.275	34.086	36.847	
33.930	40.528	36.613	36.613	
37.112	37.378	33.681	33.696	
36.769	33.743	40.497	39.437	
39.952	39.562	36.551	33.665	
37.144	42.837	33.431	33.462	
34.023	48.423	36.332	39.687	
33.759	39.593	39.250	36.706	
33.977	42.338	33.540	33.556	
33.821	51.449	36.535	33.571	
36.925	45.474	39.515	39.702	
39.842	36.551	36.770	39.547	
33.946	39.296	33.337	36.660	
34.039	42.323	33.571	33.508	
33.946	39.437	36.769	45.553	
36.910	49.015	36.676	33.243	
33.836	45.599	36.800	36.130	
37.144	45.537	36.770	36.395	
37.268	42.510	42.603	33.306	
40.279	42.198	39.609	36.395	
71.402	48.641	36.598	33.321	
110.588	36.348	33.618	36.738	
101.525	39.593	36.597	33.946	
83.351	39.530	33.088	33.820	
112.586	51.277	41.137	33.634	
65.052	45.506	33.025	36.645	
83.928	49.218	33.650	36.738	
48.274	**43.173**	**36.878**	**35.965**	**Média**
24.451	**4.981**	**3.923**	**2.811**	**D Padrão**

Tabela B.10.1 – 31 amostras capturadas de 1MB com Média e Desvio Padrão – Software MS-Excel

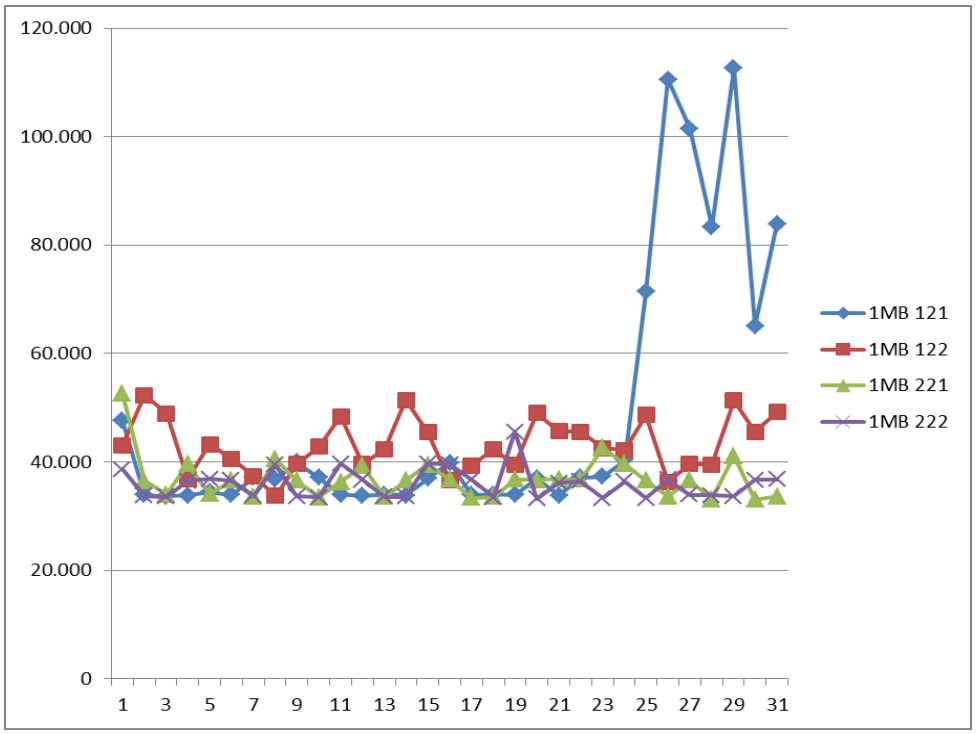


Gráfico B.10.1 – Comparativo entre amostras de 1MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
121	31	1496479	48273,51613	597835872,7		
122	31	1338358	43172,83871	24807344,81		
221	31	1143233	36878,48387	15391347,26		
222	31	1114918	35965,09677	7899130,69		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	3098183245	3	1032727748	6,395255461	0,000467607	2,68016757
Dentro dos grupos	19378010862	120	161483423,9			
Total	22476194107	123				

Tabela B.10.2 – Teste ANOVA para 1MB – Software MS-Excel

```
                EXPERIMENTO INTEIRAMENTE CASUALIZADO

                          QUADRO DE ANÁLISE
------------------------------------------------------------------------
FV              GL           SQ              QM              F
------------------------------------------------------------------------
Tratamentos      3       3098.18324      1032.72775       6.3953 **
Resíduo        120      19378.01086       161.48342
------------------------------------------------------------------------
Total          123      22476.19411
------------------------------------------------------------------------
** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

    GL           GLR          F-crit       F            p
    3            120          3.9491       6.3953       <0.001

                          MÉDIAS E MEDIDAS

                  Médias de tratamento  Experimento
                 ---------------------- -----------
                    1      48.27352 a       121
                    2      43.17284 ab      122
                    3      36.87848  b      221
                    4      35.96510  b      222
                 ---------------------- -----------
                   dms =       8.42189

   MG = 41.07248                                    CV% = 30.94
                        Ponto médio = 72.80550

          As médias seguidas pela mesma letra não diferem
          estatisticamente entre si. Foi aplicado o Teste
             de Tukey ao nível de 5% de probabilidade

 SIGLAS E ABREVIAÇÕES
          FV = Fonte de variação  GL = Graus de liberdade
          SQ = Soma de quadrado       QM = Quadrado médio
          F = Estatística do teste F     MG = Média geral
          CV% = Coeficiente de variação em %
          dms = Diferença mínima significativa
```

Tabela B.10.3 – Teste de Tukey para 1MB – Software ASSISTAT

B.10.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.10.1, os experimentos 121 e 122 mantiveram as médias e desvio padrão aproximados entre si, bem como os experimentos 221 e 222. O gráfico B.10.1 retrata estas aproximações relatadas entre os experimentos. Foi possível observar que as médias dos tempos obtidos com o controlador em ambiente com IRF, foram melhores que os tempos obtidos no mesmo controlador, quando na ausência de IRF.

B.10.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.10.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.10.3, foi encontrada igualdade estatística para as amostras 121 e 122. Foi encontrada também, igualdade estatística para as amostras 221 e 222. O experimento 122 está no limiar, onde se iguala estatisticamente com o experimento 121 e com os experimentos 221 e 222.

Conclusão: Logo podemos afirmar que as transmissões de 1MB em ambientes fazendo uso de controlador, obtêm melhores tempos de transmissão, quando sujeitos a IRF.

B.11 – Estatística dos Experimentos Controlador (sem e com IRF) – 3MB

121	122	221	222	
163.192	128.123	108.248	101.493	LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)
105.487	113.443	103.865	103.538	
109.216	122.445	116.782	109.886	
106.938	136.469	138.575	109.513	
157.857	122.741	119.434	106.875	
172.848	136.235	154.237	101.666	
131.805	137.514	146.032	107.578	
103.943	128.544	100.636	104.130	
104.567	125.300	112.929	103.600	
101.166	124.909	104.395	101.603	
101.150	134.629	110.604	104.333	
101.151	131.118	108.030	113.443	
101.743	131.446	107.672	104.084	
104.583	131.133	104.270	104.114	
110.401	145.814	110.261	113.397	
102.056	140.665	105.129	107.203	
101.774	144.051	101.728	107.422	
102.586	123.178	101.416	110.636	
151.305	116.360	104.707	106.704	
104.754	125.565	107.047	116.485	
105.472	126.157	103.849	101.432	
108.482	140.791	104.380	104.910	
102.196	122.694	104.598	123.224	
102.211	128.622	107.266	116.845	
102.352	135.611	105.144	116.438	
107.999	137.686	110.261	108.358	
102.929	127.811	107.359	104.941	
107.859	132.397	101.478	104.723	
105.503	122.710	107.453	116.345	
105.675	113.553	104.770	101.775	
114.987	128.840	107.484	111.212	
113.038	**129.566**	**110.646**	**107.997**	**Média**
19.944	**8.208**	**12.724**	**5.603**	**D Padrão**

Tabela B.11.1 – 31 amostras capturadas de 3MB com Média e Desvio Padrão – Software MS-Excel

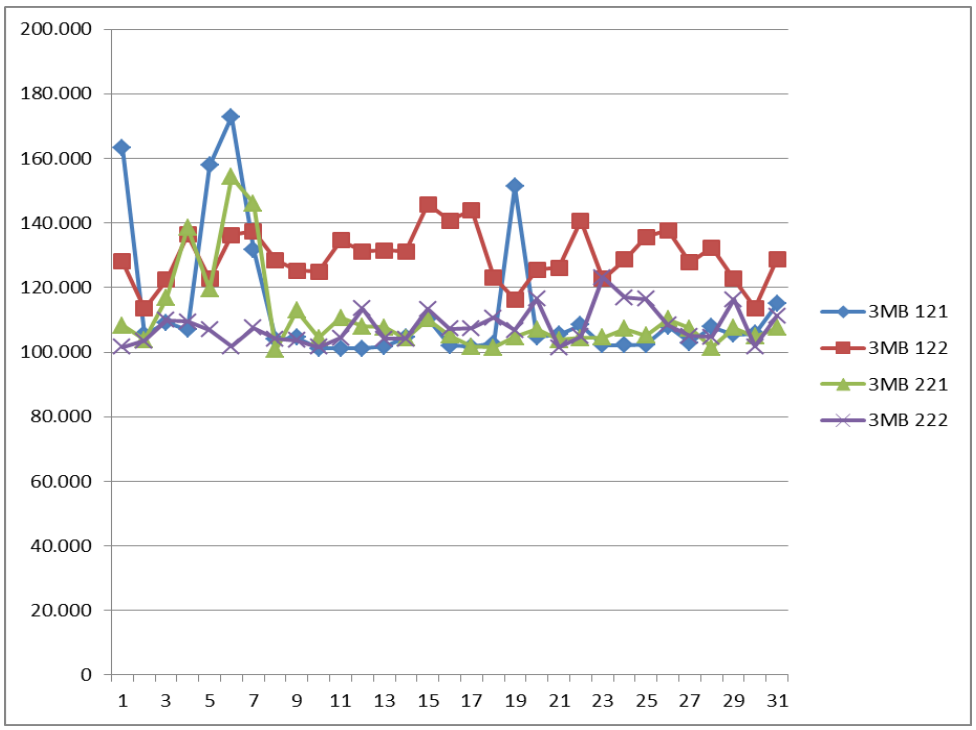


Gráfico B.11.1 – Comparativo entre amostras de 3MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*		
121	31	3504187	113038,2903	397745860,6		
122	31	4016554	129566,2581	67368456,53		
221	31	3430039	110646,4194	161893720		
222	31	3347906	107996,9677	31398977,1		
ANOVA						
Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	8792560019	3	2930853340	17,80572367	1,26375E-09	2,68016757
Dentro dos grupos	19752210427	120	164601753,6			
Total	28544770446	123				

Tabela B.11.2 – Teste ANOVA para 3MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	8792.56002	2930.85334	17.8057 **
Resíduo	120	19752.21043	164.60175	
Total	123	28544.77045		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	17.8057	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	113.03830 b	121
2	129.56630 a	122
3	110.64640 b	221
4	107.99700 b	222
dms =	8.50282	

MG = 115.31198 CV% = 11.13

Ponto médio = 136.74200

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES

FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.11.3 – Teste de Tukey para 3MB – Software ASSISTAT

B.11.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.11.1, os experimentos 121, 221 e 222 mantiveram as médias e desvio padrão aproximados entre si. O gráfico B.11.1 retrata estas aproximações relatadas entre os experimentos. Foi possível observar que as médias dos tempos obtidos com o controlador em ambiente com IRF, foram melhores que os tempos obtidos no mesmo controlador, quando na ausência de IRF.

B.11.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.11.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.11.3, foi encontrada igualdade estatística para as amostras 121, 221 e 222.

Conclusão: Logo podemos afirmar que as transmissões de 3MB em ambientes fazendo uso de controlador, mesmo obtendo médias e desvio padrão melhores no ambiente com IRF, não diferem muito entre si, já que o experimento 121 manteve-se em igualdade de condições estatísticas com o os experimentos em ambiente de IRF.

B.12 – Estatística dos Experimentos Controlador (sem e com IRF) – 5MB

121	122	221	222	
179.697	209.509	175.454	181.850	
179.385	206.029	181.225	182.770	
174.393	172.412	178.511	170.929	
175.079	172.770	175.282	170.337	
178.823	172.521	178.574	184.985	
174.658	172.583	181.725	194.423	
180.414	183.098	203.065	176.639	
185.063	190.507	175.968	172.911	
181.631	221.676	185.142	178.777	
198.495	216.654	191.365	182.036	
187.154	230.396	182.988	189.213	
197.153	201.210	188.106	181.834	
185.734	219.461	194.641	173.457	
177.434	194.314	169.588	170.165	
185.500	212.612	173.238	173.098	
196.997	211.334	179.822	173.659	
173.410	209.711	182.676	176.109	
179.572	195.344	179.884	174.517	
182.817	223.579	192.130	180.820	
194.797	222.363	180.055	178.901	
183.129	189.447	180.212	183.691	
181.319	216.435	177.575	176.436	
183.644	212.566	180.477	188.386	
189.353	204.422	192.941	174.533	
187.231	211.771	184.330	185.515	
184.720	215.498	185.687	186.702	
193.441	193.300	183.254	183.269	
185.921	232.628	181.646	200.398	
185.110	206.716	186.343	253.375	
182.364	202.910	190.335	196.670	
195.515	209.835	202.302	218.962	
184.515	**204.310**	**183.695**	**184.367**	**Média**
7.010	**17.007**	**7.772**	**16.398**	**D Padrão**

LABORATÓRIO - SEQUÊNCIA DAS AMOSTRAS EM (ms)

Tabela B.12.1 – 31 amostras capturadas de 5MB com Média e Desvio Padrão – Software MS-Excel

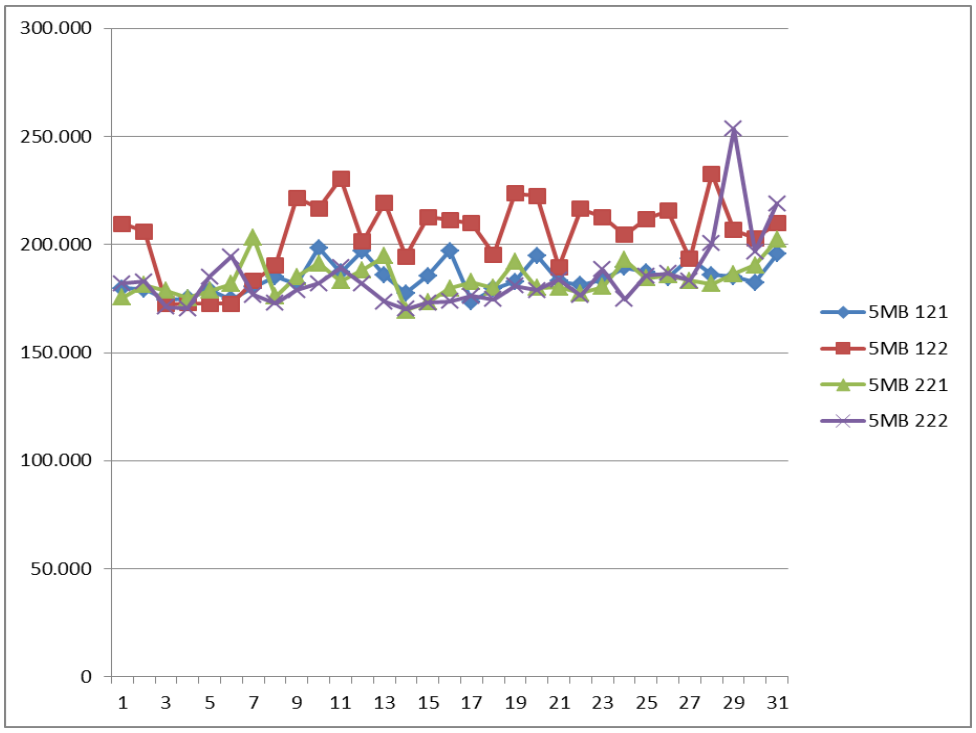


Gráfico B.12.1 – Comparativo entre amostras de 5MB – Software MS-Excel

Grupo	*Contagem*	*Soma*	*Média*	*Variância*
121	31	5719953	184514,6129	49141540,91
122	31	6333611	204310,0323	289224426,8
221	31	5694541	183694,871	60406798,58
222	31	5715367	184366,6774	268895836,3

ANOVA

Fonte da variação	*SQ*	*gl*	*MQ*	*F*	*valor-P*	*F crítico*
Entre grupos	9421877200	3	3140625733	18,81547655	4,55224E-10	2,68016757
Dentro dos grupos	20030058079	120	166917150,7			
Total	29451935279	123				

Tabela B.12.2 – Teste ANOVA para 5MB – Software MS-Excel

EXPERIMENTO INTEIRAMENTE CASUALIZADO

QUADRO DE ANÁLISE

FV	GL	SQ	QM	F
Tratamentos	3	9421.87720	3140.62573	18.8155 **
Resíduo	120	20030.05808	166.91715	
Total	123	29451.93528		

** significativo ao nível de 1% de probabilidade (p < .01)
 * significativo ao nível de 5% de probabilidade (.01 =< p < .05)
ns não significativo (p >= .05)

GL	GLR	F-crit	F	p
3	120	3.9491	18.8155	<0.001

MÉDIAS E MEDIDAS

Médias de tratamento		Experimento
1	184.51460 b	121
2	204.31000 a	122
3	183.69490 b	221
4	184.36670 b	222
dms =	8.56241	

MG = 189.22155 CV% = 6.83
Ponto médio = 211.48150

As médias seguidas pela mesma letra não diferem estatisticamente entre si. Foi aplicado o Teste de Tukey ao nível de 5% de probabilidade

SIGLAS E ABREVIAÇÕES
FV = Fonte de variação GL = Graus de liberdade
SQ = Soma de quadrado QM = Quadrado médio
F = Estatística do teste F MG = Média geral
CV% = Coeficiente de variação em %
dms = Diferença mínima significativa

Tabela B.12.3 – Teste de Tukey para 5MB – Software ASSISTAT

B.12.1 - Comentários: Tabela e Gráfico

De acordo com a tabela B.12.1, os experimentos 121, 221 e 222 mantiveram as médias e desvio padrão aproximados entre si. O gráfico B.12.1 retrata estas aproximações relatadas entre os experimentos. Foi possível observar que as médias dos tempos obtidos com o controlador em ambiente com IRF, foram um pouco melhores que os tempos obtidos no mesmo controlador, quando na ausência de IRF.

B.12.2 – Comentários: ANOVA e Teste de Tukey

Tomou-se a hipótese nula (H0), de que os tempos de transmissão seriam iguais. Utilizou-se ANOVA fator único, para comprovação ou não desta afirmação:

H0: as médias dos tempos são estatisticamente iguais
H1: existe pelo menos uma média que não é estatisticamente igual

ANOVA: De acordo com tabela B.12.2, o teste ANOVA evidenciou que F é maior que F crítico, ensejando a rejeição a H0, logo as médias de transmissão não são iguais.

Teste de Tukey: Aplicado o teste de Tukey conforme tabela B.12.3, foi encontrada igualdade estatística para as amostras 121, 221 e 222.

Conclusão: Logo podemos afirmar que as transmissões de 5MB em ambientes fazendo uso de controlador, mesmo obtendo médias e desvio padrão melhores no ambiente com IRF, não diferem muito entre si, já que o experimento 121 manteve-se em igualdade de condições estatísticas com o os experimentos em ambiente de IRF.

APENDICE C

Listagem dos softwares Servidor e Cliente

C.1 – Listagem do Software Servidor

```
/*
IES                 UEMA - Universidade Estadual do Maranhão
Curso               Engenharia da Computação - Mestrado
Prof. Orientador    Henrique Mariano Costa do Amaral
Aluno               Raimundo de Carvalho Silva Neto
Tema Dissertação    Avaliação de desempenho do protocolo TCP em redes wireless
                    local: com e sem o uso de controladores
Software            Programa *** S E R V I D O R *** para geração de tráfego TCP
                    - Versão 1.0 - 19.09.2012
/*

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <winsock.h>

#define BACKLOG_MAX 5
#define BUFFER_SIZE 1.450
#define EXIT_CALL_STRING "#quit"

int local_socket = 0;
int remote_socket = 0;

int remote_length = 0;
int message_length = 0;

int qtd_recv = 0;

unsigned short local_port = 49151;
unsigned short remote_port = 0;

// Variaveis para identificar IP do Local
hostent* local_Host;
char* local_IP;

char message[BUFFER_SIZE];

struct sockaddr_in local_address;
struct sockaddr_in remote_address;

WSADATA wsa_data;

/* Exibe uma mensagem de erro e termina o programa */
void msg_err_exit(char *msg)
{
 fprintf(stderr, msg);
 system("PAUSE");
 exit(EXIT_FAILURE);
```

```
}

int main(int argc, char **argv)
{
   // inicia o Winsock 2.0 (DLL), Only for Windows
   if (WSAStartup(MAKEWORD(2, 0), &wsa_data) != 0)
     msg_err_exit("WSAStartup() failed\n");

   // criando o socket local para o servidor
   local_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
   if (local_socket == INVALID_SOCKET)
     {
        WSACleanup();
        msg_err_exit("socket() failed\n");
     }

   // Pegar IP do Servidor Local
   local_Host = gethostbyname("");
   local_IP   = inet_ntoa (*(struct in_addr *)*local_Host->h_addr_list);
   printf("Do SERVIDOR TCP/IP ..... = %s:%d\n",local_IP,local_port);

   // zera a estrutura local_address
   memset(&local_address, 0, sizeof(local_address));

   // internet address family
   local_address.sin_family = AF_INET;

   // porta local
   local_address.sin_port = htons(local_port);

   // endereco
   local_address.sin_addr.s_addr = htonl(INADDR_ANY);

   // interligando o socket com o endereço (local)
   if (bind(local_socket, (struct sockaddr *) &local_address, sizeof(local_address)) ==
SOCKET_ERROR)
     {
        WSACleanup();
        closesocket(local_socket);
        msg_err_exit("bind() failed\n");
     }

   // coloca o socket para escutar as conexoes
   if (listen(local_socket, BACKLOG_MAX) == SOCKET_ERROR)
     {
        WSACleanup();
        closesocket(local_socket);
        msg_err_exit("listen() failed\n");
     }
```

```
   // Loop infinito para receber solcitações
   do
   {
     // Get IP do host remoto
     struct sockaddr_in remote_address;

     remote_length = sizeof(remote_address);

     printf("Aguardando conexao.");
     remote_socket = accept(local_socket, (struct sockaddr *) &remote_address,
&remote_length);
     if(remote_socket == INVALID_SOCKET)
     {
        WSACleanup();
        closesocket(local_socket);
        msg_err_exit("accept() failed\n");
     }

     // limpa o buffer
     memset(&message, 0, BUFFER_SIZE);

     // recebe a mensagem do cliente
     message_length = recv(remote_socket, message, BUFFER_SIZE, 0);
     if(message_length == SOCKET_ERROR)
        msg_err_exit("recv() failed\n");

     qtd_recv = qtd_recv + 1;

     system("cls");
     printf("*** UEMA - Universidade Estadual do Maranhao\n");
     printf("*** Mestrado em Engenharia da Computação\n");
     printf("*** Professor: Henrique Mariano\n");
     printf("*** Aluno: Raimundo de Carvalho Silva Neto\n");
     printf("*** Programa SERVIDOR para geracao de trafefo TCP\n");
     printf("=============================================\n");
     printf("\n");

     // Pegar IP do Servidor Local
     local_Host = gethostbyname("");
     local_IP   = inet_ntoa (*(struct in_addr *)*local_Host->h_addr_list);

     printf("Do SERVIDOR TCP/IP ..... = %s:%d\n",local_IP,local_port);
     printf("Para o CLIENTE TCP/IP ..... =
%s:%d\n",inet_ntoa(remote_address.sin_addr),remote_address.sin_port);
     printf("Quantidade pacotes recebidos ... = %d\n",qtd_recv);
     printf("Caracteres recebidos ..... = \n%s",message);

     shutdown(remote_socket,SD_RECEIVE);
     closesocket(remote_socket);
   }
```

```
    // sai quando receber um "#quit" do cliente
    while(strcmp(message, EXIT_CALL_STRING));

    printf("encerrando\n");
    WSACleanup();
    closesocket(local_socket);
    closesocket(remote_socket);

    system("PAUSE");
    return 0;
}
```

C.2 – Listagem do Software Cliente

```
/*
IES                 UEMA - Universidade Estadual do Maranhão
Curso               Engenharia da Computação - Mestrado
Prof. Orientador    Henrique Mariano Costa do Amaral
Aluno               Raimundo de Carvalho Silva Neto
Tema Dissertação    Avaliação de desempenho do protocolo TCP em redes wireless
                    local: com e sem o uso de controladores
Software            Programa *** C L I E N T E *** para geração de tráfego TCP -
                    Versão 1.0 - 19.09.2012


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <time.h>
#include <winsock.h>
#define BUFFER_SIZE 1.450

int local_socket = 0;
int remote_socket = 0;

int message_length = 0;

int bytes_env = 0;
int contador = 0;
int qtd_send = 0;
int laco0, laco1, laco2 = 0;
int qtd_megas = 0;

clock_t t_ini, t_fim;
double  t_tot;

int num_experimento;
char experimento[11];

unsigned short local_port = 0;
unsigned short remote_port = 49151;

// Variaveis para identificar IP Local
hostent* local_Host;
char* local_IP;
//

char remote_ip[32];
char message[BUFFER_SIZE];
char dados[11] = "xxxxxxxxx1";
```

```
struct sockaddr_in local_address;
struct sockaddr_in remote_address;

WSADATA wsa_data;

FILE * pFile;

/* Exibe uma mensagem de erro e termina o programa */
void msg_err_exit(char *msg)
{
  fprintf(stderr, msg);
  system("PAUSE");
  exit(EXIT_FAILURE);
}

int main(int argc, char **argv)
{
   if (WSAStartup(MAKEWORD(2, 0), &wsa_data) != 0)
      msg_err_exit("WSAStartup() failed\n");

   system("cls");
   printf("*** UEMA - Universidade Estadual do Maranhao\n");
   printf("*** Mestrado em Engenharia da Computação\n");
   printf("*** Professor: Henrique Mariano\n");
   printf("*** Aluno: Raimundo de Carvalho Silva Neto\n");
   printf("*** Programa CLIENTE para geracao de trafefo TCP\n");
   printf("=======================================\n");
   printf("\n");

   printf("Entre com o numero do IP do servidor: ");
   scanf("%s", remote_ip);
   fflush(stdin);

   printf("Entre com o Numero do Experimento: ");
   scanf("%d", &num_experimento);

   // criando o socket local para o cliente
   do
   {
     printf("*** Criando socket local ...\n");
     local_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
   }
   while (local_socket == INVALID_SOCKET);

   // preenchendo o local_address (cliente)
   memset(&local_address, 0, sizeof(local_address));
   local_address.sin_family = AF_INET;
   local_address.sin_addr.s_addr = htonl(INADDR_ANY);;
   local_address.sin_port = htons(local_port);
```

```
// Pegar IP do Cliente Local
local_Host = gethostbyname("");
local_IP   = inet_ntoa (*(struct in_addr *)*local_Host->h_addr_list);

// Cria e abre arquivo de registros de transmissao
sprintf(experimento,"wifi%d.txt", num_experimento);

pFile = fopen (experimento,"w");

// laco0 -> Laço para: 1MB, 3MB, 5MB
// quantidade de Megabytes tranmitidos: 685=1MB, 2055=3MB, 3425=5MB.
//for(laco0=1; laco0<=3; laco0++)
   for(laco0=1; laco0<=3; laco0++)
{
  switch (laco0)
  {
    case 1:
      qtd_megas = 685;
      break;
    case 2:
      qtd_megas = 2055;
      break;
    case 3:
      qtd_megas = 3425;
      break;
  }

  // laco1 -> Laço Principal
  for(laco1=1; laco1<=31; laco1++)
  {

    //Inicio da contagem do tempo
    t_ini = clock();

    // laco2 -> Laço Secundario
    for(laco2=1; laco2<=qtd_megas; laco2++)
    {
      // limpa o buffer
      memset(&message, 0, BUFFER_SIZE);

      // Dá carga na menssage em 1.450 bytes
      for (contador = 1; contador <= 145; contador++)
      {
        strcat(message,dados);
      }

      message_length = strlen(message);
      // fim limpa o buffer
```

```
        do
        {
          printf("*** Abrindo socket no servidor ...\n");
          remote_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
        }
        while (remote_socket == INVALID_SOCKET);

        // preenchendo o remote_address (servidor)
        memset(&remote_address, 0, sizeof(remote_address));
        remote_address.sin_family = AF_INET;
        remote_address.sin_addr.s_addr = inet_addr(remote_ip);
        remote_address.sin_port = htons(remote_port);

        // conexao com o servidor
        do
        {
          printf("*** Tentanto conexao com servidor ...\n");
        }
        while (connect(remote_socket, (struct sockaddr *) &remote_address,
sizeof(remote_address)) == SOCKET_ERROR);

        // envia a mensagem para o servidor
        bytes_env = 0;
        do
        {
          printf("*** Enviando bytes ao servidor ...\n");
          bytes_env = send(remote_socket, message, message_length, 0);
        }
        while (bytes_env == 0 or (send(remote_socket, message, message_length, 0) ==
SOCKET_ERROR));

        // Encerrando conexão
        shutdown(remote_socket,SD_SEND);
        closesocket(remote_socket);

        qtd_send = qtd_send + 1;

        //*
        system("cls");
        printf("*** UEMA - Universidade Estadual do Maranhao\n");
        printf("*** Mestrado em Engenharia da Computação\n");
        printf("*** Professor: Henrique Mariano\n");
        printf("*** Aluno: Raimundo de Carvalho Silva Neto\n");
        printf("*** Programa CLIENTE para geracao de trafefo TCP\n");
        printf("=======================================\n");
        printf("\n");

        printf("Experimento ..... = %s\n", experimento);
        printf("Do CLIENTE TCP/IP ..... = %s:%d\n", local_IP,
local_address.sin_port);
```

```
        printf("Para o SERVIDOR TCP/IP ..... = %s:%d\n", remote_ip, remote_port);
        printf("Tamanho do segmento ..... = %d\n", bytes_env);
        printf("Quantidade de Bytes transmitidos = %d\n", bytes_env * qtd_send);
        printf("Pacotes enviados ..... = %d\n", qtd_send);
        printf("Repeticao ..... = %d\n", laco1);
        printf("Caracteres enviados ..... = %d\n", qtd_megas);
        //system("PAUSE");
        //*/
      }

      qtd_send = 0;

      //system("PAUSE");
      t_fim = clock();

      t_tot =(1000*(t_fim - t_ini) / CLOCKS_PER_SEC);
      printf("\n\nTempo de transmissao: %lf ms\n", t_tot);

      switch (laco0)
      {
        case 1:
          contador = 1;
          break;
        case 2:
          contador = 3;
          break;
        case 3:
          contador = 5;
          break;
      }

      // Gravar no arquivo de transmissoes
      fprintf(pFile, "%dMB, %d, %lf\n", contador, laco1, t_tot);
    }
}

// Fechar arquivo de transmissoes
fclose (pFile);

strcat(message,"#quit");
message_length = strlen(message);

remote_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

memset(&remote_address, 0, sizeof(remote_address));
remote_address.sin_family = AF_INET;
remote_address.sin_addr.s_addr = inet_addr(remote_ip);
remote_address.sin_port = htons(remote_port);

connect(remote_socket, (struct sockaddr *) &remote_address, sizeof(remote_address));
```

```
  bytes_env = send(remote_socket, message, message_length, 0);

  shutdown(remote_socket,SD_SEND);
  closesocket(remote_socket);

  WSACleanup();
  system("PAUSE");

  return 0;
}
```