

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Informática

Dissertação de Mestrado

Um Protocolo Multicast de Distribuição de
Conteúdo em Redes Ad Hoc Móveis

Sidney Doria

Campina Grande, Paraíba, Brasil

Agosto - 2008

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Informática

Um Protocolo Multicast de Distribuição de
Conteúdo em Redes Ad Hoc Móveis

Sidney Doria

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande -
Campus I como parte dos requisitos necessários para obtenção do grau
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores e Sistemas Distribuídos

Marco Aurélio Spohn

(Orientador)

Campina Grande, Paraíba, Brasil

©Sidney Doria, 15/08/2008

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG.

Doria, Sidney Santos.

D696p Um protocolo multicast de distribuição de conteúdo em redes ad hoc móveis
/ Sidney Santos Doria - Campina Grande,
UFCG, 2008.

51f.

Dissertação (Mestrado em Ciência da Computação) - Universidade Federal
de Campina Grande. Centro de Engenharia Elétrica e Informática.

Referências.

Orientador: Prof. Dr. Marco Aurélio Spohn.

1 - Redes sem fio 2 - Ad Hoc

3 - Redes P2P 4 - Multicast I - Título

CDU 004.7(81)

**“UM PROTOCOLO MULTICAST DE DISTRIBUIÇÃO DE CONTEÚDO EM
REDES AD HOC MÓVEIS”**

SIDNEY SANTOS DORIA

DISSERTAÇÃO APROVADA EM 15.08.2008



PROF. MARCO AURÉLIO SPOHN, Ph.D
Orientador



PROF. FRANCISCO VILAR BRASILEIRO, Ph.D
Examinador



PROF. RENATO MARIZ DE MORAES, Ph.D
Examinador

CAMPINA GRANDE – PB

Resumo

A busca por novas possibilidades de sinergia entre as redes entre-pares (P2P) e as redes móveis *ad hoc* (MANETs) tem ocupado muitos pesquisadores. Alguns trabalhos tentaram portar redes P2P de compartilhamento de conteúdo sobre MANETs. Entretanto, pondera-se neste trabalho que portar redes P2P diretamente para MANETs pode não ser a melhor solução. Propõe-se então um novo protocolo de distribuição de conteúdo para MANETs: o Peer-to-MANET. O diferencial do Peer-to-MANET é considerar plenamente as características e problemas de uma MANET, mantendo-se os mecanismos básicos de uma rede P2P. Peer-to-MANET foi implementado e avaliado no simulador de redes NS-2 e seu código está disponível publicamente. Para o roteamento foi escolhido o protocolo PUMA, pelo seu desempenho superior. Como contribuição adicional, PUMA foi implementado no NS-2 e seu código também está disponível publicamente. Além da confiabilidade intrínseca à sua especificação, os resultados de avaliação do Peer-to-MANET indicaram se tratar de um protocolo robusto e eficiente.

Abstract

The search for new synergies between Mobile Ad Hoc Networks (MANETs) and Peer-to-Peer (P2P) networks has engaged many researchers. Some works have tried to implement P2P content distribution networks over MANETs. However, we consider that such a direct implementation of P2P networks in MANETs could not be the best choice. We propose a new content distribution protocol for MANETs: Peer-to-MANET. Peer-to-MANET takes into account the main characteristics and problems of MANETs, while keeping the basic concepts of P2P networks. Peer-to-MANET was implemented and tested in the NS-2 network simulator, and its source code is publicly available. PUMA was chosen as the Peer-to-MANET routing protocol, due to PUMA's best performance compared to the most representative multicast routing protocols. As an additional contribution to this work, PUMA was implemented in NS-2 and its source code is publicly available too. Besides its reliability by design, extensive simulation results show that Peer-to-MANET is efficient and robust.

Dedico esta dissertação à memória da minha avó Valdete, que em leito de morte, com sua sapiência peculiar, me fez prometer que estudaria para ser um "doutor".

Agradecimentos

Agradeço inicialmente a minha família. Sem a ajuda dos meus pais, sem meus irmãos, certamente não chegaria onde estou agora.

Agradeço demais a minha esposa amada. Priscila foi fundamental para sobreviver a tudo isso. Não foi fácil. Não há reparação para o que deixamos de viver. Priscila, amo você!

Agradeço ao meu orientador. Marco Aurélio me presenteou com uma nova oportunidade, a qual não estou certo se tinha direito, mas que muito fiz por merecer. Mantive minha palavra Spohn!

Agradeço aos meus colegas de mestrado. Guga e Milena, não tenho palavras para compensar o seu investimento e sua torcida por mim. Tenho uma dívida impagável com vocês! Agradeço a "dona Zida", mãe de Guga, pela hospitalidade de mãe. Degas e Celso me ajudaram sobremaneira. Agradeço muito por tudo. Até pelas maluquices intermináveis. A uma lista interminável de colegas-amigos, os quais não discorro pelo tamanho risco de esquecer e magoar, muito obrigado!

Agradeço aos colegas do TRE-PB/NATT-CG. Foi tão pouco tempo, mas vivi uma ótima experiência. Boas risadas, desafios de trabalho únicos. Gente de primeira. Foi massa!

Agradeço ao professor Raimundo Machado da UFS pelas aulas formais e informais de programação, sem preço. P2MAN carrega sua influência.

A todos os que mencionei e aos que esqueci, o mesmo muito obrigado!

Conteúdo

1	Introdução	1
2	Redes Ad Hoc Móveis e Redes Entre-Pares	4
2.1	Redes Ad Hoc Móveis	4
2.1.1	Protocolos de Roteamento Unicast	6
2.1.2	Protocolos de Roteamento Multicast	8
2.2	Redes Entre-Pares	9
2.2.1	Distribuição de Conteúdo	12
2.3	Distribuição de Conteúdo Entre-Pares sobre MANETs	12
3	Peer-to-MANET	16
3.1	Visão Geral	16
3.2	Busca de Conteúdos	18
3.3	Metadados	19
3.4	Fracionamento de Conteúdos	20
3.5	Transmissão de Conteúdos	20
3.6	Modo de Reparação	22
3.7	Reciclagem de Endereços Multicast	23
3.8	Considerações sobre Desempenho	23
3.8.1	Mensagens de Controle	23
3.8.2	Protocolo de Transporte	24
4	Avaliação do Peer-to-MANET	26
4.1	Visão Geral	26
4.2	Ambiente das Simulações	26

4.3	Cenários	27
4.3.1	Variação da Quantidade de Nós Interessados	27
4.3.2	Variação do Tamanho do Conteúdo	30
4.3.3	Variação da Mobilidade	32
5	Conclusão	33
5.1	Resumo das Atividades	33
5.2	Considerações Finais	34
5.3	Resumo das Contribuições	34
5.4	Trabalhos Futuros	35
	Referências Bibliográficas	36
A	Instalação do P2MAN no NS-2	47
A.1	Alteração Necessária ao Arquivo <code>Makefile</code>	47
A.2	Visualização da Simulação Exemplo	48
B	Instalação do PUMA no NS-2	49
B.1	Alterações Necessárias ao Arquivo <code>Makefile</code>	50
B.2	Alterações Necessárias ao Arquivo <code>ns-lib.tcl</code>	50
B.3	Alterações Necessárias ao Arquivo <code>packet.h</code>	50
B.4	Visualização da Simulação Exemplo	51

Lista de Acrônimos

- ABR - Associativity-Based Routing*
- AODV - Ad Hoc On-Demand Distance Vector Routing*
- ARA - Ant-Colony Based Routing Algorithm for MANETs*
- CGSR - Clusterhead Gateway Switch Routing*
- CBRP - Cluster Based Routing Protocol*
- DARPA - Defense Advanced Research Projects Agency*
- DDR - Distributed Dynamic Routing*
- DFR - Direction Forward Routing*
- DHT - Distributed Hash Table*
- DREAM - Distance Routing Effect Algorithm for Mobility*
- DSDV - Destination-Sequenced Distance Vector*
- DSR - Dynamic Source Routing*
- DST - Distributed Spanning Tree*
- FORP - Flow Oriented Routing Protocol*
- FSR - Fisheye State Routing*
- FSLs - Fuzzy-Sighted Link State*
- GSR - Global State Routing*
- HSR - Hierarchical State Routing*
- IP - Internet Protocol*
- LAR - Location-Aided Routing*
- LMR - Lightweight Mobile Routing*
- MANET - Mobile Ad Hoc Network*
- MFR - Most Forward within Radius*
- NACK - Negative Acknowledgment*

OLSR - *Optimized Link State Routing*

OSPF - *Open Shortest Path First*

P2MAN - *Peer-to-MANET*

P2P - *Peer-to-Peer*

PRNET - *Packet Radio Network*

PUMA - *Protocol for Unified Multicast Through Announcements*

RDMA - *Relative Distance Microdiscovery Ad Hoc Routing*

RIP - *Routing Information Protocol*

ROAM - *On-demand Loop-Free Multipath Routing*

SLURP - *Scalable Location Update-based Routing Protocol*

SSA - *Signal Stability Adaptive Routing*

START - *Source-Tree Adaptive Routing*

SURAN - *Survivable Adaptive Radio Networks*

TBRPF - *Topology Broadcast based on Reverse-Path Forwarding*

TCP - *Transmission Control Protocol*

TORA - *Temporally-Ordered Routing Algorithm*

UDP - *User Datagram Protocol*

WRP - *Wireless Routing Protocol*

ZHLS - *Zone-Based Hierarchical Link State*

ZRP - *Zone Routing Protocol*

Lista de Figuras

3.1	Cenário com Nós P2MAN	19
3.2	Busca de Conteúdo	19
3.3	Nós Aptos a Receber Conteúdo	21
3.4	Nós Recebendo o Conteúdo	21
3.5	Exemplo de Reparação	23
3.6	Exemplo de Múltipla Reparação	23
4.1	Relação entre Tempo de Entrega e Número de Nós Interessados	29
4.2	Relação entre Vazão de Entrega e Número de Nós Interessados	30
4.3	Relação entre Tempo de Entrega e Tamanho do Conteúdo	31
4.4	Relação entre Tempo de Entrega e Mobilidade dos Nós	32

Lista de Tabelas

4.1	Ambiente de Simulações do P2MAN	28
-----	---	----

Capítulo 1

Introdução

Devido à natureza descentralizada de uma rede *ad hoc* móvel (MANET), sistemas implantados nesse tipo de rede precisam ser projetados para trabalhar com essa infra-estrutura. Por isso, a comunidade científica tem atentado para outro tipo de rede descentralizada: as redes entre-pares (P2P). O interesse científico na integração de redes P2P e MANETs parte do princípio de que elas comungam de muitas características chave, tais como auto-organização, descentralização e mudanças freqüentes de topologia causadas pela mobilidade (MANETs) e por entrada e saída de nós à rede (P2P e MANETs) [9], [86]. Além disso, ambas as redes utilizam conexão entre nós distantes via comunicação salto-a-salto (i.e., roteamento).

As redes P2P e MANETs têm intersecção num desafio fundamental: prover comunicação de forma descentralizada em ambiente dinâmico. Assim, é razoável considerar que soluções de problemas das MANETs podem ser aproveitadas em redes P2P e vice-versa. Trata-se de um desafio explorar as possibilidades de sinergia entre as duas redes, na busca de novas aplicações [100], [91], [101].

Considerando essa diretriz sinérgica, as primeiras publicações marcaram implementações de redes P2P populares sobre MANETs [34], [53], [50], [25], [67]. As publicações subsequentes fizeram uma análise dessas implementações e de seus resultados. Os novos trabalhos contestaram o desempenho obtido nos trabalhos anteriores e demonstraram que o problema deriva, essencialmente, das dissonâncias entre os dois modelos [19], [25].

Na Internet, tipicamente, as redes P2P de compartilhamento de conteúdo são de camada de aplicação, valem-se de nós estáticos, usam comunicação *unicast* e ocasionalmente usam *broadcasts* virtuais (através de vários *unicasts*). Em contraste, as MANETs são referenciadas como camada de rede, usam *broadcasts* físicos e os nós são móveis, com conexões montadas dinamicamente [102].

O foco das pesquisas passou então a como otimizar as implementações P2P, de modo a considerar melhor as peculiaridades das MANETs. Surgiram então algumas propostas de ferir os encapsulamentos do modelo OSI para que uma rede P2P pudesse usar informações diretamente das camadas inferiores, na MANET [19], [25], [75], [54], [64].

Neste trabalho defende-se que a aglutinação das duas tecnologias precisa ser mais habilidosa. Considerar a implementação de uma rede P2P, nos moldes em que ela é utilizada na Internet, sobre uma MANET é ignorar o problema principal de cada uma das duas infra-estruturas de rede. Na Internet o problema fundamental é a escala, na ordem de milhões de nós. Nas MANETs os problemas fundamentais são a dinâmica e os recursos escassos. As redes P2P primam por soluções que, para as MANETs, podem ser desnecessárias e têm falhado em proporcionar um desempenho adequado.

Há exemplos que corroboram tais conjecturas. Na Internet, apenas um subconjunto dos roteadores implementam *multicast*. Isso justifica porque as redes P2P não adotam *multicast* para melhorar o desempenho de entrega de conteúdos. Bem assim têm sido implementadas as redes P2P sobre MANETs: sem se valerem de *multicast*. Em teoria, havendo mais de um interessado para um mesmo conteúdo, o uso de *multicast* pode melhorar o desempenho de entrega desse conteúdo. Também, as redes P2P implementam soluções de localização de conteúdo em grande escala, como *Distributed Hash Table* (DHT). É possível que tal mecanismo não se justifique numa escala cuja ordem é de dezenas ou centenas de nós.

Considerando o exposto em trabalhos anteriores e as ponderações realizadas, identificou-se lacuna para um mecanismo eficiente de distribuição de conteúdo, adequado às características e à escala típica de uma MANET. O objetivo deste trabalho é implementar um protocolo

de distribuição de conteúdo, adequado para redes *ad hoc* móveis, valendo-se do uso de *multicast* para melhorar o desempenho de transmissão.

O protocolo proposto é o Peer-to-MANET. Alguns de seus princípios de funcionamento são baseados em redes P2P de distribuição de conteúdo, em especial a rede BitTorrent [18]. São realizados experimentos de validação no simulador de redes NS-2 [65]. O protocolo de roteamento PUMA [96], pelo seu desempenho superior, é usado como base para os experimentos. O desempenho de entrega é testado em diversos cenários, variando-se valores significativos como número de nós, tamanho dos conteúdos e mobilidade. Os resultados demonstram que Peer-to-MANET distribui conteúdos de forma eficiente e robusta, em MANETs.

O restante do trabalho está organizado como segue. O Capítulo 2 detalha o estado da arte em MANETs e redes P2P. Além disso, são citados alguns trabalhos relevantes de serviços P2P sobre MANETs. No Capítulo 3 o protocolo Peer-to-MANET é tecnicamente detalhado. O Capítulo 4 versa sobre os resultados obtidos em experimentos de avaliação. Finalmente, no Capítulo 5, os resultados são discutidos e é formulada uma conclusão.

Capítulo 2

Redes Ad Hoc Móveis e Redes Entre-Pares

2.1 Redes Ad Hoc Móveis

Em 1972 a Agência de Projetos Avançados de Pesquisa para Defesa (DARPA) dos EUA iniciou uma pesquisa para avaliar a viabilidade de se criar uma infra-estrutura de telecomunicação confiável, baseada em comutação de pacotes de rádio, então intitulada PRNET. A PRNET foi motivada pela necessidade de se prover acesso a terminais de computador para estações móveis, bem como prover computação em ambientes que precisam de celeridade no processo de montagem e desmontagem (e.g., projetos militares) [77], [48], [49].

Os projetos envolvidos incluíam dispositivos de rede, protocolos de roteamento e protocolos para gerenciamento automático de rede distribuída. A PRNET utilizava uma combinação dos protocolos ALOHA [2] e CSMA [92], [93] para ter acesso ao meio de transmissão e um protocolo de roteamento próprio, baseado em vetor de distância [77], [48]. Nos anos oitenta o projeto PRNET foi incorporado ao projeto de redes de rádio adaptativas e resilientes, SURAN [30].

Nos anos noventa os *notebooks* e outros dispositivos ubíquos se tornaram populares. Iniciou-se então uma nova fase das redes *ad hoc* móveis. A idéia de uma coleção de dispositivos móveis em uma rede sem infra-estrutura formal foi proposta em duas conferências

[45], [72]. O IEEE criou um subcomitê para definir um protocolo de comunicação *ad hoc* móvel, que em 1997 resultou no IEEE 802.11 [23], [22], [42] e na mudança oficial do termo **redes de pacotes via rádio** para **redes ad hoc móveis** (MANET) [77].

Pela definição formal dada pelo Internet Engineering Task Force [21], uma *rede ad hoc móvel* é um sistema autônomo de roteadores móveis (e *hosts* associados) interligados por conexões sem fio, cuja união forma um grafo arbitrário. Os roteadores são livres para se mover aleatoriamente e se organizar arbitrariamente; portanto, a topologia sem fio da rede pode mudar rapidamente e de forma imprevisível. Esse tipo de rede pode operar de maneira independente ou ser conectada a uma rede maior.

Numa MANET os nós desejam se comunicar, mas presume-se que não há possibilidade de comunicação direta entre todos os nós, por limitações como baixa potência de sinal disponível, modos de conservação de energia e obstáculos à transmissão que podem interromper a comunicação entre dois nós quaisquer. Nós de uma MANET se comportam como *hosts* que devem descobrir dinamicamente com que outros nós é possível ter comunicação direta e são requisitados a agir como roteadores, repassando pacotes de dados entre nós que não teriam alcance de transmissão entre si [77], [48], [34], [100], [49].

Há várias aplicações para MANETs. Em operações militares os nós podem ser soldados, tanques, aviões, etc. No mundo civil, os nós podem formar uma rede urbana de taxis, uma rede de policiamento e até uma rede de combate a incêndios com sensoriamento de locais críticos [95]. A motivação chave que tem formado MANETs é sua facilidade de implantação.

Um cenário que ilustra o potencial das MANETs é o de desastres. Num desastre hipotético em que haveria áreas de destroços e a busca por sobreviventes deveria ser célere, uma MANET poderia ser montada para sensoriamento dos locais. Os nós sensores cobririam a região afetada e repassariam as informações aos nós interessados (e.g., defesa civil). Além de sensoriamento, poder-se-ia utilizar uma rede de repasse de informações *ad hoc*, a fim de estabelecer rapidamente comunicação, com mobilidade assegurada aos nós.

2.1.1 Protocolos de Roteamento Unicast

MANETs empregam a estrutura tradicional do TCP/IP para comunicação fim-a-fim entre nós. Entretanto, devido às limitações de recursos, à mobilidade e às condições variáveis dos canais de rádio, as conexões entre os nós não são confiáveis. Essa peculiaridade é um desafio que tem desprendido grandes esforços de pesquisadores para criar protocolos mais eficientes para as MANETs. Os protocolos de roteamento *ad hoc* móveis já propostos são classificados em **proativos**, **reativos** e **híbridos** [1], [81], [41].

Os protocolos proativos, também chamados de protocolos de rota disponível, são similares a protocolos de roteamento de redes cabeadas (e.g., OSPF [60], RIP [38]). Tais protocolos tentam manter uma tabela consistente e atualizada de rotas para toda a rede, em cada nó. São exemplos de protocolos de roteamento proativos: DSDV [72], WRP [61], GSR [12], FSR [69], [33], STAR [32], FSLs [83], DREAM [7], CGSR [13], HSR [70], OLSR [17], [16] e TBRPF [66].

É fácil perceber que a abordagem proativa de roteamento tem como vantagem a celeridade para remessa de um pacote qualquer, devido à grande probabilidade de já existir uma tabela de roteamento pronta e atualizada no nó remetente. Também, é possível perceber que protocolos de rotas disponíveis geram considerável sobrecarga na rede para manter atualizadas as tabelas de roteamento, quer os nós precisem ou não [1], [81], [59], [41].

Os protocolos reativos, também chamados de protocolos sob demanda, são baseados em eventos. Nós iniciam descobertas de rotas provocadas por eventos (e.g., enviar dados a outro nó). Protocolos de roteamento reativos são subclassificados em protocolos **salto-a-salto** (*hop-by-hop*) e protocolos de **rota definida na origem** (*source-based*).

Os protocolos salto-a-salto disparam pequenas descobertas de rota em cada nó, montando assim a tabela de roteamento durante as interações com os nós roteadores, salto-a-salto. Os protocolos de rota definida na origem iniciam a descoberta de rotas por completo, antes de enviar o pacote, e põem a rota completa dentro do pacote para que seja obedecida pelos nós roteadores. São exemplos de protocolos de roteamento reativos: DSR [47], [46], AODV

[71], ROAM [76], LMR [20], TORA [68], ABR [94], SSA [26], RDMAR [4], LAR [51], [52], ARA [10], FORP [89], CBRP [43] e DFR [58].

Os protocolos reativos têm a vantagem da economia de recursos, pois não precisam manter e atualizar tabelas de roteamento da rede inteira em cada nó. Outra vantagem é a maior velocidade de adaptação às mudanças de topologia, já que cada interação promove novas descobertas de rotas. Uma desvantagem desse tipo de protocolo é a alta latência de remessa de pacote, visto que antes de tentar se comunicar, um nó deve iniciar uma descoberta de rota para seu destinatário [1], [81], [59], [41].

Protocolos híbridos são uma nova geração de protocolos que têm em sua natureza considerações proativas e reativas. Tais protocolos são modelados para aumentar a escalabilidade, por permitir que nós próximos possam formar um eixo de comunicação que reduz sobrecargas de descobertas de rotas. A mescla das técnicas proativa e reativa está no fato de se manter proativamente as rotas próximas e buscar as rotas mais distantes de forma reativa, sob demanda. Muitos protocolos híbridos lançam mão do particionamento das redes em zonas, árvores ou agrupamentos para definir a noção de distância entre nós [1], [81], [59], [41]. São exemplos de protocolos de roteamento híbridos: ZRP [36], [37], ZHLS [44], SLURP [99], DST [74] e DDR [63].

Além dos protocolos supracitados, alguns pesquisadores investigam roteamento assistido por posicionamento geográfico. Em tais algoritmos, nós tipicamente sabem a posição física de outros nós. Essa informação é utilizada para criar agrupamentos e para o planejamento do roteamento de um pacote. Há várias abordagens que compreendem o uso de informação geográfica para roteamento. São exemplos de protocolos assistidos por posicionamento geográfico: MFR [90], LAR [51], DREAM [7]. Um artigo [88] propõe o uso de vários métodos geométricos para melhorar o roteamento e, em alguns casos, até dispensa a necessidade de que o nó saiba a localização geográfica exata de outros nós. Há também uma abordagem cartesiana para o roteamento [29].

2.1.2 Protocolos de Roteamento Multicast

O objetivo de um protocolo de roteamento *multicast* para MANETs é suportar a disseminação de informação de um nó remetente para todos os nós destinatários em um grupo *multicast*, enquanto tenta utilizar eficientemente a banda disponível, na presença de mudanças de topologia. Vários protocolos *multicast* já foram propostos. São exemplos de protocolos *multicast*: MAODV [80], CAMP [31], ODMRP [57] e PUMA [95], [96]. As abordagens podem ser classificadas em protocolos baseados em árvore (*tree-based protocols*) e protocolos baseados em malha (*mesh-based protocols*).

Para despachar pacotes dos nós originários para nós destinatários de um grupo *multicast*, um protocolo baseado em árvore estabelece e mantém uma árvore *multicast* compartilhada ou múltiplas árvores *multicast* baseadas nas origens (uma para cada remetente). Por se tratar de uma árvore (*spanning tree*), o caminho de repasse do pacote *multicast*, entre o remetente e cada destinatário, é único (i.e., não utiliza redundância de rotas). Em contraste, um protocolo baseado em malha mantém uma malha contendo todos os destinatários do grupo *multicast*. Tal abordagem cria múltiplas rotas redundantes para envio dos pacotes *multicast* aos participantes do grupo. Dois exemplos são o MAODV (árvore) [80] e o PUMA (malha) [95], [96].

MAODV mantém uma árvore compartilhada para cada grupo *multicast*, formada somente por roteadores e destinatários. Nós remetentes que desejarem enviar um pacote ao grupo adquirem rotas para o grupo, sob demanda, de forma similar ao AODV [71]. Cada árvore *multicast* tem um líder de grupo, que é o primeiro receptor a se juntar à árvore. O líder periodicamente transmite um pacote de anúncio (*hello*) para se certificar das conexões ativas. Nós se juntam ao grupo através de um pacote especial de roteamento.

PUMA mantém uma malha com os nós participantes. O líder de cada grupo *multicast* (*core*) é eleito em vez de ser determinado. A manutenção da malha é feita pelo líder. PUMA utiliza um único tipo de pacote de controle: o anúncio. Os anúncios são transmitidos através de *broadcast* não confiável. O líder é o único nó que transmite anúncios periodicamente. No PUMA um único anúncio pode sinalizar mais de uma atividade para os nós (e.g., eleger líder, manter a malha, pedido de inserção/remoção de participante). PUMA não utiliza uni-

cast para anúncios porque não assume nenhum protocolo de roteamento *unicast*. O objetivo é evitar sobrecarga, detectada em simulações, causada pelas conexões *unicast* sob presença de mobilidade.

Um anúncio viaja pela rede orientando cada nó sobre quais são seus vizinhos com o caminho mais curto para o líder. Dessa forma, pacotes destinados a um grupo *multicast* podem ser roteados rapidamente para o líder do grupo. O líder então repassa o pacote ao grupo por um *broadcast* que não é propagado para fora do grupo. O líder não é um ponto de falha severo, pois um novo líder é eleito rapidamente, através de um algoritmo distribuído. Se um nó perceber que é caminho mais curto até o líder, para outros nós, torna-se um roteador para esses nós, mesmo que não participe do grupo. Com essas características, PUMA busca um baixo custo de mensagens de controle e uma rápida distribuição dos pacotes.

PUMA obteve desempenho superior aos protocolos anteriores mais expressivos (i.e., MAODV e ODMRP) [95], [96]. Essas características fazem com que PUMA seja um forte indicado para experimentos. Por isso, PUMA foi selecionado como protocolo de roteamento *multicast* nos experimentos do Capítulo 4. Entretanto, não havia uma implementação do PUMA disponível para a comunidade científica. Assim, primeiramente foi necessário implementar o PUMA, a partir de sua especificação. A implementação do PUMA para NS-2 foi feita e está disponível publicamente no sítio Source Forge, em <http://sourceforge.net/projects/puma-adhoc/>. Um tutorial de instalação do PUMA está descrito no Apêndice B. Ressalte-se que a implementação do PUMA é uma importante contribuição deste trabalho para a comunidade científica porque incentivará novos experimentos com esse protocolo de alto desempenho.

2.2 Redes Entre-Pares

A definição formal de redes entre-pares (P2P) ainda é controversa na literatura. Uma definição é que uma rede distribuída é dita rede P2P se seus participantes (nós) compartilham parte de seus recursos (e.g., processador, armazenamento, banda de rede) para prover o serviço ou conteúdo oferecido por esta rede. Os nós são então fornecedores e consumidores de

recursos. Quando o acesso aos recursos é feito diretamente entre os nós e a rede não depende de nenhum nó específico, diz-se que se trata de uma rede P2P pura. Quando existe um nó específico, necessário ao funcionamento da rede, diz-se que esta é uma rede P2P híbrida [85].

As redes P2P que se popularizaram de início foram as de compartilhamento de arquivos (e.g., Napster [62], Gnutella [35], Freenet [15], Pastry [79], Chord [87], BitTorrent [18]). Tais redes provêm endereçamento e roteamento baseado em conteúdo, o que não é possível na camada IP. Um segundo exemplo que bem ilustra o potencial das redes P2P são as redes de *multicast* em tempo real, que contornam a falta de suporte *multicast* em roteadores IP na Internet [34], [100].

De forma genérica, uma rede P2P pode prover serviços que são desejáveis em sistemas distribuídos, tais como balanceamento de carga, tolerância a falhas, localização de objetos distribuídos e redundância. Redes P2P têm demonstrado ser um paradigma útil para a construção de sistemas distribuídos escaláveis, com excelente resiliência e com autonomia dada aos nós para se inserirem e se removerem da rede [100].

Devido ao grande número de redes P2P existentes hoje e de suas diferentes propriedades, não existe uma classificação única na literatura. Podem ser considerados alguns critérios de classificação. De acordo com sua data de apresentação e propósito podem-se classificar as redes P2P em três gerações [5], [97], [104]:

PRIMEIRA GERAÇÃO

Redes P2P mais antigas como Napster [62] e Gnutella [35]. Destacam-se pela facilidade de implantação. São redes muito simples, mas têm escalabilidade ruim e baixa eficiência de consultas ao serviço.

SEGUNDA GERAÇÃO

As redes P2P de segunda geração são caracterizadas pelo uso da técnica de tabela de espalhamento distribuída (DHT) para prover eficiência nas consultas ao serviço, balanceamento de carga e garantia determinística (assumindo inexistência de falhas) em resultados de pes-

quisas na rede. Entretanto, tais redes não detêm boa resiliência, principalmente a ataques maliciosos.

TERCEIRA GERAÇÃO

As propostas de redes P2P mais recentes [28], [82], [24], [18] buscam prover alta resiliência, assumindo que cada nó tem uma probabilidade de falha. A técnica comum para prover resiliência inclui replicação de objetos, expansão do número de conexões entre os nós e estruturas especiais de topologia. A rede P2P de distribuição de conteúdo mais popular na atualidade é o BitTorrent. Segundo o sítio BigChampagne [8] existiam aproximadamente 10 milhões de usuários simultâneos em agosto de 2005. Para exemplificar a evolução, o BitTorrent tem estrutura de indexação de conteúdos descentralizada entre os *trackers*. O Napster utiliza um servidor central para essa tarefa.

De acordo com as diferenças entre os mecanismos de busca de objetos e a sua topologia lógica, as redes P2P podem ser classificadas em três tipos [5], [97], [104]:

CENTRALIZADAS

Todos os índices de objetos são mantidos em um servidor central, sob alguma forma de par <chave-do-objeto, endereço-do-nó>. Cada nó que se insere na rede deve avisar esse servidor sobre a sua inserção e seus objetos. Então, os nós podem pesquisar sobre determinado objeto, apenas consultando o servidor. Este tipo de arquitetura é bastante simples e fácil de ser implantada. Entretanto, existe o problema do ponto de falha singular (i.e., servidor). Um exemplo clássico desse tipo de arquitetura é o Napster [62].

DESCENTRALIZADAS e DESESTRUTURADAS

Uma consulta a um objeto é distribuída e a topologia lógica é sempre uma malha aleatória e desestruturada. A consulta é executada salto-a-salto através dessa malha, até obter um resultado (sucesso/falha) ou ter o tempo de consulta expirado (*timeout*). Um exemplo clássico desse tipo de arquitetura é o Gnutella [35]. Gnutella não tem ponto de falha singular, mas a eficiência de consulta a objetos pode ser baixa.

DESCENTRALIZADAS e ESTRUTURADAS

As consultas a objetos são distribuídas, mas a topologia lógica da rede é uma **malha estruturada** [14], [79], [103], [105], um **anel** [87], uma **toróide d-dimensional** [78], uma **borboleta**, [28], [82], [24] e outras possibilidades. Essas topologias estruturadas são normalmente construídas usando-se uma técnica de DHT. A consulta a objetos também é executada salto-a-salto através da topologia estruturada, e, na inexistência de falhas, nela é determinístico que em alguns saltos se obtenha uma resposta [5].

2.2.1 Distribuição de Conteúdo

Pessoas podem se valer de um sistema de distribuição de conteúdo para se comunicar, através de dispositivos computacionais. O processo de comunicação pode ser através da entrega de arquivos digitais (e.g., um documento digital) ou através de fluxos contínuos de informação (e.g., notícias de rádio em formato digital). Do ponto de vista computacional, a idéia é que nós de uma rede representem fornecedores de informação e interessados nesta informação.

Sistemas entre-pares de distribuição de conteúdo são programas criados para compartilhar dados digitais entre usuários. Tais sistemas variam do mais simples aplicativo de compartilhamento direto ao sistema mais complexo de repositório distribuído. São capazes de realizar publicação, organização, indexação, pesquisa, atualização e recuperação de informações. À medida que se tornam mais sofisticados, os sistemas de distribuição de conteúdo têm mais funções adicionadas. Incluem-se anonimidade, segurança, equidade de compartilhamento, desempenho superior e escalabilidade aprimorada [5].

Arquiteturas de distribuição de conteúdo apresentam um desafio em particular no quesito segurança, para prover os níveis aceitáveis de disponibilidade, privacidade, confidencialidade, integridade e autenticidade [5], [104].

2.3 Distribuição de Conteúdo Entre-Pares sobre MANETs

As redes entre-pares, ao longo de sua evolução, lograram êxitos que têm cativado a atenção de pesquisadores. Em especial, algumas redes entre-pares de distribuição de conteúdo

alcançaram uma popularidade de milhões de usuários na Internet. Isso tem sustentado um cenário de grande escala, dinâmico e hostil o suficiente para comprovar sua escalabilidade e sua resiliência.

Também, propriedades como auto-organização, descentralização e resiliência às mudanças de topologia ressaltaram sua similaridade ao *modus operandi* das MANETs. Intuitivamente, considerou-se que soluções de problemas das MANETs poderiam ser aproveitadas para as redes P2P e vice-versa. Percebeu-se então um desafio em explorar as possibilidades de sinergia entre as duas redes, na busca de novas aplicações [100], [91], [101].

Nesse intuito, os primeiros trabalhos científicos expuseram implementações de redes P2P populares sobre MANETs [34], [53], [50], [25], [67]. Em um segundo momento, as novas publicações contestaram tais implementações e seus resultados. Os novos experimentos contestaram o desempenho obtido e demonstraram que tal problema tem origem nas diferenças existentes entre os dois modelos de rede [19], [25].

De uma forma sucinta, na Internet, as redes P2P de distribuição de conteúdo tipicamente são de camada de aplicação, pautam-se em nós estáticos, usam comunicação *unicast* e ocasionalmente usam *broadcasts* virtuais (através de vários *unicasts*). Em comparação, as MANETs são referenciadas como camada de rede, usam *broadcasts* físicos e os nós são móveis, com conexões montadas dinamicamente [102].

Considerando-se essas diferenças e que as redes P2P são sobrepostas, é possível visualizar que o processo de envio de pacotes pela rede P2P, na camada de aplicação, pode não ser sinônimo do processo de envio de pacotes pela MANET, na camada de rede. Por exemplo, imagine-se que determinado nó de uma rede P2P deseja enviar um pacote a todos os seus vizinhos. É possível que na camada de rede a suposta vizinhança seja, na verdade, composta por nós físicos distantes. Dessa forma haveria envio de pacotes através de toda a extensão da rede, até chegar aos nós longínquos, pelo método salto-a-salto.

Os esforços dos pesquisadores se voltaram às adaptações necessárias para que uma rede

P2P pudesse funcionar sobreposta a uma MANET, mas considerando melhor as suas peculiaridades. Surgiram então propostas de ferir os encapsulamentos do modelo OSI para que uma rede P2P pudesse usar informações diretamente das camadas inferiores, na MANET [19], [25], [75], [54], [64].

Por se tratarem de tecnologias emergentes sendo unidas, ainda há certa latência nas pesquisas e não há um grande número de publicações de alta qualidade. Destacam-se:

- Estudos de desempenho de protocolos de roteamento de MANETs sob um sistema P2P [73], [67];
- Comparativos entre MANETs e redes P2P [9], [86];
- Argumentações no sentido de se buscar sinergia entre MANETs e redes P2P [91], [101];
- Experimentos com aplicações P2P sobre MANET [53], [50], [25];
- Experimentos que defendem o uso de implementações entre-camadas (*cross-layer*) para melhoria de desempenho [19], [25], [75], [64].

A despeito dos avanços em experimentos e teorias desenvolvidas para melhorar o desempenho das redes P2P sobre MANETs, considerar o uso de um algoritmo P2P, nativo da Internet, diretamente sobre uma MANET pode não ser a melhor solução para prover distribuição de conteúdo neste tipo de infra-estrutura de rede. Vê-se que as implementações P2P usadas nos experimentos têm as mesmas doutrinas, mas enfrentam problemas diferentes quando sobrepostas a cada uma das infra-estruturas.

O problema fundamental na Internet é a escala, cuja ordem de grandeza pode chegar a milhões de nós. Nas MANETs a escala é, geralmente, muito menor (na ordem de dezenas ou centenas de nós). Entretanto, há problemas consideráveis nas MANETs como a falta de recursos e a dinâmica (e.g., mobilidade). Alguns mecanismos próprios das redes P2P, como os algoritmos de busca de conteúdos, cujos desempenhos em larga escala são logarítmicos

(e.g., DHT), podem ser dispensáveis para uma MANET típica. Ademais, é possível que potenciais próprios das MANETs não estejam sendo explorados a contento.

Para sustentar tais conjecturas, exemplifica-se que na Internet uma pequena porção dos roteadores implementam *multicast*. Este fato explica porque as redes P2P geralmente não utilizam *multicast* para otimizar a entrega dos conteúdos. Em MANETs a adoção de *multicast* pode trazer ganho no desempenho de entrega, na condição de haver vários interessados pelo mesmo conteúdo. Essa condição não é rara à distribuição de conteúdo.

Capítulo 3

Peer-to-MANET

3.1 Visão Geral

Peer-to-MANET (P2MAN) é um protocolo de distribuição de conteúdo robusto e confiável para redes móveis *ad hoc*. P2MAN usa uma abordagem de busca e entrega similares às utilizadas pelas redes P2P. Entretanto, não se limita às doutrinas impostas pela herança da Internet, já discutidas neste trabalho, nem implementa mecanismos que não sejam estritamente necessários ao funcionamento do protocolo numa MANET. Em especial, P2MAN adota entrega de conteúdo através de *multicast*. P2MAN prossegue a convergência entre as tecnologias P2P e MANETs, entretanto rompe com uma seqüência de adaptações da Internet e propõe uma implementação nativa em MANETs.

Assim como nas redes P2P, P2MAN permite que qualquer nó, a qualquer tempo, possa entrar ou sair da rede ou iniciar suas transferências de conteúdos. P2MAN utiliza o conceito de entrega de arquivo em pedaços para aumentar a confiabilidade e o desempenho de entrega numa MANET. Como o BitTorrent, P2MAN utiliza metadados para orientar os nós receptores sobre como um conteúdo foi fracionado e a quem ele deve ser requerido. Quando um nó, proprietário de um arquivo digital, decide compartilhar, ele também decide como o conteúdo a ser compartilhado será fracionado. Nenhuma outra ação é requerida para um compartilhamento de conteúdo.

Devido à dinâmica das MANETs, o modelo de serviço utilizado é o de melhor esforço.

P2MAN não utiliza servidores de informações de arquivos. Em vez disso, utiliza o modelo *Publish / Subscribe*, onde grupos *multicast* são alocados para transferências dos conteúdos. Para mensagens de controle, é utilizado um grupo *multicast* especial, cujo endereço é convencionalizado, denominado "canal público". O canal público é um grupo *multicast* ao qual todos os nós P2MAN aderem no início de suas atividades. Quando um nó deseja um conteúdo, faz uma consulta ao canal público. Se a solicitação foi detectada por um nó detentor do conteúdo, este responderá, também no canal público, com uma mensagem que confirma a disponibilidade do conteúdo. Nessa mensagem, o nó detentor do conteúdo envia também metadados sobre o conteúdo. Após uma quantidade definida¹ de tentativas de localizar o conteúdo, o nó solicitante desistirá de procurá-lo por um tempo determinado.

Ao receber uma mensagem de resposta a uma pesquisa, o nó solicitante lê os metadados. Imediatamente ganha a informação de qual grupo *multicast* será alocado para a transmissão do conteúdo em questão. Então, o nó solicitante se junta ao grupo *multicast* indicado e envia uma mensagem de autorização para o nó remetente, no canal público. Ao receber a autorização, o nó transmissor inicia o envio dos pedaços do conteúdo. O transmissor aguardará sua autorização por um tempo definido e parametrizável. Portanto, até o momento da autorização de transmissão, o receptor pode escolher abandonar o processo. Se isso ocorrer o transmissor também abandonará o processo de envio, após o tempo determinado.

Se algum nó perder algum pedaço ou se juntar a um grupo com uma transmissão já em andamento, há a necessidade de garantir a entrega dos pedaços restantes a todos os nós. Isso é feito através de um procedimento denominado reparação, detalhado mais adiante. Um nó entrará em reparação se não receber pedaços por determinado tempo (i.e., *timeout*) e ainda restarem pedaços a receber. O nó então enviará uma mensagem ao canal público, informando os pedaços que tem, para que o transmissor envie os pedaços que restarem.

Por se tratar de uma transmissão *multicast*, ressaltam algumas diferenças das redes P2P. O protocolo de transporte utilizado é o UDP. De fato, o UDP faz parte da estratégia de desempenho, que será detalhada mais tarde. A garantia de entrega, portanto, ocorre no nível de

¹Essa quantidade é perfeitamente parametrizável numa implementação.

aplicação. Mas a diferença mais importante é que não são necessárias as diversas transmissões *unicast* simultâneas para se obter desempenho. Portanto, não será necessária nenhuma transmissão além da realizada pelo detentor do conteúdo (i.e., *seeder*). A adoção de um único transmissor *multicast* para cada conteúdo, cria um cenário que difere consideravelmente de uma rede P2P. Isso porque reduz a quantidade de transmissões simultâneas necessárias na rede e, conseqüentemente, reduz a quantidade de colisões. É fácil perceber que esse cenário elimina a necessidade de mecanismos de incentivo, visto que os nós que detêm apenas alguns pedaços (i.e., *peers*) não são necessários às transmissões.

3.2 Busca de Conteúdos

Ao iniciar, um nó P2MAN imediatamente se junta ao grupo *multicast* especial, denominado canal público. A partir daí, seus conteúdos compartilhados poderão ser acessados pelos demais nós. Quando um nó deseja um conteúdo, ele envia uma mensagem de busca ao canal público. Se já possuir alguns pedaços do conteúdo, anexa essa informação à mensagem, antes de enviá-la. Se nenhum nó responder à mensagem, ocorrem mais tentativas de busca², até que o solicitante desista do conteúdo. Quando um nó compartilhador do conteúdo recebe a solicitação, envia, em resposta, uma mensagem que carrega os metadados do conteúdo. Se mais de um nó responder à busca, apenas o primeiro será considerado. Os metadados serão melhor descritos na seção 3.3.

As figuras 3.1 e 3.2 ilustram o processo de busca de conteúdo na rede. Os nós E e K compartilham um determinado conteúdo. Em dado momento o nó M deseja tal conteúdo. M então envia um pedido de busca ao canal público. O pedido chega a ambos, E e K, que respondem também no canal público, enviando na mensagem os metadados do conteúdo. Mais adiante, o nó M escolherá o nó E para enviar o conteúdo. A resposta do nó K será desconsiderada. Isso ocorrerá devido à menor distância, em saltos, entre E e M.

²Quantidade parametrizável.



Figura 3.1: Cenário com Nós P2MAN

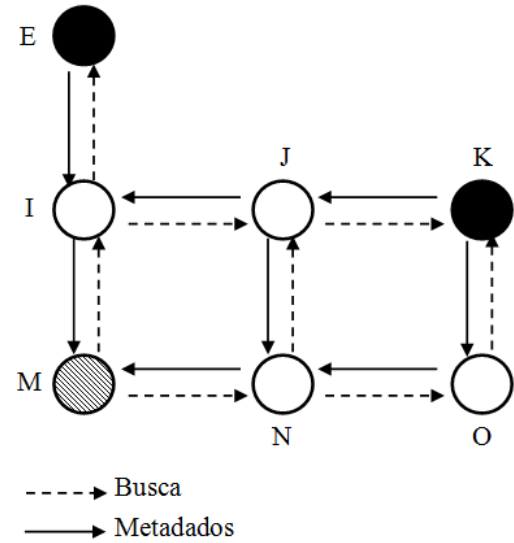


Figura 3.2: Busca de Conteúdo

3.3 Metadados

Tomando-se novamente o BitTorrent como exemplo, seu arquivo de metadados (i.e., .torrent) contém informações sobre o conteúdo, dentre outras:

- O nome do conteúdo;
- Informações sobre o servidor de metadados (i.e., *tracker*);
- O número de pedaços que compõem o conteúdo;
- O tamanho dos pedaços;
- Uma chave de verificação para cada pedaço.

De forma similar, P2MAN usa troca de metadados entre o nó transmissor e os nós interessados. No P2MAN as metadados são:

- O nome do conteúdo;

- O endereço do grupo *multicast*, determinado pelo nó transmissor, para envio do conteúdo (mais detalhes na seção 3.7);
- O número de pedaços que compõem o conteúdo;
- O tamanho dos pedaços;
- Quando necessário, uma lista de pedaços que já foram adquiridos pelo nó receptor.

3.4 Fracionamento de Conteúdos

Assim como nas redes P2P, P2MAN adota o fracionamento de conteúdos. Fracionar o conteúdo diminui a possibilidade de colisões na rede e melhora a qualidade da entrega. Ao decidir compartilhar um conteúdo, o nó que o possui decide também como será fracionado esse conteúdo. Por questão de adequação do escopo deste trabalho ao tempo disponível, nesta versão de P2MAN não serão discutidos os mecanismos para verificação de integridade dos pedaços. Entretanto, como o mecanismo de fracionamento do P2MAN é similar aos utilizados pelas redes P2P, é possível adotar uma das abordagens de verificação de integridade das redes P2P mais populares.

3.5 Transmissão de Conteúdos

Ao receber os metadados, o solicitante obtém o grupo *multicast* que será alocado para a transmissão do conteúdo. Então, o solicitante se junta a esse grupo *multicast* e envia uma mensagem autorizando um transmissor a enviar pedaços do conteúdo. Ao contrário do que ocorre nas redes P2P que usam *unicast*, a mensagem de autorização é necessária para garantir que somente uma transmissão do conteúdo esteja sendo realizada. Também, assegura que haja tempo suficiente para o nó se juntar ao grupo *multicast*, antes que o transmissor envie pedaços ao grupo. Autorizado, o transmissor envia os pedaços para o grupo *multicast* alvo. Após o envio do último pedaço, o transmissor silencia suas atividades.

As figuras 3.3 e 3.4 ilustram o processo de transmissão de conteúdos. Na figura 3.3, os

nós J e M recebem metadados de E e K e decidem³ que o nó E será seu transmissor para o conteúdo. Como ilustrado na figura 3.4, J e M ingressam no grupo *multicast* indicado por E. Os nós J e M enviam autorização de transmissão, mas, para este exemplo, a autorização de M chegará primeiro a E. Em seguida, E envia pedaços do conteúdo solicitado por M ao grupo *multicast* previamente definido. Como integrantes deste grupo *multicast*, J e M começam a receber os pedaços do conteúdo.



Figura 3.3: Nós Aptos a Receber Conteúdo

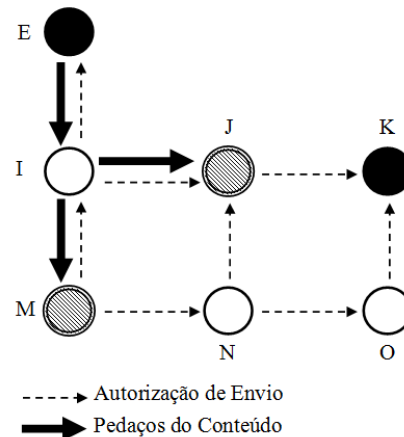


Figura 3.4: Nós Recebendo o Conteúdo

Se o nó J tivesse escolhido K e M tivesse escolhido E, duas transmissões simultâneas e independentes teriam sido iniciadas. Como P2MAN prioriza o primeiro recebimento de mensagens, este fenômeno indica que há condições de tráfego melhores entre J e K e entre K e M, o que no dado instante será benéfico para a robustez de entrega.

Ao autorizar uma transmissão, o nó receptor aciona um contador de tempo. Sempre que um pedaço chega, o contador é reiniciado. Se não houver mais pedaços a receber, o receptor silencia suas atividades. A expiração do contador, sem que todos os pedaços tenham sido adquiridos, indica que: (i) o transmissor finalizou seu envio, (ii) o transmissor saiu do alcance do receptor ou (iii) está ocorrendo falha na transmissão dos pedaços (e.g., congestionamento). Nesses caso, o receptor inicia um procedimento denominado **reparação**.

³De fato o nó J está mais próximo do nó K. Optou-se por E para demonstrar uma entrega múltipla. Interferências poderiam ter causado essa condição, como a ocorrência de colisão durante a resposta de K.

3.6 Modo de Reparação

A reparação é um procedimento para assegurar a confiabilidade na entrega do conteúdo a todos os interessados. Consiste em reiniciar o processo de obtenção do conteúdo, embora informando que pedaços o nó interessado já possui. Dessa forma o nó transmissor será instado a enviar apenas os pedaços restantes. Se um novo transmissor for selecionado, um novo grupo *multicast* será utilizado. Os nós interessados se desligarão do grupo multicast anterior. A abordagem de reparação foi criada porque as transmissões *multicast* não usam TCP para dar confiabilidade de entrega aos pacotes.

Se vários nós entrarem em reparação ao mesmo tempo, o nó transmissor receberá várias solicitações de retransmissão, porém só atenderá à primeira. Os demais nós receberão os pedaços em comum com a demanda do primeiro nó. Na rodada seguinte, os nós que precisarem de mais pedaços, entrarão em reparação novamente, num processo que converge à entrega total. A convergência ocorre em um tempo finito e proporcional ao número de pedaços em demanda que não formam intersecção entre si. Por exemplo, se o nó **A** solicitar o pedaço 2 e o pedaço 3, e o nó **B** solicitar pedaço 5, haverá duas reparações, obrigatoriamente. Mas se **A** solicitar pedaço 2 e pedaço 3, e **B** solicitar pedaço 3 também, só haverá uma reparação.

As figuras 3.5 e 3.6 ilustram o mecanismo de reparação. Na figura 3.5, ocorre uma reparação a partir do nó M. Na figura 3.6, ocorre uma reparação a partir dos nós M e J. Neste caso, como no início de uma aquisição de conteúdo, ambos enviam mensagem de busca, recebem resposta com metadados e escolhem seu transmissor. Se o transmissor for o mesmo, enviam mensagens de autorização, sendo que apenas a primeira será considerada.

O procedimento de reparação mostrou-se similar ao de outro protocolo em desenvolvimento: o NORM [55]. NORM é um protocolo de transporte para envio de dados via *multicast*, com NACK explícito. Os nós receptores enviam um NACK para o transmissor quando percebem que perderam pacotes. Para tanto, se pautam no número de seqüência dos pacotes. Após um período, o remetente envia um pacote especial de sincronismo e se prepara para receber "reclamações" dos receptores. Esse sincronismo denota um direcionamento do protocolo para fluxos contínuos cuja seqüência de dados não pode ser quebrada. Esta não é

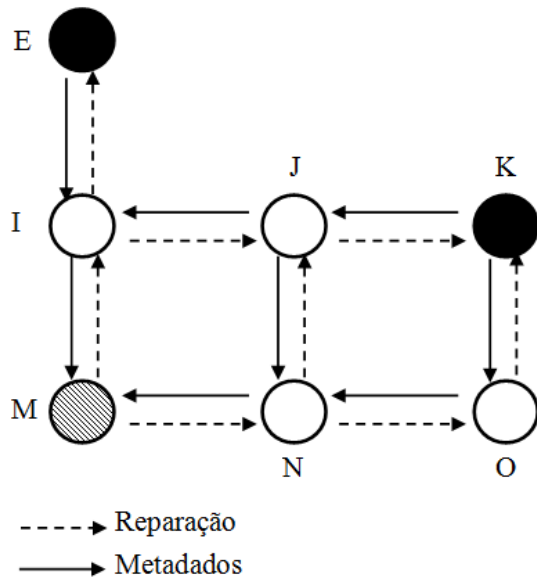


Figura 3.5: Exemplo de Reparação

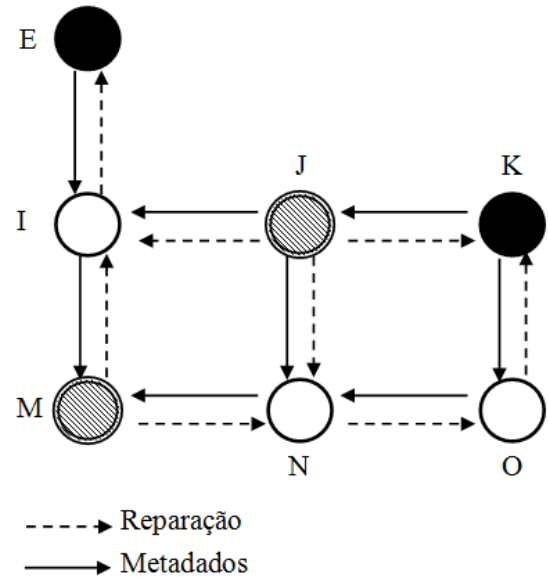


Figura 3.6: Exemplo de Múltipla Reparação

uma necessidade para entrega de conteúdos.

3.7 Reciclagem de Endereços Multicast

Com o passar do tempo é possível que se esgotem os endereços *multicast* utilizados para transmitir conteúdos. A especificação do P2MAN não cuida da exclusividade desses grupos para os conteúdos. É possível fazer envio de dois ou mais conteúdos utilizando-se um único grupo *multicast* porque as mensagens de controle são enviadas somente pelo canal público e os pedaços levam identificação nos metadados. Sendo assim, não é preciso fazer reciclagem de endereços de grupos *multicast* e o uso de endereços aleatórios resolve o problema.

3.8 Considerações sobre Desempenho

3.8.1 Mensagens de Controle

Quando vários nós interessados entram em reparação ao mesmo tempo o transmissor receberá várias solicitações de retransmissão. Apesar de atender apenas à demanda do primeiro pedido, o transmissor deve responder no canal público a todas as requisições de retransmis-

são. Isso porque alguns nós solicitantes podem ser novos solicitantes ou podem ter optado por um novo transmissor. Assim, sempre que o transmissor é instado a transmitir ou retransmitir, deve enviar os metadados para orientar o solicitante. Com isso, é possível visualizar que muitas requisições simultâneas podem atrapalhar o processo de reparação.

Entretanto, P2MAN foi concebido considerando-se as peculiaridades das MANETs. Dentre essas peculiaridades, é improvável que P2MAN seja executado em uma rede de grande escala ou de escala global. Além disso, mensagens de controle, dentre elas as de reparação, tipicamente movimentam um volume de dados muito menor que a transmissão dos conteúdos em si. Portanto, salvo em situações específicas, conjectura-se que este fator não deve prejudicar o desempenho do protocolo. Entretanto, é necessário medir a sobrecarga dos pacotes de controle do protocolo e a escalabilidade do uso do canal público. Pelo esgotamento do tempo disponível, estas análises não puderem ser feitas e constarão nos trabalhos futuros.

3.8.2 Protocolo de Transporte

O TCP foi originalmente projetado para redes cabeadas, cujas taxas de erro de bits são geralmente baixas. Ele assume que qualquer perda de pacotes está relacionada a congestionamento de rede. Quando isso ocorre, o TCP inicia seu mecanismo de controle de congestionamento e fluxo. O controle de congestionamento é feito reduzindo-se o tamanho da janela deslizante. Em seguida o TCP usa um algoritmo de partida lenta ou de prevenção de congestionamento para que a janela deslizante seja aumentada gradualmente [3].

É da natureza das MANETs ter altas taxas de erro devido às desconexões temporárias, provocadas pela dinâmica da rede (e.g., sinal baixo, particionamento). Quando o cenário de uma MANET muda, é comum ocorrerem mudanças de roteamento entre os nós. A descoberta de novas rotas provoca perda transitória de pacotes. Isso faz com que o transmissor TCP inicie o controle de congestionamento e reduza a janela deslizante. Após o ajuste de rotas, a janela continuará reduzida por algum tempo até que o mecanismo de partida lenta atue. Enquanto a janela deslizante estiver reduzida, a vazão da rede também será menor. Do ponto de vista de desempenho, essa é uma razão para evitar o TCP em MANETs [40], [98], [27], [6].

Porque P2MAN é um protocolo de distribuição de conteúdo *multicast*, faz-se necessário o uso de UDP como transporte. Dessa forma o problema de desempenho do TCP é afastado, porém a responsabilidade de garantir a entrega dos pacotes do conteúdo é deslocada para a camada de aplicação. Este é um custo razoável pela adoção de dois vetores de desempenho: um protocolo de transporte mais eficiente e o uso de *multicast*.

Capítulo 4

Avaliação do Peer-to-MANET

4.1 Visão Geral

Para conduzir a avaliação do P2MAN, a metodologia utilizada foi realizar simulações do protocolo em cenários pertinentes. Simulações são necessárias, visto que, em geral, as instituições acadêmicas não dispõem de MANETs adequadas para testes reais. A escolha do simulador foi pautada no trabalho de Hogie et al. [39]. Foi escolhido o simulador livre Network Simulator 2 (NS-2) [65]. O NS-2 é um simulador acadêmico, mantido dentre outros pela Universidade da Califórnia, cujos resultados são respeitados, e é *de facto* o simulador mais utilizado pela comunidade científica para experimentos dessa natureza.

O P2MAN foi implementado para NS-2 e a implementação está disponível publicamente no sítio Source Forge, em <http://sourceforge.net/projects/p2man/>. Um tutorial de instalação do P2MAN foi descrito no Apêndice A. Os dados das simulações foram coletados, apreciados e os resultados foram dispostos, conforme detalhamento a seguir.

4.2 Ambiente das Simulações

As simulações foram feitas com a versão 2.33 do NS-2. Vários cenários de simulação foram montados, variando-se valores relevantes como número de nós envolvidos, tamanho dos conteúdos e mobilidade. Foi verificado então o tempo de entrega de conteúdo para esses cenários. Tempo de entrega de conteúdo é definido como o tempo em que um nó transmissor

leva para entregar o conteúdo a todos os nós interessados a partir da primeira requisição.

Para cada cenário foram executadas dez rodadas de simulação, perfazendo um total de 300 rodadas de simulação. Todos os valores foram representados com um intervalo de confiança de 95%. Em cada rodada as sementes dos geradores aleatórios foram trocadas ¹ de forma que valores aleatórios (e.g., *jitter*) não se repetiram. Isso faz com que os eventos na simulação (e.g., transmissões, colisões, falhas) ocorram de forma diferente em cada rodada. Para cada tipo de cenário foram considerados os valores médios das dez rodadas.

Em todos os cenários, os nós foram aleatoriamente dispostos em um plano cujas dimensões são 1000m x 1000m. O alcance do transmissor, por padrão do simulador, foi ajustado para 250m. A largura de banda do canal foi ajustada para 2 Mbps. O protocolo MAC utilizado foi o 802.11 [23], [22], [42]. Em cenários com nós em movimento, o modelo de mobilidade utilizado foi o *Random Waypoint*. *Random Waypoint* é um modelo de mobilidade amplamente adotado pela comunidade científica em simulações de MANETs [11]. Os valores de mobilidade são variáveis, em um intervalo de zero ao valor máximo definido. O intervalo será sempre aberto no valor zero e fechado no valor máximo.

Para melhor percepção do efeito da mobilidade, o tempo de parada nos cenários com mobilidade foi ajustado para zero. Em observância ao trabalho de Lee et al. (2002) [56], o tamanho dos pedaços foi fixado em 512 Bytes ou 1KB, para um melhor desempenho. A tabela 4.1 sumariza as configurações adotadas:

4.3 Cenários

4.3.1 Variação da Quantidade de Nós Interessados

O foco deste cenário é analisar o quanto a opção por *multicast* foi adequada para entrega de conteúdos com o P2MAN. A idéia é verificar dois comportamentos à medida que o número

¹As versões mais recentes do NS-2 contemplam o uso de semente aleatória obtida à partir da combinação do relógio de tempo real e de um sequenciador.

Tabela 4.1: Ambiente de Simulações do P2MAN

Simulador	Network Simulator 2.33
Rodadas para cada Cenário	10
Área de Simulação	1000m x 1000m
Modelo de Mobilidade	<i>Random Waypoint</i>
Tempo de Parada	0s
Alcance do Transmissor	250m
Largura de Banda	2 Mbps
Protocolo MAC	802.11
Tamanho de um Peçaço	512 Bytes ou 1KB

de nós interessados aumenta: (i) como cresce o tempo de entrega e (ii) como cresce a vazão de entrega. Vazão de entrega é definida como o somatório dos dados entregues aos nós interessados, por unidade de tempo.

Foram dispostos 50 nós aleatoriamente na área de simulação. De forma aleatória, foi selecionado um nó para compartilhar seu conteúdo, com tamanho de 100KB. Em seguida outros nós foram selecionados para se interessarem pelo conteúdo, simultaneamente. O número de nós interessados variou de 1 a 30. Foram executados dois conjuntos de cenários: um com mobilidade e outro sem mobilidade dos nós. A execução sem mobilidade se propõe a verificar o comportamento do protocolo sem as possíveis interferências de desconexões temporárias, provocadas pela mobilidade. Na execução com mobilidade, a velocidade foi variável de 0 m/s a 5 m/s. As figuras 4.1 e 4.2 mostram os resultados.

Observando-se os resultados das simulações é possível perceber que P2MAN tem um comportamento razoavelmente homogêneo em relação às diferentes situações propostas. Em todas as situações o tempo de entrega aumenta linearmente, em proporção ao aumento de nós interessados. Nota-se que o tempo de entrega não sofreu maiores impactos devido à mobilidade. Além disso, ressaltam-se pelas inclinações dos gráficos que, em algumas situações propostas,

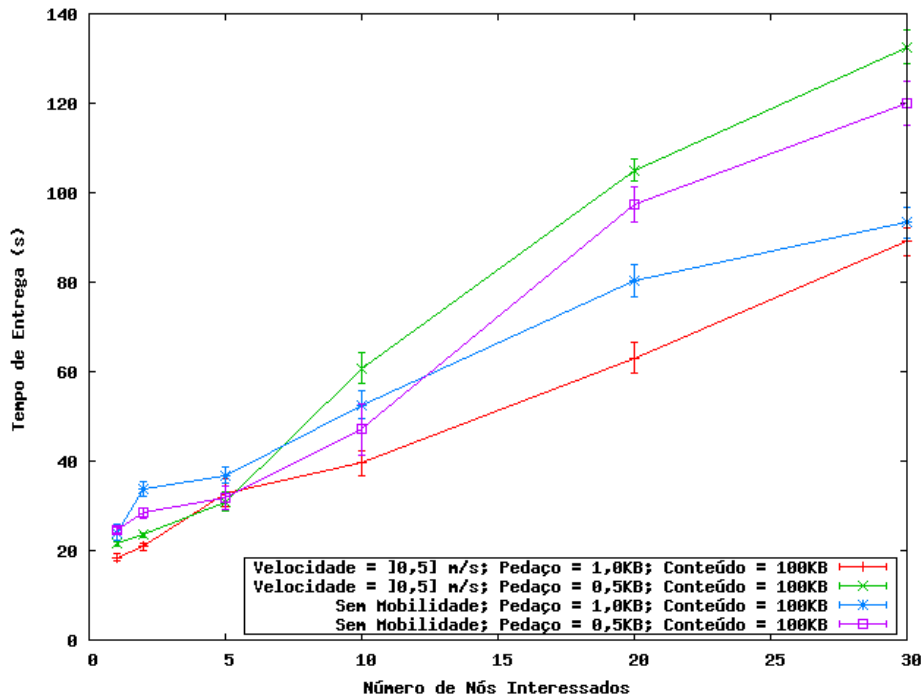


Figura 4.1: Relação entre Tempo de Entrega e Número de Nós Interessados

o aumento do tempo de entrega foi menos acentuado. Em especial, o aumento do tempo de entrega é mais brando nos experimentos onde o tamanho do pedaço foi de 1KB.

O trabalho de Lee et al. (2002) [56] sugere que o tamanho ideal para os segmentos UDP em MANETs deve estar entre 512 Bytes e 750 Bytes, mas admite 1000 Bytes para algumas situações. Num protocolo de distribuição de conteúdo onde o tamanho do pedaço é fixo, o número de pedaços aumenta para um conteúdo maior. Essa quantidade maior de pedaços menores sendo lançados na rede pode, por exemplo, ter contribuído para aumentar as chances de colisão nas simulações específicas realizadas. Essa hipótese pode ser verificada e constará então nos trabalhos futuros.

Pelo gráfico de vazão de entrega, figura 4.2, percebe-se um ganho acentuado em volume de dados entregues à medida que o número de nós interessados aumenta. Esses resultados são animadores porque indicam que em cenários com vários receptores, o desempenho do P2MAN tende a ser melhor. Portanto, os resultados das simulações sugerem ter sido acertada a escolha de UDP e *multicast* para entregar conteúdos em MANETs. Isso também corrobora

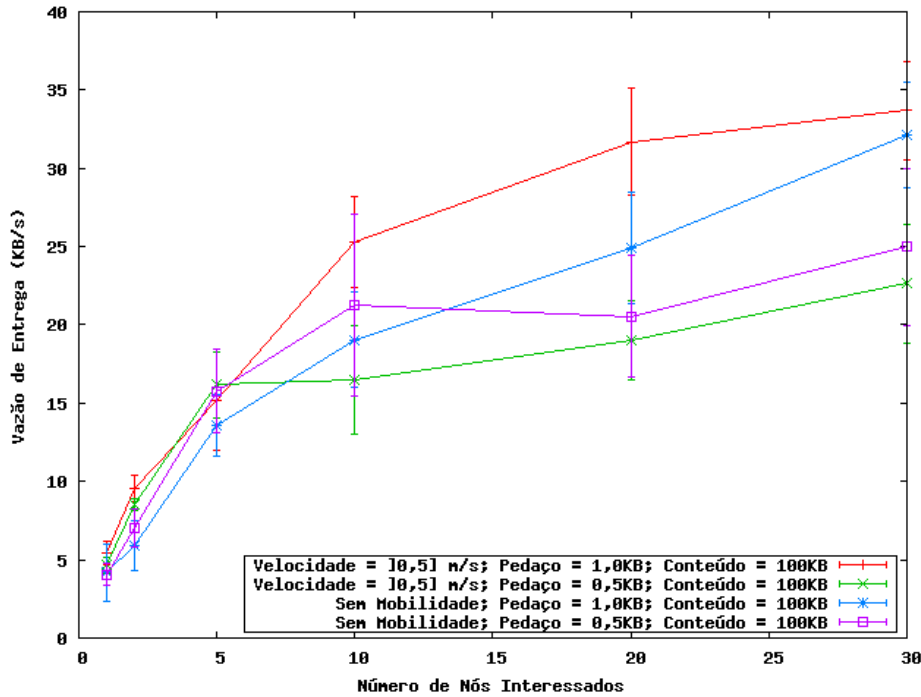


Figura 4.2: Relação entre Vazão de Entrega e Número de Nós Interessados

com a idéia de eficiência, argumento central para o P2MAN.

Para uma confirmação definitiva, é possível fazer um comparativo direto entre a implementação do P2MAN aqui proposta e uma outra implementação usando TCP e *unicast*, como nas redes P2P. Esse estudo também constará nos trabalhos futuros.

4.3.2 Variação do Tamanho do Conteúdo

A motivação deste cenário é medir a influência do tamanho de um conteúdo, e da quantidade de pedaços sendo transmitidos, no tempo de entrega. Foram dispostos 50 nós aleatoriamente na área de simulação. De forma aleatória, foi selecionado um nó para compartilhar seu conteúdo. Em seguida 5 nós foram selecionados para se interessarem pelo conteúdo, simultaneamente. A mobilidade é variável, com velocidade máxima de 5 m/s. Na figura 4.3, o resultado deste conjunto de simulações:

Como se vê, o aumento do tamanho do conteúdo provoca um aumento linear e diretamente proporcional ao tempo de entrega. O aumento de pedaços em circulação na rede resulta em maior probabilidade de transmissões com falhas, e portanto uma quantidade maior

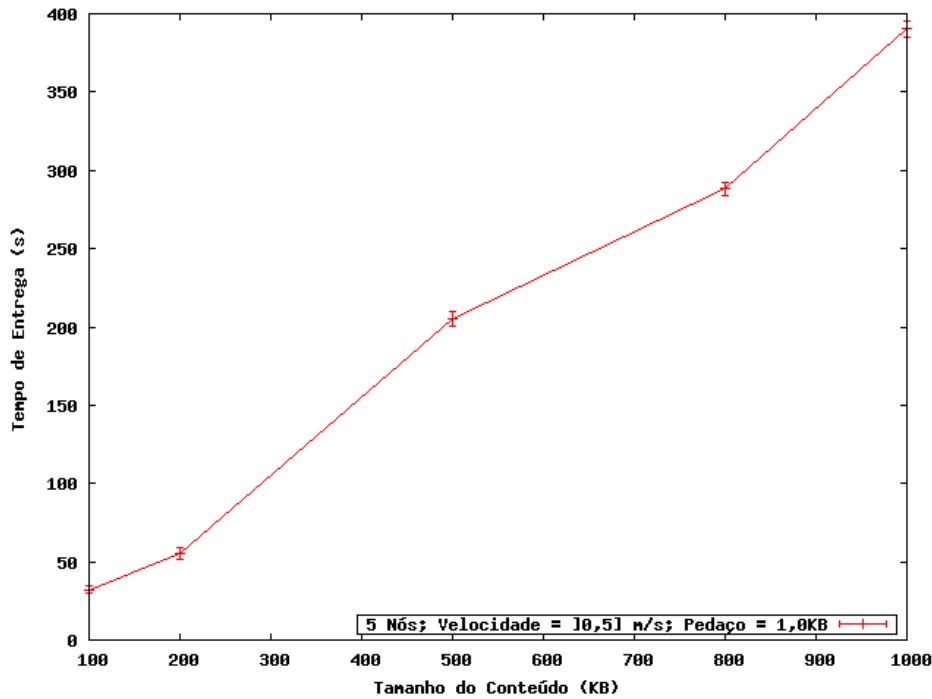


Figura 4.3: Relação entre Tempo de Entrega e Tamanho do Conteúdo

de reparações. As reparações se propõem a ser um mecanismo mais eficiente que um conjunto de transmissões e retransmissões individuais, como ocorreria com o TCP e *unicast* [40], [98], [27], [6]. Porém, mesmo mais eficientes em teoria, as reparações são o mecanismo mais dispendioso do P2MAN. Isto porque as reparações provocam retransmissão de pedaços que podem já ter sido adquiridos por nós interessados. Durante o período de reparação, a eficiência do protocolo diminui, podendo chegar a uma situação em que atenderá a entrega de pedaços específicos que restam ser entregue a cada nó.

É possível medir a frequência e a influência das reparações numa transmissão. Tal medição seria útil para otimizar a especificação e/ou a implementação do P2MAN e torná-lo mais adaptativo às situações diversas. Por exemplo, há um balanço para o tempo de espera por um pedaço. Se o valor for muito baixo, pode-se considerar precocemente um pedaço como perdido, acionando o mecanismo de reparação. Se o valor for muito alto, pode-se protelar demais o acionamento de uma reparação, reduzindo-se também o desempenho.

4.3.3 Variação da Mobilidade

O foco deste cenário é compreender o efeito da mobilidade no tempo de entrega de conteúdos. Foram dispostos 50 nós, aleatoriamente no campo. Um nó foi selecionado, também de forma aleatória, para compartilhar seu conteúdo, com o tamanho de 100KB. Em seguida, outros nós foram selecionados para se interessarem pelo conteúdo, simultaneamente. A cada nó foi permitido variar sua velocidade em um intervalo aberto em zero e fechado no valor máximo. Variou-se a velocidade máxima dos nós entre 0 m/s e 30 m/s, não incluindo o valor zero. Na figura 4.4, estão os resultados dessas simulações:

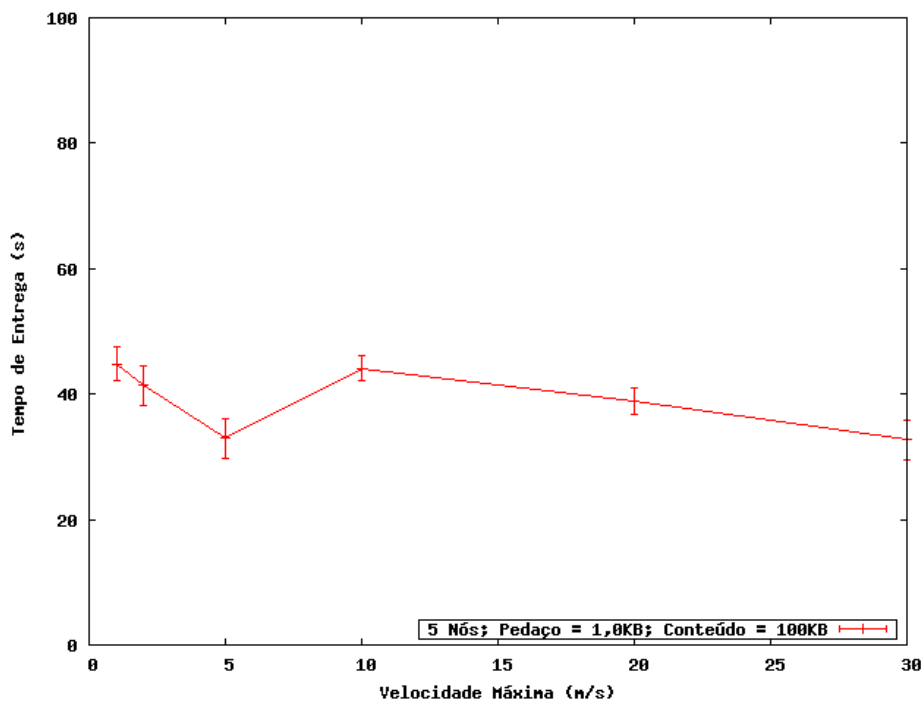


Figura 4.4: Relação entre Tempo de Entrega e Mobilidade dos Nós

É possível perceber que, diante do aumento na mobilidade, ocorreu uma leve variação no tempo de entrega. Portanto, os resultados das simulações indicam robustez do protocolo a essa condição. Ainda mais, ponderando-se que 20 m/s ou 30 m/s são valores de mobilidade mais severos.

Capítulo 5

Conclusão

5.1 Resumo das Atividades

Foram estudados dois tipos de redes dinâmicas: as redes móveis *ad hoc* e as redes entre-pares. Suas características foram detalhadas e foram discutidos os rumos tomados pela ciência no sentido de unir sinergicamente as duas redes. Ponderou-se que portar uma rede P2P, nos moldes em que elas são utilizadas na Internet, pode não ser a melhor solução. Propôs-se uma adequação diferente, onde uma rede de distribuição de conteúdo, nativa para MANETs, poderia ser mais adequada e eficiente.

Propôs-se então o protocolo de distribuição de conteúdo Peer-to-MANET. Escolheu-se o protocolo de roteamento PUMA para as simulações de validação do P2MAN, devido ao seu melhor desempenho. Implementou-se o PUMA e o P2MAN no simulador NS-2 e ambos os códigos foram disponibilizados para a comunidade científica.

Os resultados de avaliação do P2MAN, nos cenários propostos, indicaram ser um protocolo eficiente e robusto. Sua especificação, com garantia de entrega no nível de aplicação, assegurou-lhe a confiabilidade necessária para distribuir conteúdos.

5.2 Considerações Finais

Somente após o trabalho de pesquisa e da implementação, quando da avaliação do P2MAN, alguns questionamentos foram surgindo. Os números dos experimentos mostraram que distribuir conteúdo em uma MANET requer muita habilidade, pois as MANETs, por utilizarem meio compartilhado, dentre outros, sofrem com colisões. É preciso compreender a fundo os mecanismos que regem uma MANET, porque independe da natureza do tráfego, ao se tentar aumentar demais o tráfego numa MANET, obtém-se o efeito contrário. São tantas as interferências que há uma tendência visível ao colapso, quando se atinge uma certa vazão.

Tanto assim, que alguns experimentos de aplicações de redes P2P, em suas validações, alteram parâmetros notórios às MANETs, para abrandar as interferências. BitHoc [84], por exemplo, foi validado sem mobilidade, com o alcance de rádio reduzido e outros recursos para evitar colisões e falhas de transmissão. Isso sem considerar o fato de que os simuladores de rede são otimistas e não consideram reflexos de sinal, interefêrências ou outros elementos hostis. É neste ponto que esse trabalho defende o mérito do P2MAN. Trata-se de um protocolo avaliado sob condições de uso consideradas normais a uma MANET. P2MAN é um protocolo jovem, e seu desempenho ainda precisa ser aprimorado.

5.3 Resumo das Contribuições

As principais contribuições providas por este trabalho foram:

- Um estudo das redes móveis *ad hoc* e das redes P2P;
- A implementação do protocolo de roteamento *multicast* PUMA, através de sua especificação, para o simulador NS-2;
- A implementação do protocolo de distribuição de conteúdo P2MAN para o simulador NS-2;
- Uma avaliação do protocolo P2MAN, em cenários diversos, e suas respectivas análises.

5.4 Trabalhos Futuros

Como visto, os resultados obtidos pelo P2MAN demandaram outros estudos mais profundos, tanto para avanço do protocolo quanto para uma melhor compreensão dos mecanismos envolvidos. Além disso, há de se fazer comparativos do desempenho do P2MAN com outros protocolos, recém portados para MANETs. Nessa linha de ação, alguns trabalhos futuros são propostos:

- Comparar o desempenho do P2MAN proposto com uma versão que utilize *unicast* e TCP;
- Analisar em profundidade o impacto do tamanho do conteúdo no tempo de entrega e propor mecanismos para redução desse impacto;
- Otimizar a implementação do P2MAN, de forma a torná-lo mais adaptativo às diversas situações propostas;
- Comparar o desempenho do P2MAN sobre outros protocolos de roteamento;

O funcionamento do P2MAN foi inspirado nas redes P2P. Entretanto, a busca da comunidade é pela sinergia entre as MANETs e redes P2P. Portanto, um outro passo a ser trilhado é levar de volta às redes P2P o conhecimento adquirido e as soluções adotadas no P2MAN. Soma-se então mais uma pergunta a ser respondida em um trabalho futuro: "Que lições podem ser levadas do P2MAN para as redes P2P?".

Referências Bibliográficas

- [1] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2:1–22, January 2004.
- [2] Norman Abramson. The aloha system – another alternative for computer communications. In *Proceedings of the Fall Joint Computer Conference*, volume 36, pages 177–186, Montvale, New Jersey, USA, 1970.
- [3] Defense Advanced Research Projects Agency. Transmission control protocol. Disponível em <<http://www.ietf.org/rfc/rfc0793.txt>>.
- [4] George Aggelou and Rahim Tafazolli. Rdmr: a bandwidth-efficient routing protocol for mobile ad hoc networks. In *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia*, pages 26–33, 1999.
- [5] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computer Surveys*, 36(4):335–371, 2004.
- [6] Hari Balakrishnan, Venkata Padmanabhan, Srinivasan Seshan, and Randy Katz. A comparison of mechanisms for improving tcp performance over wireless links. In *Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 256–269, 1996.
- [7] Stefano Basagni, Imrich Chlamtac, Violet Syrotiuk, and Barry Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 76–84, 1998.
- [8] BigChampagne. Sítio bigchampagne. <<http://www.bigchampagne.com>>.

-
- [9] Joseph Borg. A comparative study of ad hoc & peer to peer networks. Master's thesis, University of London, University College London, Faculty of Engineering Department of Electronic & Electrical Engineering, 2003.
- [10] Imed Bouazizi. Ara - the ant-colony based routing algorithm for manets. In *Proceedings of the International Conference on Parallel Processing Workshops*, page 79, 2002.
- [11] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, 2(5):483–502, 2002.
- [12] Tsu-Wei Chen and Mario Gerla. Global state routing: A new routing scheme for ad hoc wireless networks. In *IEEE International Conference on Communications*, pages 171–175, 1998.
- [13] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, and Mario Gerla. Routing in clustered multihop, mobile wireless networks. In *IEEE Singapore International Conference on Networks*, pages 197–211, April 1997.
- [14] Ian Clarke, Scott Miller, Theodore Hong, Oskar Sandberg, and Brandon Wiley. Protecting free expression online with freenet. *IEEE Internet Computing*, 6(1):40–49, 2002.
- [15] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore Hong. *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, volume 2009, page 46. Springer Berlin / Heidelberg, Berkeley, CA, USA, 2001.
- [16] Thomas Clausen and Philippe Jacquet. RFC 3626. optimized link state routing protocol (OLSR), 2003.
- [17] Thomas Clausen, Philippe Jacquet, Anis Laouiti, Paul Mühlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of the 5th IEEE Multi Topic Conference*, Pakistan, 2001.

-
- [18] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, EUA, May 2003.
- [19] Marco Conti, Enrico Gregori, and Giovanni Turi. Towards scalable p2p computing for mobile ad hoc networks. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, page 109, 2004.
- [20] Mathew Scott Corson and Anthony Ephremides. A distributed routing algorithm for mobile wireless networks. *Wireless Networks*, 1(1):61–81, 1995.
- [21] Scott Corson and Joseph Macker. RFC 2501. mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations, 1999.
- [22] Brian Crow, Indra Widjaja, Jeong Kim, and Prescott Sakai. IEEE 802.11 wireless local area networks. *IEEE Communications Magazine*, pages 116–126, 1997.
- [23] Brian Crow, Indra Widjaja, Jeong Geun Kim, and Prescott Sakai. IEEE 802.11 wireless local area networks. In *Communications Magazine*, volume 35, pages 116–126, September 1997.
- [24] Mayur Datar. Butterflies and peer-to-peer networks. In *Proceedings of the 10th Annual European Symposium on Algorithms*, pages 310–322, 2002.
- [25] Gang Ding and Bharat Bhargava. Peer-to-peer file-sharing over mobile ad hoc networks. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, page 104, 2004.
- [26] Rohit Dube, Cynthia Rais, Kuang-Yeh Wang, and Satish Tripathi. Signal stability based adaptive routing (SSA) for ad-hoc mobile networks. *IEEE Personal Communications*, pages 36–45, February 1997.
- [27] Thomas D. Dyer and Rajendra V. Boppana. A comparison of tcp performance over three routing protocols for mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 56–66, 2001.

- [28] Amos Fiat and Jared Saia. Censorship resistant peer-to-peer content addressable networks. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 94–103, 2002.
- [29] Gregory Finn. Routing and addressing problems in large metropolitan-scale inter-networks. Technical report, University of Southern California. Marina Del Rey Information Sciences Institute, 1987.
- [30] James Freebersyser and Barry Leiner. *A DoD Perspective on Mobile Ad Hoc Networks*, pages 29–51. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [31] Jose Joaquin Garcia-Luna-Aceves and Ewerton Madruga. The core assisted mesh protocol. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, 17(8):1380–1394, 1999.
- [32] Jose Joaquin Garcia-Luna-Aceves and Marcelo Spohn. Transmission-efficient routing in wireless networks using link-state information. *Mobile Networks and Applications*, 6(3):223–238, 2001.
- [33] Mario Gerla, Xiaoyan Hong, and Guangyu Pei. Internet-Draft. fisheye state routing protocol (FSR) for ad hoc networks, 2002.
- [34] Mario Gerla, Christoph Lindemann, and Ant Rowstron. P2P manet's - new research issues. In *Perspectives Workshop: Peer-to-Peer Mobile Ad Hoc Networks - New Research Issues*, number 05152 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005.
- [35] GNUTELLA. Sítio do protocolo gnutella. Disponível em <http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html>, Acessado em 01/01/2007.
- [36] Zygmunt Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of 6th IEEE International Conference on Universal Personal Communications*, volume 2, pages 562–566, San Diego, California, USA, October 1997.

-
- [37] Zygmunt Haas, Marc Pearlman, and Prince Samar. The zone routing protocol (ZRP) for ad hoc networks. Internet-draft, IETF MANET Working Group, July 2002.
- [38] Charles Hedrick. RFC 1058. routing information protocol. Internet Engineering Task Force - RFC Editor, 1988.
- [39] Luc Hogue, Pascal Bouvry, and Frédéric Guinand. An overview of manets simulation. In *Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems*, LNCS, 2005.
- [40] Gavin Holland and Nitin Vaidya. Analysis of tcp performance over mobile ad hoc networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 219–230, 1999.
- [41] Xiaoyan Hong, Kaixin Xu, and Mario Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, July/August 2002.
- [42] Lifei Huang and Ten-Hwang Lai. On the scalability of IEEE 802.11 ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 173–182, 2002.
- [43] Mingliang Jiang, Jinyang Li, and Yong Tay. Cluster based routing protocol(CBRP) functional specification. Internet-draft, IETF MANET Working Group, August 1998.
- [44] Mario Joa-Ng and I-Tai Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE Journal on Selected Areas In Communication*, 17(8):1415–1425, August 1999.
- [45] David Johnson. Routing in ad hoc networks of mobile hosts. In *Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA, 1994.
- [46] David Johnson, Yih-Chun Hu, and David Maltz. RFC 4728. the dynamic source routing protocol (DSR), 2004.
- [47] David Johnson and David Maltz. Dynamic source routing in ad hoc wireless networks. In Thomasz Imielinski and Hank Korth, editors, *Mobile Computing*, volume 353, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

-
- [48] John Jubin and Janet Tornow. The darpa packet radio network protocols. In *Proceedings of IEEE*, volume 75, pages 21–32, 1987.
- [49] Robert Kahn, Steven Gronemeyer, Jerry Burchfiel, and Ronald Kunzelman. Advances in packet radio technology. In *Proceedings of IEEE*, volume 66, pages 1468–1496, 1978.
- [50] Alexander Klemm, Christoph Lindemann, and Oliver Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. *IEEE 58th Vehicular Technology Conference*, 4:2758–2763, 2003.
- [51] Young-Bae Ko and Nitin Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 66–75, 1998.
- [52] Young-Bae Ko and Nitin Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.
- [53] Gerd Kortuem, Jay Schneider, Dustin Preuitt, Thaddeus Thompson, Stephen Fickas, and Zary Segall. When peer-to-peer comes face-to-face: Collaborative peer-to-peer computing in mobile ad hoc networks. In *Proceedings of the First International Conference on Peer-to-Peer Computing*, page 75, 2001.
- [54] Ulag Kozat, Iordanis Koutsopoulos, and Leandros Tassiulas. A framework for cross-layer design of energy-efficient communication with qos provisioning in multi-hop wireless networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1446–1456, 2004.
- [55] US Naval Research Laboratory. Norm - negative-acknowledgment oriented reliable multicast. Disponível em <<http://www.ietf.org/rfc/rfc3940.txt>>, Acessado em 01/04/2008.
- [56] JangYeon Lee, GyeYoung Kim, and SungKwon Park. Optimum udp packet sizes in ad hoc networks. In *Workshop on High Performance Switching and Routing. Merging Optical and IP Technologies*, pages 214–218, Seoul, South Korea, 2002.

- [57] Sung-Ju Lee, Mario Gerla, and Ching-Chuan Chiang. On-demand multicast routing protocol. *Wireless Communications and Networking Conference*, 3:1298–1302, 1999.
- [58] Yeng-Zhong Lee, Mario Gerla, Jason Chen, Jiwei Chen, Biao Zhou, and Antonio Caruso. Direction forward routing for highly mobile ad hoc networks. *Ad Hoc & Sensor Wireless Networks Journal*, 2006.
- [59] Marc Mosko. *Routing in Mobile Ad Hoc Networks*. PhD thesis, University of California at Santa Cruz, 2004.
- [60] John Moy. RFC 2328. ospf version 2, 1998.
- [61] Shree Murthy and Jose Joaquin Garcia-Luna-Aceves. A routing protocol for packet radio networks. In *Proceedings of the 1st Annual International Conference on Mobile Computing and Networking*, pages 86–95, 1995.
- [62] NAPSTER. Sítio do napster. Disponível em <<http://www.napster.com>>, Acessado em 01/01/2007.
- [63] Navid Nikaein, Houda Labiod, and Christian Bonnet. Ddr: Distributed dynamic routing algorithm for mobile ad hoc networks. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 19–27, 2000.
- [64] Guevara Noubir, Wei Qian, Bishal Thapa, and Yin Wang. Real-world experimentation-oriented platform for development and evaluation of manet cross-layer protocols. Technical report, College of Computer Science and Information Science, Northeastern University, Boston, MA 02115, 2008.
- [65] NS-2. The network simulator. <<http://www.isi.edu/nsnam/ns>>.
- [66] Richard Ogier, Fred Templin, and Mark Lewis. RFC 3684. topology dissemination based on reverse-path forwarding (TBRPF), 2004.
- [67] Leonardo Oliveira, Isabela Siqueira, and Antonio Loureiro. On the performance of ad hoc routing protocols under a peer-to-peer application. *Journal on Parallel and Distributed Computing*, 65(11):1337–1347, 2005.

-
- [68] Vincent Park and Mathew Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *INFOCOM*, 00:1405, 1997.
- [69] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing in mobile ad hoc networks. In *ICDCS Workshop on Wireless Networks and Mobile Computing*, pages D71–D78, 2000.
- [70] Guangyu Pei, Mario Gerla, Xiaoyan Hong, and Ching-Chuan Chiang. A wireless hierarchical routing protocol with group mobility. *IEEE Wireless Communications and Networking Conference*, pages 1538–1542, 1999.
- [71] Charles Perkins, Elisabeth Belding-Royer, and Samir Das. RFC 3561. ad hoc on-demand distance vector (AODV) routing, 1999.
- [72] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.
- [73] Himabindu Pucha, Saumitra Das, and Ying Charlie Hu. Imposed route reuse in ad hoc network routing protocols using structured peer-to-peer overlay routing. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1452–1467, 2006.
- [74] S. Radhakrishnan, Nageswara Rao, Gopal Racherla, Chandra Sekharan, and Steven Batsell. DST - a routing protocol for ad hoc networks using distributed spanning trees. *IEEE Wireless Communications and Networking Conference*, 3:1543–1547, 1999.
- [75] Rajagopalan, Sundaram, Shen, and Chien-Chung. A cross-layer decentralized bittorrent for mobile ad hoc networks. In *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pages 1–10, 2006.
- [76] Jyoti Raju and Jose Joaquin Garcia-Luna-Aceves. A new approach to on-demand loop-free multipath routing. In *IEEE Eight International Conference on Computer Communications and Networks*, pages 522–527, Boston, Massachusetts, USA, 1999.
- [77] Ram Ramanathan and Jason Redi. A brief overview of ad hoc networks: Challenges and directions. *Communications Magazine, IEEE*, 40(5):20–22, 2002.

- [78] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 161–172, 2001.
- [79] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329, 2001.
- [80] Elisabeth Royer and Charles Perkins. Internet-Draft. multicast ad hoc on-demand distance vector (MAODV) routing, 2000.
- [81] Elisabeth Royer and Chai-Keong Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, April 1999.
- [82] Jared Saia, Amos Fiat, Steven Gribble, Anna Karlin, and Stefan Saroiu. Dynamically fault-tolerant content addressable networks. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 270–279, 2002.
- [83] César Santiváñez, Ram Ramanathan, and Ioannis Stavrakakis. Making link-state routing scale for ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 22–32, 2001.
- [84] Mohamed Karim Sbai, Chadi Barakat, Jaeyoung Choi, Anwar Al Hamra, and Thierry Turletti. BitHoc: Bittorrent for wireless ad hoc networks. Technical report, inria-00196313, France, 2008.
- [85] Rüdiger Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of IEEE First International Conference on Peer-to-Peer Computing*, page 101, Washington, DC, USA, 2001.
- [86] Rüdiger Schollmeier, Ingo Gruber, and Michael Finkenzeller. Routing in mobile ad-hoc and peer-to-peer networks a comparison. In *Revised Papers from the NETWORKING Workshops on Web Engineering and Peer-to-Peer Computing*, pages 172–186, 2002.

-
- [87] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM Conference*, pages 149–160, 2001.
- [88] Ivan Stojmenović. *Location Updates for Efficient Routing in Ad Hoc Networks*, pages 451–471. John Wiley & Sons, Inc., New York, NY, USA, 2002.
- [89] William Su and Mario Gerla. Ipv6 flow handoff in ad-hoc wireless networks using mobility prediction. In *IEEE Global Communications Conference*, pages 271–275, Rio de Janeiro, Brazil, 1999.
- [90] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.
- [91] Domenico Talia and Paolo Trunfio. Toward a synergy between p2p and grids. *IEEE Internet Computing*, 7(4):96–95, 2003.
- [92] Fouad Tobagi. *Random Access Techniques for Data Transmission Over Packet Switched Radio Networks*. PhD thesis, University of California Los Angeles. School of Engineering and Applied Science, 1974.
- [93] Fouad Tobagi and Leonard Kleinrock. Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple access modes and the busy-tone solution. *IEEE Transactions on Communication*, 23(12):1417–1433, 1975.
- [94] Chai-Keong Toh. A novel distributed routing protocol to support ad-hoc mobile computing. In *Conference Proceedings of the IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pages 480–486, Scottsdale, AZ, USA, 1996.
- [95] Ravindra Vaishampayan. *Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks*. PhD thesis, University of California at Santa Cruz, 2006.
- [96] Ravindra Vaishampayan and Jose Joaquin Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 304–313, 2004.

-
- [97] Chonggang Wang and Bo Li. Peer-to-peer overlay networks: A survey. Disponível em: <<http://lumumba.uhasselt.be/stofke/thesis/papers/15.pdf>>, 2003.
- [98] Feng Wang and Yongguang Zhang. Improving tcp performance over mobile ad hoc networks ith out-of-order detection and response. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 217–225, 2002.
- [99] Seung-Chul Woo and Suresh Singh. Scalable routing protocol for ad hoc networks. *Wireless Networks*, 7(5):513–529, 2001.
- [100] Jie Wu. *Handbook On Theoretical And Algorithmic Aspects Of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, pages 857–874. Auerbach Publications, Boston, MA, USA, 2005.
- [101] Saumitra Das Y. Charlie Hu and Himabindu Pucha. Exploiting the synergy between peer-to-peer and mobile ad hoc networks. In *9th Workshop on Hot Topics in Operating Systems*, pages 37–42, Lihue, HI, USA, 2003.
- [102] Lu Yan. Performance evaluation and modeling of peer-to-peer systems over mobile ad hoc networks. Technical report, TUCS Technical Report No. 678, Turku, Finland, 2005.
- [103] Ben Zhao, John Kubiawicz, and Anthony Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, University of California, Berkeley, CA, USA, 2001.
- [104] Ben Zhao, John Kubiawicz, and Anthony Joseph. Peer-to-peer computing. Technical report, HP Labs, 2002.
- [105] Ben Zhao, John Kubiawicz, and Anthony Joseph. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22:41–53, 2004.

Apêndice A

Instalação do P2MAN no NS-2

O código do P2MAN, utilizado nos experimentos desta dissertação, já está disponível em <http://sourceforge.net/projects/p2man>. Este anexo visa apresentar um tutorial com os passos para instalação do código do P2MAN no NS-2. O objetivo é permitir que novos trabalhos possam aprimorar o P2MAN à partir do ponto atual de desenvolvimento.

Para os passos abaixo, deve-se considerar a pasta de instalação do NS-2. Para o pacote *All-in-One* do NS-2 na versão 2.33, a instalação ocorrerá em `ns-allinone-2.33/ns-2.33`. Como é possível que o usuário não utilize um pacote *All-in-One* e instale os componentes de forma independente, denominou-se a pasta de instalação `ns-2.33` como `<raiz do NS-2>`.

Por se tratar de um aplicativo, P2MAN não requer maiores intervenções no simulador. Os passos para a instalação são:

1. obtenha o arquivo `p2man.zip` em <http://sourceforge.net/projects/p2man>;
2. descompacte o arquivo `p2man.zip` em `<raiz do NS-2>/p2man`;
3. edite o arquivo `Makefile` conforme detalhamento a seguir;
4. a partir da pasta `<raiz do NS-2>/p2man`, execute o *script* de comandos `./run`

A.1 Alteração Necessária ao Arquivo Makefile

Edite o arquivo `<raiz do NS-2>/Makefile` e adicione a linha:

Logo acima do DSDV:

```
p2man/p2man.o \
```

A.2 Visualização da Simulação Exemplo

Após a conclusão dos passos acima, o Peer-to-MANET já estará instalado. O *script* de comandos iniciará a compilação do código e em seguida uma simulação de exemplo. Trata-se de uma simulação num cenário de 50 nós dispostos aleatoriamente. Um nó compartilha seu conteúdo de 1MB e mais cinco nós se interessam por esse conteúdo simultaneamente. Ao final da simulação, o arquivo é adquirido por todos os interessados.

É possível visualizar o comportamento do P2MAN durante a simulação com o Network Animator (NAM). NAM é um programa livre, distribuído com o NS-2, que gera apresentações gráficas a partir dos registros das simulações.

Apêndice B

Instalação do PUMA no NS-2

Uma das contribuições deste trabalho foi a implementação do protocolo de roteamento *multicast* PUMA no NS-2. Durante a feitura desta dissertação, o código do PUMA foi anunciado na lista de discussão do NS-2. Até o presente momento, dois estudantes asiáticos já fizeram contato para obter maiores informações de como usar o PUMA em seus projetos de mestrado. Sendo assim, este anexo visa apresentar um tutorial com os passos para instalação do código do PUMA no NS-2. O objetivo é acelerar a obtenção de resultados por pesquisadores que venham a se interessar.

Abaixo deve-se considerar a pasta de instalação do NS-2. Para o pacote *All-in-One* do NS-2 na versão 2.33, a instalação ocorrerá em `ns-allinone-2.33/ns-2.33`. Denominou-se também a pasta de instalação `ns-2.33` como <raiz do NS-2>.

Os passos para a instalação são:

1. obtenha o arquivo `puma.zip` em <<http://sourceforge.net/projects/puma-adhoc>>;
2. descompacte o arquivo `puma.zip` em <raiz do NS-2>/`puma`;
3. edite os arquivos fonte conforme detalhamento a seguir;
4. a partir da pasta <raiz do NS-2>/`puma`, execute o *script* de comandos `./run`

B.1 Alterações Necessárias ao Arquivo Makefile

Edite o arquivo <raiz do NS-2>/Makefile e adicione a linha:

Logo acima do DSDV:

```
puma/puma.o \
```

B.2 Alterações Necessárias ao Arquivo ns-lib.tcl

Edite o arquivo <raiz do NS-2>/tcl/lib/ns-lib.tcl e adicione as linhas:

Logo acima do AODV:

```
PUMA {  
    set ragent [$self create-puma-agent $node]  
}
```

Logo acima do AODV:

```
Simulator instproc create-puma-agent { node } {  
    # Create PUMA routing agent  
    set ragent [new Agent/PUMA [$node node-addr]]  
    $self at 0.0 "$ragent start"      ;# start agent  
    $node set ragent_ $ragent  
    return $ragent  
}
```

B.3 Alterações Necessárias ao Arquivo packet.h

Edite o arquivo <raiz do NS-2>/common/packet.h e adicione as linhas:

O número abaixo pode variar se já houver adição de outros experimentos.

Incremente o número do último protocolo existente. No arquivo original o último número é 60.

```
// PUMA packet  
static const packet_t PT_PUMA = 61;
```


Adicione também, logo acima do DSDV:

```
type == PT_PUMA ||
```

Execute a seguir:

```
touch <raiz do NS-2>/common/packet.cc
```

B.4 Visualização da Simulação Exemplo

Após a conclusão desses passos, o PUMA já estará instalado. O *script* de comandos iniciará a compilação do código e uma simulação de exemplo. Trata-se de uma simulação em que um nó com gerador de tráfego constante (CBR) envia pacotes de 512 Bytes para um determinado grupo multicast, numa razão de 10 pacotes por segundo. É possível visualizar o comportamento do PUMA durante a simulação com o NAM. Neste cenário os pacotes são repassados nó-a-nó. É possível visualizar que os pacotes passam por membros da malha *multicast* e por nós que apenas foram requisitados a rotear os pacotes devido à sua distância para o líder.