UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

FÁBIO LUIZ PESSOA ALBINI

PTTA: PROTOCOLO PARA DISTRIBUIÇÃO DE CONTEÚDO EM REDES TOLERANTES AO ATRASO E DESCONEXÕES

TESE

CURITIBA

2013

FÁBIO LUIZ PESSOA ALBINI

PTTA: PROTOCOLO PARA DISTRIBUIÇÃO DE CONTEÚDO EM REDES TOLERANTES AO ATRASO E DESCONEXÕES

Tese apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Doutor em Ciências" – Área de Concentração: Telecomunicações e Redes.

Orientadora: Profa. Dra. Anelise Munaretto

Co-orientador: Prof. Dr. Mauro Fonseca

CURITIBA

2013

Dados Internacionais de Catalogação na Publicação

A336 Albini, Fábio Luiz Pessoa
PTTA : protocolo para distribuição de conteúdo em redes tolerantes ao atraso e desconexões / Fábio Luiz Pessoa Albini. — 2013.
92 f. : il. ; 30 cm

Orientador: Anelise Munaretto Fonseca.
Coorientador: Mauro Sérgio Pereira Fonseca.
Tese (Doutorado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2013.
Bibliografia: f. 87-92.

1. Protocolos de roteamento (Protocolos de redes de computadores). 2. Tolerância a falha (Computação). 3. Redes ad hoc (Redes de computadores). 4. Redes de computação – Confiabilidade. 5. Códigos corretores de erros (Teoria da informação). 6. Simulação (Computadores). 7. Engenharia elétrica – Teses. I. Fonseca, Anelise Munaretto, orient. II. Fonseca, Mauro Sérgio Pereira, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD (22. ed.) 621.3

Biblioteca Central da UTFPR, Campus Curitiba

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Câmpus Curitiba

Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

Título da Tese Nº.

"PTTÁ: Protocolo para Distribuição de Conteúdo em Redes Tolerantes a Atrasos e Desconexões"

por

Fábio Luiz Pessoa Albini

Orientadora: Prof.ª Dr.ª Anelise Munaretto Fonseca
Coorientador: Prof. Dr. Mauro Sérgio Pereira Fonseca

Esta tese foi apresentada como requisito parcial à obtenção do grau de DOUTOR EM CIÊNCIAS – Área de Concentração: Telecomunicações e Redes, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14h do dia 30 de outubro de 2013. O trabalho foi aprovado pela Banca Examinadora, composta pelos doutores:

Prof.ª Anelise Munaretto Fonseca, Dr.
(Presidente – UTFPR)

Prof. Marcelo Dias de Amorim, Dr.
(Université Pierre et Marie Curie)

Prof. Marcelo Eduardo Pellenz, Dr.
(PUC/PR)

Prof.ª Michele Nogueira, Dr.
(UFPR)

Prof. João Luiz Rebelatto, Dr.
(UTFPR)

Visto da Coordenação:

Prof. Ricardo Lüders, Dr.
(Coordenador do CPGEI)

Aos meus pais
Luiz Carlos Albini e
Vera Lúcia Pessoa Albini

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus. Em segundo à toda a minha família, aos meus orientadores Profa. Anelise, Prof. Mauro e Prof. Francesco, à minha família italiana, aos meus amigos e pessoas queridas, aos membros da banca e a todos os meus professores.

Nada acontece por acaso!

Coincidência é a maneira com que os incrédulos afirmam as coisas que aconteceram por acaso para não assumirem que o acaso não existe!

Fábio Albini

RESUMO

ALBINI, Fabio L. P.. PTTA: PROTOCOLO PARA DISTRIBUIÇÃO DE CONTEÚDO EM REDES TOLERANTES AO ATRASO E DESCONEXÕES. 92 f. Tese – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

O presente trabalho consiste na proposta de um novo protocolo de transporte para redes tolerantes a atrasos e desconexões (DTN - *Delay Tolerant Network*) chamado PTTA - Protocolo de Transporte Tolerante a Atrasos (em inglês - DTTP - *Delay Tolerant Transport Protocol*). Este protocolo tem o objetivo de oferecer uma confiabilidade estatística na entrega das informações em redes deste tipo. Para isso, serão utilizados Códigos Fontanais como técnica de correção de erros. Os resultados mostram as vantagens da utilização do PTTA. Este trabalho ainda propõe um mecanismo de controle da fonte adaptável para o PTTA a fim de limitar a quantidade de dados gerados pela origem (fonte). O esquema proposto almeja aumentar a diversidade das informações codificadas sem o aumento da carga na rede. Para atingir este objetivo o intervalo de geração e o TTL (*Time To Live* - Tempo de vida) das mensagens serão manipulados com base em algumas métricas da rede. A fim de validar a eficiência do mecanismo proposto, diferentes cenários foram testados utilizando os principais protocolos de roteamento para DTNs. Os resultados de desempenho foram obtidos levando em consideração o tamanho do buffer, o TTL das mensagens e a quantidade de informação redundante gerada na rede. Os resultados de simulações obtidos através do simulador ONE mostram que nos cenários avaliados, o PTTA alcança um aumento na taxa de entrega das informações em um menor tempo, quando comparado com outro protocolo de transporte sem confirmação, permitindo assim um ganho de desempenho na rede.

Palavras-chave: Redes Tolerantes a Atrasos, Redes Ad-hoc Oportunistas, Protocolo de Transporte, Códigos Fontanais, Códigos Corretores de Erros

ABSTRACT

ALBINI, Fabio L. P.. PTTA: A DELAY AND DISRUPTION NETWORK PROTOCOL FOR CONTENT DISTRIBUTION. 92 f. Tese – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

The present work consists in the proposal of a new transport protocol for delay tolerant networks and disconnections (DTN - Delay Tolerant Network) called DTTP - Delay Tolerant Transport Protocol (in portuguese - PTTA - Protocolo de Transporte Tolerante a Atrasos). This protocol aims to provide a statistical reliability in DTNs' information delivery. For this, we use fountain codes as error correction technique. The results show the advantages of using DTTP. This work also proposes an adaptive control mechanism for the DTTP source to limit the amount of generated data. The proposed scheme aims at increasing the diversity of encoded information without increasing the load on the network. To achieve this goal the messages generation interval and TTL (Time To Live) will be handled based on some network metrics. In order to validate the efficiency of the proposed mechanism, different scenarios will be tested using the main routing protocols for DTNs. The performance results were obtained taking into account the buffer size, messages TTL and the amount of redundant information generated on the network. The simulation results, obtained through The ONE simulator, show that in the evaluated scenarios PTTA achieves an increase in the information delivery rate in a shorter time compared to other transport protocol for confirmation, thus allowing a gain in the network performance.

Keywords: Delay Tolerant Networks, Opportunistic Ad-hoc Networks, Transport Protocol, Fountain Codes, Error Correction Codes

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DTTP	*Delay Tolerant Transport Protocol*
PTTA	Protocolo de Transporte Tolerante a Atrasos
JPL	Jet Propulsion Laboratory
IPN	*Interplanetary Network*
IPNSIG	*IPN Special-Interest Group*
RFC	*Request For Comments*
IRTF	*Internet Research Task Force*
DTNRG	*Delay Tolerant Networking Research Group*
MANET	*Mobile Ad-hoc Networks*
UDP	*User Datagram Protocol*
TCP	*Transmission Control Protocol*
XOR	*eXclusive-OR* (ou-exclusivo)
LT	*Luby Transform* (Transformada de Luby)
OPF	*Optimal Probabilistic Forwarding*
CLA	*Convergence Layer Adapter*
LTP	*Licklider Transmission Protocol*
BER	*Bit Error Rate*
DRSS	*Directional Routing and Scheduling Scheme*
MB	MegaBytes
MP3	*MPEG 1 Layer-3*
KB	KiloBytes
TTL	*Time To Live*
FC	*Fountain Codes*
RWP	*Random WayPoint*

LISTA DE SÍMBOLOS

f	Probabilidade de falha
q	Tamanho do alfabeto
l	Número de bits do pacote
E	Pacotes em excesso
k	Número de pacotes originais
λ	Intensidade entre encontros ou taxa de geração de pacotes
$X_i(t)$	Fração dos nós que possuem i primeiros pacotes no tempo t
T_c	Tempo de contato
$a_{i,i-j}$	Probabilidade de se repassar i pacotes em $i-j$ contatos
Θ_i	i-ésima auto-convolução
h_r	Número de ocorrências do inteiro r nas partições
$v(n,k)$	Número de partições ordenadas n em k partes
$p(n,k)$	Número de partições de n em k partes
T_K	Atraso da entrega da mensagem
$D_X(t)$	Probabilidade de sucesso (caso sequencial)
$X_K(t)$	Fração dos nós que contém todas as K mensagens
p	Probabilidade de sucesso na transmissão de um pacote
$X_M(t)$	Fração de nós que possuem mais de M pacotes codificados
t	Tempo para fechamento da fonte
α	Diversidade de informações
ψ_p	Tempo de vida (em segundos) das mensagens sem o PTTA
M	Número máximo de mensagens que coexistem na rede
Δg_P	Intervalo de tempo entre a geração de uma mensagem e outra usando o PTTA
ψ_r	Tempo de vida das mensagens sem o uso do PTTA
Δg_r	Intervalo de geração das mensagens sem o PTTA
α_{PTTA}	Diversidade de informações usando o PTTA
Δt	Tempo de geração de mensagens
TTL_{fluxo}	Tempo de vida do fluxo de mensagens
Δt_{PTTA}	Tempo de parada da geração de segmentos do PTTA

SUMÁRIO

1 INTRODUÇÃO

A distribuição de conteúdo, o compartilhamento de recursos e a necessidade de se comunicar fazem com que a cada dia mais pessoas tenham interesse em interagir entre si. Essa necessidade das pessoas interagirem incita o desenvolvimento de novas ferramentas, aplicações e dispositivos.

1.1 CARACTERIZAÇÃO DO PROBLEMA

Atualmente existe uma tendência ao uso dos dispositivos móveis. Essa tendência está criando a necessidade de comunicar diversos dispositivos em uma rede que nem sempre está conectada. Como as redes não são sempre conectadas, muitas vezes, pode não existir um caminho pré-estabelecido entre os nós que desejam se comunicar, ao contrário do que, na maioria das vezes, ocorre nas redes conectadas. Além disso, por não existir um caminho pré-estabelecido entre origem e destino, torna-se difícil o encaminhamento das mensagens e por fim a sua entrega (BURLEIGH et al., 2003).

Ainda mais difícil torna-se o nó de origem saber se a informação foi ou não entregue corretamente. Para isso, nas redes conectadas, muitas vezes o caminho de retorno com mensagens de confirmação é utilizado. Porém, em redes desconexas a entrega dos dados transmitidos já é um desafio, o recebimento das mensagens de confirmação se torna outro problema (LEE et al., 2009).

1.2 MOTIVAÇÃO

Para evitar a necessidade do envio das mensagens de confirmação, uma das formas é evitar problemas na entrega dos dados aprimorando as técnicas de encaminhamento das mensagens. Em geral, esse encaminhamento é realizado por algum método que tenta prever um possível caminho para atingir o destino, visto que, geralmente, não existe um caminho estabelecido para isso porém, essa previsão pode ser equivocada o que pode levar as mensagens para

um lugar isolado (FRANÇOIS; LEDUC, 2009). Para contornar esse problema, alguns métodos podem ser utilizados como, por exemplo, o armazenamento do comportamento passado (ou histórico) da rede para que estatisticamente tente-se aumentar a probabilidade de acertos nessa previsão (KALANTARI; LA, 2008).

Essa abordagem, nos dias atuais, é aplicada utilizando estudos de redes sociais e do comportamento dos usuários onde consegue-se até prever um certo padrão nos hábitos das pessoas. Porém, apesar de funcionar para a maioria das vezes, não se pode generalizar e manter apenas o foco em pessoas que tenham um comportamento realmente periódico, pois afinal de contas, existem feriados onde as pessoas frequentam lugares distintos e além do mais as pessoas mudam seus hábitos, o que ocasionaria um caos na comunicação. Ainda com relação ao comportamento social e ao uso das redes sociais, não é difícil hoje em dia encontrarmos diversos usuários compartilhando informações entre seus grupos de amigos, colegas de trabalho, etc. Atualmente, para compartilhamentos realizados entre esses grupos utiliza-se um acesso à Internet, como 3G, Wi-fi, entre outros. Porém, quando se deseja disseminar informações para todos os amigos de um determinado grupo, e nenhuma conexão à Internet está disponível, isso torna-se um desafio.

Ainda com relação a dificuldades encontradas, são exemplos, realizar atualização de firmware, de aplicativos, transmitir vídeo, imagem e som, ou até mesmo trocar informações de jogos entre dispositivos móveis sem conexão constante.

Outra alternativa para melhorar o encaminhamento das mensagens é a inundação da rede com cópias da mensagem para que, com efeito, alguma destas atinjam o destino. Isso aumenta, e muito, a probabilidade da entrega, porém gera um outro problema que é a sobrecarga na rede com cópias da mensagem que ficam, por vezes, sendo propagadas afim de que sejam entregues ao destino. E, infelizmente, quando estas mensagens forem entregues elas não serão mais necessárias, pois o destino já terá recebido outra cópia da mensagem anteriormente.

Em geral, a primeira vista um caminho promissor e interessante de seguir seria tentar melhorar a maneira do encaminhamento das mensagens, ou das cópias das mensagens, ou ainda melhorar a forma de se prever o comportamento da rede, para assim aumentar a taxa de entrega dos dados. Existem inúmeras pesquisas na área que tentam desenvolver novos protocolos de roteamento e métodos para melhorar a previsão sobre o comportamento dos nós na rede e a diminuição das cópias das mensagens visando reduzir a carga nos nós intermediários (SAMUEL et al., 2009) (FRANÇOIS; LEDUC, 2009) (BOICE et al., 2007) (NELSON et al., 2009) (CHEN; CHAN, 2009).

1.3 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

O objetivo geral desse trabalho é criar um protocolo de transporte que consiga ter uma garantia estatística configurável de entrega de conteúdo sem mensagens de retorno em redes oportunistas.

Alguns objetivos específicos são:

1. Que este protocolo seja versátil para que possa ser utilizado em diversas aplicações;
2. Adaptativo dependendo da rede onde é utilizado;
3. Econômico, use pouco recurso da rede para o seu funcionamento;
4. Que este protocolo seja robusto e consiga facilmente se recuperar de falhas;
5. Simples, torne o compartilhamento de conteúdo entre membros de grupos sociais mais fácil;
6. Possibilitar a comunicação de áreas remotas e regiões de calamidade;
7. Reduzir custos no uso de alguns aplicativos realizando a comunicação direta entre dispositivos;
8. Viabilizar o desenvolvimento de novas aplicações que utilizem o comportamento social do usuário para realizar a comunicação com os seus grupos de afinidade.

1.4 CONTRIBUIÇÕES

Ao contrário das pesquisas que tentam desenvolver novos protocolos de roteamento e métodos para melhorar a previsão sobre o comportamento dos nós na rede, este trabalho procura aprimorar a maneira com que as mensagens são geradas, para que independentemente do protocolo de roteamento seja alcançada uma melhora na taxa de entrega das mensagens e que esta taxa de entrega seja tolerante a uma certa quantidade de perda. Para isso este trabalho consiste na elaboração de um protocolo de transporte para DTN, sendo possível o seu uso independente de qual protocolo de roteamento será utilizado.

1.5 ORGANIZAÇÃO DO DOCUMENTO

O restante deste documento está organizado da seguinte forma: o capítulo 2 apresenta os principais conceitos envolvidos neste trabalho, que se fazem necessários para o seu

entendimento, e a descrição dos protocolos de transporte mais utilizados e conhecidos, além dos trabalhos correlatos; o capítulo 3 inicia especificando o protocolo de transporte proposto, seguido do detalhamento da proposta deste trabalho com o funcionamento do protocolo com todas as suas possíveis abordagens e aplicações além do controle da fonte desenvolvido, onde é descrito um modelo matemático para a distribuição das informações em uma rede tolerante a atrasos, que é composta por um conjunto fechado de equações que descrevem o comportamento estatístico de redes desse tipo incluindo o impacto da fragmentação das mensagens, em seguida encontra-se a descrição das equações do controle da fonte; o capítulo 4 é formada pela descrição dos protocolos de roteamento utilizados para realização dos testes, com o objetivo de mostrar que o protocolo de transporte desenvolvido é realmente independente, após, serão descritos os cenários utilizados na avaliação da proposta, seguido pelos resultados e discussão dos resultados apresentados com a avaliação do desempenho; por fim, o capítulo 5 ressalta os avanços obtidos com a pesquisa, suas particularidades e mostra ainda indicações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão descritos os elementos necessários para o entendimento do trabalho, bem como o estado da arte. Alguns tópicos muito difundidos, porém não menos importantes, seguem com suas devidas referências para futuras consultas, caso necessário.

2.1 REDES TOLERANTES A ATRASOS

Pesquisas significantes envolvendo redes com atrasos e desconexões entre seus nós começaram a ser discutidas em 1998 por um grupo de pessoas no JPL (*Jet Propulsion Laboratory*), que iniciaram o desenvolvimento da IPN - *Interplanetary Network* (FALL; FARRELL, 2008). Segundo Kevin Fall, no princípio, os interesses da arquitetura destas redes eram focados em tolerância a grandes atrasos e comunicações com interrupção previsível (*e.g.*, comunicações espaciais) (FALL; FARRELL, 2008). Nesta época da história, os esforços eram despendidos para a criação de uma arquitetura de rede interplanetária (IPN). Neste momento surgia o IPNSIG (*IPN Special-Interest Group*) que desenvolveu uma arquitetura para uma rede de larga escala, onde foram obtidos alguns progressos, porém, devido ao alto custo para se implementar esta rede interplanetária, este projeto acabou arquivado.

Em 2003, Kevin Fall, um dos autores da RFC (*Request For Comments*) 4838 (CERF et al., 2007), utilizou pela primeira vez o termo *Delay Tolerant Networking*, sugerindo que este tipo de rede fosse utilizado em outras situações além da comunicação interplanetária como, por exemplo, a comunicação das redes terrestres sem fio. Estas redes também sofrem interrupções e atrasos. Com isto o foco da arquitetura cresceu de apenas redes com comportamento periódico, como era o caso das IPNs, para resolver os problemas de outros tipos de redes com outros padrões de conectividade, como as redes de sensores, por exemplo. Levando em conta a sugestão de Fall, o IRTF (*Internet Research Task Force*) criou um grupo de pesquisa para este tipo de redes, o qual foi chamado DTNRG - *Delay Tolerant Networking Research Group*. Este grupo de pesquisas é responsável por documentar os protocolos de redes deste tipo em RFCs.

Visando à compatibilidade com a arquitetura TCP/IP, o grupo de pesquisas DTNRG

definiu uma sobrecamada (*overlay*) de agregação (chamada *bundle layer*) localizada logo acima da camada de transporte da arquitetura TCP/IP e abaixo da camada de aplicação. Esta camada de agregação é definida em (SCOTT; BURLEIGH, 2007).

Nos últimos anos, as redes tolerantes a atraso (*Delay* ou *Disruption Tolerant Networking*, ambas conhecidas por DTN) se tornaram uma linha de pesquisa interessante, tanto para desenvolvedores de aplicações e protocolos, como para projetistas de rede (FALL; FARRELL, 2008), (PUJOL et al., 2009). Isso se deve às experiências práticas adquiridas no uso das redes *ad-hoc* móveis, que possuíam certas situações onde não era possível obter uma conectividade fim-a-fim durante todo o tempo (FALL; FARRELL, 2008) (HOLLIDAY, 2009). As redes *ad-hoc* móveis (MANET - *Mobile Ad-hoc Networks*) (NORDEMANN; TONJES, 2012) consistem em nós móveis autônomos, conectados por canais sem-fios, sem qualquer infraestrutura de rede pré-existente (DU et al., 2009) (WYATT et al., 2009) (CHUAH et al., 2009) (CHUAH; XI, 2007).

Deste modo, a arquitetura DTN (definida através da RFC 4838 (CERF et al., 2007)) não surgiu apenas para lidar com o atraso encontrado nas conexões de rede, mas também para prover uma arquitetura que suportasse a heterogeneidade encontrada em *gateways* de redes de sensores e redes móveis (FALL; FARRELL, 2008).

Uma rede tolerante a atraso, pode ser definida como sendo uma rede que visa à comutação de dados onde os pontos origem e destino não necessariamente estejam conectados (no ato da transmissão) e visíveis um ao outro (GUO et al., 2008) (HOLLIDAY, 2009). Isto pode gerar um atraso de segundos, minutos, horas, etc. na troca de informações (CHOI; SHEN, 2009) (WYATT et al., 2009) (YU; KO, 2009).

Ao contrário das redes cabeadas normais, a propagação das informações em uma DTN ocorre através de um tipo de custódia dos dados. O nó origem envia as mensagens para o nó destino, mesmo sem saber onde este se encontra, sem saber o caminho até ele ou até mesmo se ele está ativo. Essas mensagens são confiadas a nós intermediários chamados *relays*. Estes *relays* repassam a outros nós intermediários e assim sucessivamente, até que as mensagens atinjam o nó destino. Esse é um método parecido com o que é utilizado nas redes IPs (*Internet Protocol*), o *store-and-forward*, porém esse é tolerante a atrasos maiores do que os tolerados pelas redes IPs e recebe o nome de *store-carry-forward* (FALL; FARRELL, 2008). A forma com que a origem e os *relays* elegem os nós para os quais eles encaminharão os pacotes depende do protocolo de roteamento utilizado.

2.2 PROTOCOLOS DE TRANSPORTE NA INTERNET

Os protocolos de transporte mais utilizados em redes conectadas são: UDP (*User Datagram Protocol*) (POSTEL, 1980) e TCP (*Transmission Control Protocol*) (POSTEL, 1981). A seguir serão descritos sucintamente os dois protocolos que servirão como base para a elaboração do novo protocolo de transporte proposto.

2.2.1 UDP

O UDP oferece um serviço sem conexão e sem confirmação para a entrega de dados, sendo assim, sem garantia de entrega. Este protocolo de transporte foi padronizado em (POSTEL, 1980). Este protocolo oferece às aplicações um serviço direto à entrega de datagramas. O cabeçalho deste protocolo conta com poucos campos, sendo eles: porta de origem; porta de destino; comprimento da mensagem e; verificação de paridade (*checksum*).

A Figura 1 mostra os campos do protocolo UDP.

Porta de origem	Porta do destino
Tamanho da mensagem	Checksum
Dados	

16 bits	16 bits

Figura 1: Cabeçalho do protocolo de transporte UDP.

Os campos porta de origem e porta de destino são importantes pois é a maneira pela qual a camada de transporte sabe para qual aplicativo os dados são destinados e de qual aplicativo eles são originários, funcionando como uma multiplexação/demultiplexação de informações para aplicativos. Tamanho da mensagem especifica o tamanho da mensagem incluindo o cabeçalho.

Ao utilizar o protocolo UDP, os campos porta de origem e *checksum* são opcionais pois a porta de origem é útil caso uma resposta seja necessária e o *checksum* serve para validar a mensagem e verificar se existem erros (POSTEL, 1980).

O protocolo UDP é largamente utilizado em comunicações em tempo real como, por exemplo, aplicações de voz e vídeo que possam ter alguma perda. Em geral, aplicações que não necessitam receber mensagens na sequência correta e que toleram perda de algumas mensagens são as que mais utilizam este protocolo. Uma das vantagens é o fato de não estabelecer conexão, o que poupa o envio de algumas mensagens que podem ser utilizadas para transmitir dados ao invés de informações para o estabelecimento da conexão.

2.2.2 TCP

Ao contrário do UDP, o TCP proporciona um serviço orientado a conexão e com confirmação, afim de assegurar a entrega das informações. O protocolo TCP foi padronizado em (POSTEL, 1981) e já foi atualizado por (BRADEN, 1989), (RAMAKRISHNAN et al., 2001), (GONT; YOURTCHENKO, 2011) e (GONT; BELLOVIN, 2012).

O TCP realiza o estabelecimento de conexões e, por muitas vezes, verifica a entrega das informações utilizando o envio de mensagens do tipo "ACK" (de confirmação) do nó destino para o origem, ou seja, com canal de retorno.

A Figura 2 mostra os campos que formam o cabeçalho do TCP.

<table>
<tr><td colspan="3">Porta de origem</td><td>Porta de destino</td></tr>
<tr><td colspan="4">Número de sequência</td></tr>
<tr><td colspan="4">Número de reconhecimento</td></tr>
<tr><td>Comp do cab</td><td>Res</td><td>Flags</td><td>Tamanho da janela</td></tr>
<tr><td colspan="3">Checksum</td><td>Ponteiro de urgência</td></tr>
<tr><td colspan="4">Opções</td></tr>
<tr><td colspan="4">Dados</td></tr>
<tr><td colspan="3">16 bits</td><td>16bits</td></tr>
</table>

Figura 2: Campos do cabeçalho do protocolo de transporte TCP.

Os campos Porta de origem, Porta de destino, Checksum e Dados são equivalentes aos mesmos campos do UDP. Além destes campos, o cabeçalho TCP contém os campos descritos a seguir.

Número de sequência é utilizado para identificar qual é o primeiro Byte do segmento que está na mensagem atual. Isso é importante para a confiabilidade na transmissão dos dados com o TCP e esta segurança é complementada com o uso do campo Número de reconhecimento que serve para avisar a origem que os dados foram recebidos corretamente.

O campo Tamanho da janela é usado para o controle de fluxo, e seu objetivo é limitar a quantidade de informações recebidas pelo destino. Comp do cab significa comprimento do cabeçalho e é especificado em quantidade de palavras de 32 bits, este campo é necessário pois o campo Opções é variável. Opções é um campo opcional e de comprimento variável, maiores informações sobre este campo podem ser obtidas em (POSTEL; REYNOLDS, 1983) e (JACOBSON et al., 1992). Os campos flags servem para, por exemplo, estabelecer e interromper conexões, para marcar os dados como "urgentes", para marcar que os dados contidos são vá-

lidos, entre outros. A parte delimitada como Res significa reservado, e foi reservada para uso futuro.

2.3 CÓDIGOS CORRETORES DE ERROS

Como a transmissão de mensagens em redes está sujeita a fatores externos, como interferência, ruído, etc., os dados enviados podem, muitas vezes, não chegar ao destino ou chegar com erros. Na sequência serão descritos os canais com apagamento, que serão utilizados como uma aproximação do comportamento da transmissão em DTN

2.3.1 CANAIS COM APAGAMENTO

Canais com apagamento são aqueles onde os dados enviados podem ser recebidos, apagados ou perdidos no caminho (MACKAY, 2005). A importância do estudo desses canais reside no fato das redes sem fio se comportarem como canais deste tipo, pois os canais ruidosos podem ser vistos desta forma. Um modelo simples de canal com apagamento pode ser visto na figura abaixo, onde existe (para todo o alfabeto de entrada *{0, 1, 2, ..., q - 1})* uma probabilidade *1 - f* de transmitir a entrada sem erro, e a probabilidade *f* de entregar a saída '?' (ou desconhecida, ou ainda com erro) (MACKAY, 2003). Na Figura 3 estão ilustradas as possibilidades existentes nos canais com apagamento. O tamanho do alfabeto q é 2^l, onde l é o número de bits no pacote.

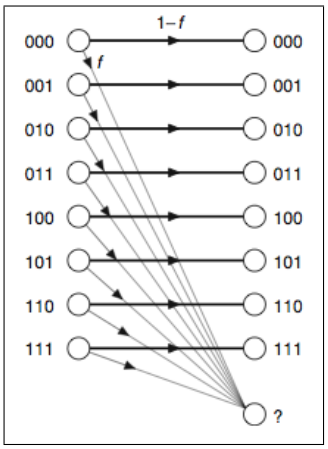

Figura 3: Exemplo de um canal com apagamento para mensagens de 3 bits. Fonte: (MACKAY, 2005).

A comunicação neste tipo de meio utiliza-se de mensagens de retorno, do receptor para o nó de origem que são usadas para controlar a retransmissão das mensagens apagadas. Por exemplo, o receptor deve enviar mensagens de retorno para identificar os pacotes faltantes,

os quais serão retransmitidos. Como outra alternativa, o receptor pode enviar mensagens de confirmação a cada pacote recebido corretamente, onde o nó de origem faz um controle das mensagens confirmadas e retransmite as não confirmadas até obter a confirmação de todas as mensagens. Para se detectar se ocorreram erros na transmissão os códigos verificadores de erros são utilizados. Esses códigos são capazes de reportar ao receptor a inconsistência nos dados de um pacote, o que implica na necessidade de sua retransmissão.

Os protocolos que utilizam o canal de retorno possuem uma vantagem, eles funcionam independentemente da probabilidade de apagamento f. Se a probabilidade de apagamento f for grande, duas situações podem ocorrer: i) no protocolo que envia mensagens de confirmação de recebimento de pacotes, o nó de destino receberá várias cópias da mesma mensagem. ii) Já naquele que envia mensagens dos pacotes que deverão ser retransmitidos, o número de mensagens de retorno enviadas ao nó de origem será grande. Mas de acordo com a teoria de Shannon não é necessário um canal de retorno: a capacidade do canal de transmissão é $(1-f)\times l$ bits, tendo ou não o retorno, visto que f é a probabilidade de apagamento (SHANNON, 1948). E a comunicação bem sucedida pode ocorrer a esta taxa desde que conte com a ajuda do código corretor de erros apropriado (MACKAY, 2003) (RAMOS, 2010). O código corretor de erros é, como seu próprio nome diz, um código para detecção e correção de erros. Existem diversos códigos deste gênero na literatura, porém aqui neste trabalho, serão utilizados os códigos fontanais, descritos adiante.

O desperdício da simples retransmissão é mais evidente quando se tem o envio de uma informação em *broadcast*, onde cada um dos receptores recebe um fragmento aleatório da informação com probabilidade de sucesso *(1 - f)*. Se cada um dos pacotes não recebidos por cada nó receptor tiver que ser retransmitido, estas retransmissões gerarão uma infinidade de redundância. E, além do mais, cada receptor já terá recebido muitos dos pacotes retransmitidos.

Devido a estes fatos é importante utilizar algum tipo de código corretor de erro que necessite de pouco ou nenhum retorno. Os mais clássicos são os Códigos de Reed-Solomon (MACKAY, 2003). Estes códigos possuem a propriedade de que se X símbolos codificados de um conjunto de N símbolos transmitidos forem recebidos, então os X símbolos originais poderão ser recuperados. Eles são usuais para alfabetos e número de pacotes pequenos, porque o custo computacional para a sua utilização é muito grande (quadrático) e o número de combinações (pacotes codificados) possíveis é limitado. Além disso, para se utilizar esta abordagem, deve-se estimar a taxa de perda de pacotes, no meio de transmissão, previamente. Por este motivo, não se pode controlar em tempo real.

Para contornar este problema foram desenvolvidos os chamados códigos fontanais. Es-

tes códigos foram iniciados por Michael Luby em 2002 com os códigos LT (Luby Transform), que serão explicados adiante (LUBY, 2002). A ideia dos códigos deste tipo é que o nó de origem funcione como uma fonte que fica jorrando informações o tempo todo e que haja informações redundantes em cada informação jorrada. O objetivo disso é que após algum tempo, qualquer destinatário que receber um número k de informações desta fonte (onde k é o número de pacotes originais), possa recuperar toda a informação original. Em seguida será explicado o funcionamento destes códigos e algumas formas de códigos fontanais existentes.

2.4 CÓDIGOS FONTANAIS

Os códigos fontanais ou *Fountain Codes* são uma metáfora de uma fonte, pois a origem (fonte) dispara várias "gotas" (pacotes codificados) nos destinos, onde estes, por sua vez, ficam capturando (recebendo) estas "gotas" até que possam decodificar as informações transmitidas pela fonte (MACKAY, 2003).

O procedimento real é o nó de origem (fonte) transmitindo diversos pacotes de informação codificada (gotas) na rede, onde, qualquer destino que queira receber a informação precisa receber e armazenar em seu buffer (copo) a quantidade um pouco maior que k (cerca de 5% maior) de mensagens codificadas afim de decodificar toda a informação original (MACKAY, 2003) (MACKAY, 2005).

O número de pacotes codificados que podem ser gerados é ilimitado e pode ser determinado em tempo real (a forma com que os pacotes são gerados será explicado a seguir). Além disso, os códigos fontanais são universais, pois são próximos do ótimo para qualquer canal com apagamento. Levando em consideração a estatística do meio de transmissão, pode-se transmitir tantos pacotes quantos sejam necessários para o destino decodificar os dados. Ou seja, a transmissão cessa quando estatisticamente o destino conseguiu decodificar os dados, caso contrário ela continuará (MACKAY, 2005).

A principal característica dos códigos fontanais está relacionada com a decodificação (recuperação) dos dados originais. Esta recuperação da informação pode ser obtida utilizando-se qualquer número de pacotes codificados (independente da sequência) desde que seja um pouco maior do que k. Além disso, este tipo de codificação pode ter complexidade muito pequena na codificação e decodificação, pois geralmente as operações realizadas são operações lógicas do tipo *XOR* (ou-exclusivo) (MACKAY, 2003) (MACKAY, 2005) (LUBY, 2002).

Segundo (MITZENMACHER, 2004), uma fonte ideal precisa ter as seguintes propriedades:

- A fonte deve ser capaz de gerar um repertório potencialmente infinito de pacotes codificados a partir dos dados originais.
- Qualquer combinação de k pacotes deve proporcionar a reconstrução de todos os dados originais sendo k o tamanho do buffer. Além disso, esta reconstrução precisa também ser extremamente rápida, preferencialmente linear a k.

Ao utilizar os códigos fontanais obtém-se algumas vantagens, sendo elas:

- a possibilidade quase infinita de combinações entre as partes para geração de pacotes codificados;
- a liberdade de se controlar em tempo real quantos pacotes codificados que serão gerados;
- a velocidade e baixo uso de processamento, visto que, em geral, este se resume em funções XOR (ou-exclusivo);
- a independência de quais serão os pacotes codificados recebidos para se poder realizar a decodificação;
- a garantia do recebimento das informações de maneira correta;
- a segurança das informações, pois os dados são enviados codificados e podem ser criptografados anteriormente;
- a não necessidade de confirmação do recebimento dos pacotes codificados visto que não importam quais foram perdidos, as informações poderão ser recuperadas com praticamente qualquer combinação deles;
- não necessita ordenação uma vez que os pacotes codificados capturados serão depois processados para remontar a informação original;
- o envio de mensagens *multicast* torna-se muito mais fácil e eficiente, pois as informações podem ficar continuamente sendo enviadas, da origem, o tempo que for necessário para a informação ser disseminada;
- o destino necessita apenas de cerca de 5% de informação redundante (a mais) para decodificar a informação original (MACKAY, 2003).

2.4.1 *RANDOM LINEAR FOUNTAIN*

O funcionamento deste tipo de códigos fontanais se dá através da geração de uma matriz crescente, de k bits (linhas) por largura sempre crescente e podendo ser infinita, Figura 4. Esta matriz gerada é resultante da operação *XOR* entre os pacotes (MACKAY, 2003).

O codificador e o decodificador possuem um gerador de números pseudo-aleatórios que são sincronizados, ou seja, eles geram a mesma sequência de números aleatórios. Além disso, como opção, o transmissor pode selecionar uma chave gerada por um processo pseudo aleatório e enviar esta chave no cabeçalho dos pacotes, que servirá para a sincronização dos geradores pseudo-aleatórios do codificador e decodificador (MACKAY, 2005).

Quando os pacotes são transmitidos, alguns não são recebidos (as colunas grifadas em cinza na Figura 4), porém, é possível realinhar as colunas para definir a matriz geradora do ponto de vista do receptor (matriz inferior da Figura 4).

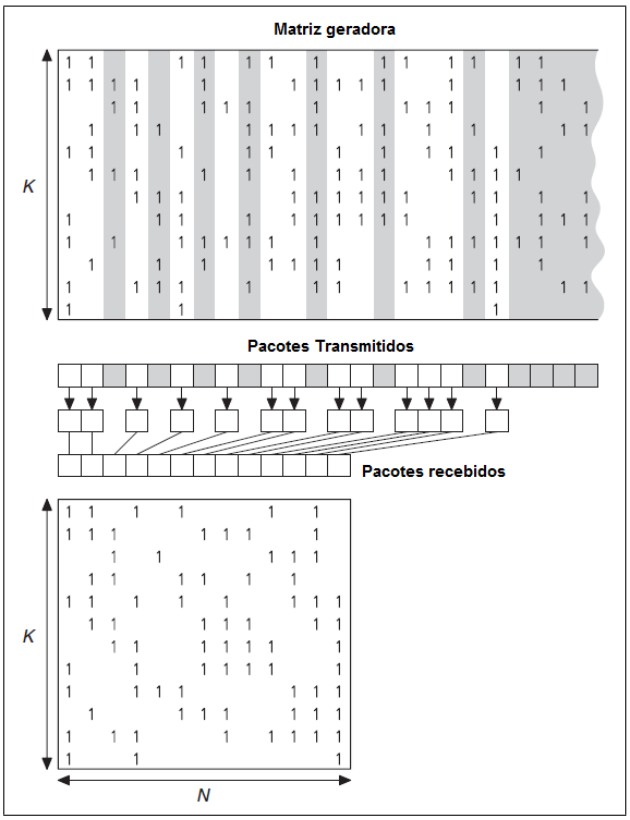

Figura 4: Matriz geradora de um *Random Linear Fountain*. Fonte: Tradução de (MACKAY, 2005).

A probabilidade da decodificação completa dos dados não ocorrer está representada pela linha sólida na Figura 5 em função do número de pacotes em excesso E (redundantes). A linha pontilhada mostra o limite superior, 2^{-E} na probabilidade de erro (MACKAY, 2005).

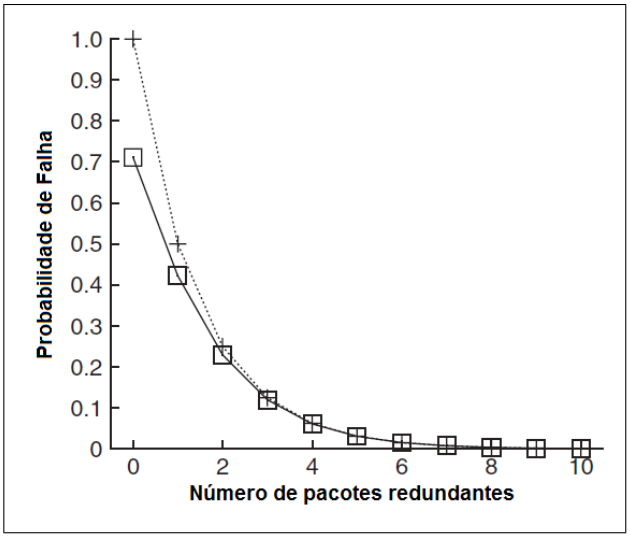

Figura 5: Desempenho do *Random Linear Fountain*. Fonte: Tradução de (MACKAY, 2005).

O método *Random Linear Fountain* é muito interessante para aplicações onde k é muito grande. Por outro lado, suas desvantagens incluem o custo computacional quadrático e cúbico para a codificação e decodificação. Este custo computacional elevado, torna-se um problema nesse tipo de códigos fontanais, por isso foram desenvolvidos outros que serão descritos adiante.

2.4.2 *LT CODES*

Os códigos chamados LT - Luby Transform (Transformada de Luby) recebem este nome devido ao seu criador Michael Luby (2002) (LUBY, 2002) e foram os primeiros códigos ditos *rateless* (códigos com taxa variável, segundo (LUBY, 2002) e (RAMOS, 2010)), este código recebe esta classificação por gerar tantos pacotes codificados quantos forem necessários à decodificação da informação original. Além disso, este código foi o originário do nome *Fountain Codes* (códigos fontanais).

A codificação dos códigos LT se dá da seguinte maneira. Primeiro separa-se os dados em k partes da informação original. Em seguida é realizada a combinação desses pacotes para gerar os pacotes codificados. Estes pacotes codificados são simples aplicações da operação ou-exclusivo (XOR) entre as partes originais envolvidas neste pacote codificado. Quantas partes originais estarão envolvidas em cada pacote codificado é dito grau do nó, onde o nó é o pacote original. Portanto, nos códigos do tipo LT a codificação é do tipo $y_6 = x_1 \oplus x_9 \oplus x_3$, onde $\oplus$ é a operação binária Ou-exclusivo (XOR), y_i representa os dados com redundância (codificados) e x_j são os dados reais (originais).

Já a decodificação dos códigos LT ocorre com o recebimento dos pacotes codificados e realiza-se a mesma operação XOR entre as partes até que se recupere todas as k partes da

informação original.

Segundo (LUBY, 2002) se os dados originais consistirem em k símbolos de entrada então cada símbolo codificado pode ser gerado, independentemente de todos os outros símbolos codificados, na média por $O(\ln(k/f))$ operações, e os k símbolos de entrada podem ser recuperados a partir de quaisquer $k + O(\sqrt{k}\ln^2(k/f))$ símbolos codificados com probabilidade $1 - f$ com uma média de $O(k \cdot \ln(k/f))$ operações, onde f é a probabilidade da decodificação não ocorrer (falha).

Na Figura 6 mostra um exemplo da decodificação utilizando um código fontanal do tipo LT, onde na Figura 6*a* são mostrados os bits originais s_1, s_2 e s_3 na parte superior, na parte inferior são exibidos os bits codificados (*1*, *0*, *1* e *1*) e as linhas que aparecem são os bits originais que foram utilizados para a operação *XOR* e que deram origem aos respectivos bits codificados. Na Figura 6*b*, como o primeiro bit codificado tem ligação apenas com o bit original s_1, sabe-se que $s_1 = 1$, assim é realizada a operação *XOR* em todos os bits codificados que se ligam com s_1 e o grau destes bits são diminuídos (Figura 6*c*). Na Figura 6*d*, como o último bit codificado só possui ligação com o bit s_2, este recebe o valor do último bit codificado e as operações realizadas para s_1 se repetem, até que obtém-se todos os k bits originais na Figura 6*f*.

2.4.3 *TORNADO CODES*

Nos códigos do tipo tornado a codificação é semelhante à dos códigos LT, do tipo $y_6 = x_1 \oplus x_9 \oplus x_3$, onde $\oplus$ é a operação binária Ou-exclusivo (XOR), y_i representa os dados com redundância (codificados) e x_j são os dados reais (originais). Os códigos tornado ainda podem usar codificação do tipo $y_{34} = y_3 \oplus y_{12} \oplus y_9$, ou seja, informações redundantes geradas a partir de informações redundantes. O tempo de codificação é determinado pelo número de operações XOR no sistema de equações (BYERS et al., 1998a) (BYERS et al., 1998b).

Na decodificação dos códigos tornado são utilizadas duas operações básicas. A primeira consiste em substituir as variáveis recebidas pelos seus valores nas equações onde elas aparecem. E a segunda é uma regra simples de inversão. Esta regra pode ser aplicada para recuperar qualquer variável faltante que esteja em uma equação onde aquela variável é a única ausente. Por exemplo, considerando $y_6 = x_1 \oplus x_9 \oplus x_3$, supondo que um pacote redundante contendo y_6 tenha sido recebido, assim como os pacotes contendo os dados reais x_1 e x_3 mas não tendo sido recebido x_9. Então é possível usar a equação acima para encontrar x_9, usando apenas operações XOR. Portanto, usando essa regra de inversão repetidamente, o recebimento de apenas um pacote pode permitir a reconstrução de vários pacotes adicionais, conforme o recebimento deste pacote é propagado. Na prática, o número de aplicações possível dessa regra

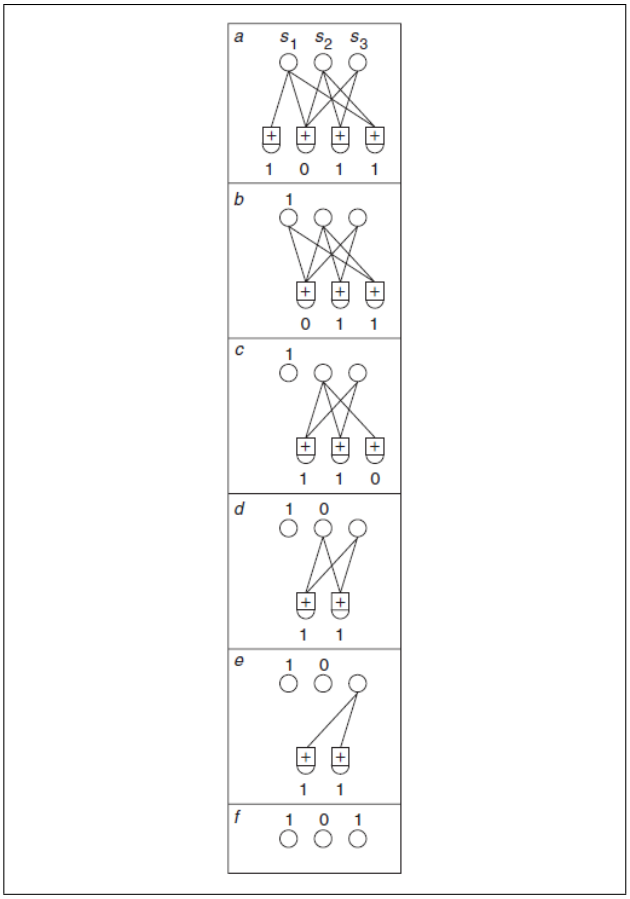

Figura 6: Exemplo de decodificação para um código fontanal do tipo LT com $k = 3$ bits originais e $N = 4$ bits codificados. Fonte: Tradução de (MACKAY, 2005).

de inversão permanece mínima até que um pouco mais de k pacotes (onde k é o número de pacotes originais) tenham sido recebidos. E aí cada simples recebimento de pacote gera um redemoinho de inversões (ou seja, inversões em vários pacotes) que permite a recuperação de todos os dados originais faltantes. Daí deriva o nome de códigos TORNADO (BYERS et al., 1998a) (BYERS et al., 1998b).

2.5 TRABALHOS CORRELATOS

O uso de códigos corretores de erros em DTNs já tem sido objeto de estudo (WANG et al., 2005), (JAIN et al., 2005), (ALTMAN; PELLEGRINI, 2009), (DAI et al., 2010) e (CHAHIN et al., 2011) .

Wang et al. comparam o ganho no desempenho com códigos corretores de erros à replicação simples, ou seja, o envio de réplicas adicionais da mesma mensagem. Para comprovar os benefícios de utilizar os códigos corretores de erros, são realizadas várias simulações, usando vários protocolos de roteamento (WANG et al., 2005).

Jain et al. levaram em conta o problema de padrões não uniformes de encontros entre os nós. Além disso, mostraram que existe uma dependência muito forte da probabilidade de entrega das informações com a distribuição das réplicas pelos vários caminhos da rede. Os autores avaliam várias técnicas de alocação dessas réplicas e é provado que o problema de alocação das informações na rede é do tipo *NP-hard* (JAIN et al., 2005).

De Pellegrini e Altman descreveram o formalismo estatístico do desempenho das DTNs. Ainda levam em conta o uso dos códigos corretores de erros e dos códigos fontanais neste formalismo. Este é o primeiro trabalho encontrado que comenta a respeito do uso de códigos fontanais em DTN (ALTMAN; PELLEGRINI, 2009).

Dai et al. propõem a integração dos códigos fontanais com o protocolo *Optimal Probabilistic Forwarding* (OPF). Os dados são codificados e uma regra de encaminhamento é definida para decidir para onde um pacote será enviado (DAI et al., 2010).

Chahin et al. levaram em conta redes DTN heterogêneas formadas por diferentes classes de nós. Nesse trabalho é proposta uma maneira de se entregar os dados, da origem ao destino, levando em conta a energia dos nós. Ainda esse método é dito "CEGO" porque o próprio algoritmo guia o nó origem para o ponto ótimo de operação sem a necessidade de estimar explicitamente os parâmetros da rede como intensidade entre encontros e o número de nós em cada classe (CHAHIN et al., 2011).

Dvir e Vasilakos consideram uma abordagem alternativa chamada de roteamento "back-pressure", no qual as decisões de encaminhamento são realizadas com base em cada pacote. Algumas informações sobre atraso, caminho percorrido e agendamento de pacotes de dados são usados nas decisões de roteamento e encaminhamento. Os resultados das simulações são usados para confirmar os benefícios obtidos (DVIR; VASILAKOS, 2010).

Spyropoulos et al. dividiram as estratégias de roteamento existentes em um pequeno número de partes comuns chamadas "módulos de roteamento", em seguida, eles mostram como um dado módulo de roteamento deve ser aplicado, de acordo com um conjunto de características da rede presentes na aplicação sem fios. Depois disso, eles tentaram identificar características genéricas da rede que fossem realmente relevantes para o processo de roteamento. A idéia era identificar um conjunto de características úteis para ajudar o usuário a escolher um protocolo de roteamento apropriado tanto para o aplicativo em uso quanto para rede (SPYROPOULOS et al., 2010).

Sun et al. estudaram o desempenho dos protocolos da *convergence layer adapter* (CLA) nas DTNs para transmissão confiável de dados sobre a infra-estrutura de comunicação

espacial constituída por taxas assimétricas de canais. Eles também deram uma atenção especial para o recém-desenvolvido *Licklider Transmission Protocol* (LTP) CLA (chamado LTPCL). O desempenho do LTPCL foi comparado com outros CLAs confiáveis, uma combinação de TCP CLA com LTPCL, e avaliados onde havia uma taxa de canais altamente assimétrica nas comunicações cislunar com atraso longo. O LTPCL também foi estudado em um cenário de comunicação do espaço profundo que caracteriza-se por um tempo de atraso e interrupção muito grande no link (SUN et al., 2013).

Wang et al. apresentaram uma avaliação experimental do protocolo Bundle (BP) em diferentes protocolos da camada de convergência. Eles usaram um cenário de comunicação cislunar simulado com diferentes atraso de propagação do sinal e perda de dados. Os autores visam a camada de convergência LTP (LTPCL) rodando em cima de UDP/IP (BP/LTPCL/UDP/IP). O desempenho de BP/LTPCL/UDP/IP na transferência de dados realistas é comparada com as duas outras pilhas de protocolo DTN, a saber, a BP/TCPCL/TCP/IP e BP/UDPCL/UDP/IP. Os resultados mostraram que o LTPCL supera o TCPCL para muitos atrasos de links, independentemente da taxa de erro de bit (BER-*Bit Error Rate*). É também afirmado que, em determinada condição, o LTPCL mostrou uma vantagem significativa no goodput sobre o TCPCL em todos os níveis de atraso analisados para o link. O LTPCL mostrou também uma vantagem significativa consistente no goodput sobre UDPCL em todos os níveis de atraso e BERs analisados (WANG et al., 2011a).

Wang et al. estudaram os protocolos da camada de convergência DTN em uma infraestrutura de comunicação cislunar onde há muito atraso, várias taxas assimétricas de canais e variável taxa de perda de dados. O objetivo do trabalho era conhecer qual CLP é o melhor para comunicações cislunar com longo atraso com as taxas de canais altamente assimétricas e se o protocolo de agregação (BP) auxilia a melhorar o goodput na presença de assimetria no canal IPN (WANG et al., 2011b)

Wang et al. estudaram o efeito da agregação de pacotes de dados na comunicação espacial com canais com taxas assimétricas e baixas. Eles também pesquisaram qual é o número de pacotes a serem agregados para as taxas de canal cislunar altamente assimétricos dentro de um bloco visando o melhor desempenho (WANG et al., 2013).

Zeng et al. propuseram não utilizar métricas, tais como atraso, contagem de saltos, e largura de banda para fazer um esquema de roteamento melhor, mas concentrar-se em green communications, como economia de energia, otimizando o desempenho da rede e evitando o aquecimento global. Eles também argumentam que os sistemas de comunicação eficientes são interessantes por causa da poluição, consumo de energia e dissipação de calor. Com isso

em mente, os autores apresentam um esquema de escalonamento e encaminhamento direcional (DRSS-Directional Routing and Scheduling Scheme) para DTNs *green vehicle*. O sistema resolve o problema de roteamento e escalonamento como um processo de aprendizagem pelo roteamento geográfico e controle de fluxo na direção ideal. Para tornar as coisas mais rápidas, eles usaram um método híbrido com encaminhamento e replicação de acordo com o tráfego. A validação do sistema proposto foi feita usando uma DTN veicular com um modelo de mobilidade predeterminado. Os resultados mostram que o DRSS, quando equipado com capacidades de aprendizagem, consegue boa eficiência energética (ZENG et al., 2013).

É importante frisar que até o momento, não foram encontradas na literatura, propostas semelhantes à contida neste trabalho. A grande maioria das pesquisas busca o aprimoramento dos protocolos na camada de rede, enquanto aqui, neste trabalho, é utilizada a camada de transporte (fim-a-fim), o que caracteriza a sua inovação e singularidade. Além disso, foram encontrados outros trabalhos que pesquisam nas camadas inferiores, de enlace de dados e rede, aplicando técnicas de network coding (IQBAL et al., 2011)(KATTI et al., 2006)(CHOU; WU, 2007) como por exemplo, (CHUAH et al., 2009) (ZHANG et al., 2009). Todavia o trabalho que será descrito nesta tese pode ser empregado em conjunto com essas técnicas, tornando possível ganhos em seu desempenho devido a independência das camadas inferiores.

3 PROTOCOLO DE TRANSPORTE TOLERANTE A ATRASOS (PTTA)

Neste capítulo será descrito o desenvolvimento do protocolo de transporte proposto para DTN. Este protocolo foi desenvolvido tendo em mente as características existentes na pilha TCP/IP, por ser confiável, possuir garantia na entrega das informações e por ser muito difundido. Além disso, foi levado em consideração que as redes, às quais será aplicado, possuem características próprias das DTN e por isso, foi projetado para trabalhar com um código corretor de erros do tipo códigos fontanais em seu funcionamento. Tudo isso visando uma confiança estatística na entrega das informações, que pode ser configurada para se adaptar a diferentes cenários.

3.1 ESPECIFICAÇÃO DO PROTOCOLO

Primeiramente é necessário recordar que para obter um serviço de entrega com confiabilidade usando o TCP, é preciso existir ao menos um caminho estável entre a origem e o destino no estabelecimento da conexão para o envio de dados. As redes tolerantes a atrasos e interrupções geralmente são esparsas e desconexas. Nestas redes é comum a inexistência de um caminho estável entre a origem e o destino, sendo muitas vezes impossível o estabelecimento de conexões para envio dos dados e confirmações (ZHANG et al., 2009) (WYATT et al., 2009). Por este motivo, o TCP não é funcional em redes tolerantes a atrasos como uma camada fim-a-fim (FALL, 2003) (BOICE et al., 2007) (CHUAH et al., 2009).

Como nos enlaces DTNs a transmissão ocorre sempre ponto-a-ponto (*relay* a *relay*), com uma camada de agregação chamada *bundle*, é possível a utilização dos protocolos de transporte TCP e UDP porém não de uma maneira fim-a-fim como são utilizados nas redes cabeadas.O protocolo de transporte descrito neste capítulo tem a sua aplicação acima desta camada de agregação, a qual não é obrigatória, de uma maneira realmente fim-a-fim como um protocolo de transporte deve ser (KIM; HAN, 2012) (NORDEMANN; TONJES, 2012).

Além disso, o TCP é um protocolo com confirmação e dependente de número de sequência, em outras palavras, todas as informações recebidas pelo destino devem ser confirma-

das e recebidas em ordem. Se essas informações foram corretamente recebidas, uma mensagem ACK é enviada, se possuírem erros, devem ser retransmitidas.

A dificuldade neste caso é desenvolver um mecanismo eficaz de transmissão das informações, sem a necessidade de confirmação e conexão entre os hospedeiros (origem e destino).

Com o intuito de melhorar a confiabilidade[1] e a entrega das informações para que se aproxime a do TCP (estatisticamente), faz-se necessário o uso de meios aplicáveis em redes tolerantes a atrasos e interrupções. Para se contornar esses problemas de incompatibilidades com a arquitetura das DTNs, deve-se ter em mente que o protocolo a ser desenvolvido não pode ser orientado à conexão e, ainda, não poderá contar com mensagens de confirmação. Isto é devido ao fato de que muitas vezes o caminho de volta pode não existir e isso apenas saturaria a rede com informações de confirmação ao invés de se utilizar os recursos para, efetivamente, entregar outros dados mais relevantes. Desta forma serão utilizadas técnicas para evitar ao máximo os erros e evitar a necessidade de mensagens ACK, para isso serão utilizados os códigos corretores de erros.

O código corretor de erro utilizado para atingir maior taxa de entrega, neste trabalho, foi os códigos fontanais, descritos anteriormente na seção 2.4. Os motivos que levaram à seleção deste tipo de código foram:

- Velocidade e baixo uso de processamento;
- Independência de quais serão os dados codificados recebidos;
- Garantia do recebimento das informações de maneira correta;
- Não é necessário confirmação do recebimento dos dados;
- Não necessita ordenação;
- Destino necessita apenas cerca de 5% de informação redundante para decodificar a informação original (MACKAY, 2003).

Para evitar equívocos serão definidos alguns termos que serão utilizados no decorrer deste documento. Quando o texto se referir a dados, este representa as informações originais a serem enviadas do nó origem ao destino, o termo dados codificados se refere às informações originais já codificadas, ou seja, aos fragmentos de dados após sofrerem a codificação que

[1]Nesta tese o conceito de confiabilidade refere-se a uma confiabilidade estatística de entrega das mensagens, para isso serão utilizadas as médias com intervalos de confiança de 95% para mais e para menos.

será explicada a seguir e por fim Pacote será o termo utilizado para os dados codificados já encapsulados, i.e., os dados codificados em conjunto com os dados do cabeçalho do PTTA - Protocolo de Transporte Tolerante a Atrasos.

A seguir será descrito o formalismo pertinente ao protocolo de transporte proposto.

Em primeiro lugar, como este protocolo é um protocolo da camada de transporte, ele deve funcionar como multiplexador e demultiplexador para aplicações. Por isso é necessário o uso das portas da fonte e destino, para saber a qual aplicação estes dados são destinados e de qual aplicação eles estão sendo originados. Estes campos estão presentes nos cabeçalhos do TCP e do UDP e foram mantidos.

Para conseguir uma certa garantia e confiabilidade estatística na informação recebida é necessário um campo *checksum* que possui as informações para a verificação da integridade dos dados do pacote. Este campo está presente nos cabeçalhos UDP e TCP também.

Como deseja-se anexar o uso dos códigos fontanais, no protocolo, a fim de melhorar a taxa de entrega dos dados, algumas informações sobre o conteúdo do pacote, devem ser adicionada ao cabeçalho para se realizar a decodificação das informações. Para isso é necessário conhecer quais são as partes dos dados originais envolvidas em cada uma das gotas. Portanto, cada gota deve carregar consigo alguma forma de identificação dessas partes. Assim, foi criado o campo Mapa de Partes. Este campo conta com 32 bits, e desta forma, pode-se representar 32 partes que podem estar envolvidas na gota atual. Porém, podem existir situações onde os dados originais tenham que ser divididos em um número maior que 32 partes. Como podem existir várias partes e o campo reservado no cabeçalho possui um tamanho fixo, pode ocorrer que número de bits disponível para representa-las (32 bits no campo Mapa de Partes) seja muito pequeno, o que ocasionaria conflito e impossibilitaria a recuperação dos dados. Dessa forma, para evitar este inconveniente, foi necessário aumentar o número de bits disponíveis para essa informação (Mapa de Partes). Devido a isto, foi criado o campo Tamanho do Mapa de Partes (com 16 bits) que representa quantos grupos de 32 bits serão utilizados no campo Mapa de Partes, o que aumenta significativamente a representação para se gerar as gotas, pois agora os dados podem ser divididos em $2^{16} \times 32$ partes. Além disso, alguns bits desse campo podem ser futuramente separados conforme as necessidades que possivelmente venham surgir.

Finalmente, o campo Dados possui realmente a gota que está sendo carregada da origem para o destino.

A Figura 7 mostra todos esses campos que formam o cabeçalho do protocolo desenvolvido.

Porta da Fonte	Porta do Destino
Checksum	Tamanho do mapa de partes
Mapa de partes	
Dados	
16 bits	16bits

Figura 7: Esquema com os campos do cabeçalho do protocolo de transporte apresentado.

Porta da Fonte: (16 bits) Este campo contém o número da porta de origem.

Porta do Destino: (16 bits) Este campo contém o número da porta do destino.

Checksum: (16 bits) Este campo contém o complemento de um da soma de todas as palavras de 16 bits pertinentes ao cabeçalho e ao campo Dados. Se uma gota tiver um número ímpar de octetos, o último deverá ser completado com zeros, sendo que estes zeros não serão transmitidos no campo dos dados.

Tamanho do Mapa de Partes: (16 bits) Este campo possui um valor inteiro.

Mapa de Partes: (Múltiplo de 32 bits) Este campo possui valor 1 na posição relativa a cada parte do dado original utilizado na geração da gota, e o valor 0 nas partes não utilizadas.

Dados: (Variável) Este campo contém a gota a ser transmitida.

3.2 FUNCIONAMENTO DO PROTOCOLO

Nesta seção serão apresentadas as diversas formas e aplicações para o protocolo especificado. Como as redes podem ser utilizadas para uma infinidade de aplicações e, cada aplicação possui as suas próprias exigências e tolerância a atrasos e falhas na entrega dos dados, o protocolo a ser utilizado deve ser versátil o suficiente para que possa ser aplicado à gama toda de aplicações disponíveis no mercado. Para essa versatilidade serão citadas algumas implementações do protocolo e aplicações de cada uma dessas implementações. A característica que diferencia cada uma das abordagens é o controle de como a fonte (nó origem) irá gerar as mensagens e encaminhá-las.

FONTE SEMPRE ABERTA: Nesta abordagem, o nó de origem permanece enviando as informações codificadas por um longo período de tempo. Sua aplicabilidade torna-se interessante para atualização de *firmware*, ou em qualquer outro cenário onde exista apenas um transmissor e esse deve ter a certeza de que todos os receptores receberam e conseguiram decodificar a informação corretamente. Geralmente, este é um cenário, onde o administrador da rede

não está preocupado com o gasto de recursos ou com o tráfego da rede e o uso dos buffers, seria melhor aplicado em dispositivos não alimentados por baterias, ou que apenas os nós de destino (objetivo da transmissão) fossem alimentados desta maneira, pois assim não teria o problema da escassez de recursos. Os prós desse modo são a taxa de entrega das mensagens, o tempo de entrega e a confiabilidade estatística, pois como são enviadas muitas mensagens codificadas, a probabilidade de sucesso na decodificação dos dados aumenta (conf. figura 5). Dentre os pontos contra estão a saturação dos buffers dos nós intermediários, a carga da rede e o gasto de recursos.

FONTE COM ABERTURA PRÉ-DEFINIDA (POR PERÍODO ou POR QUANTIDADE DE RÉPLICAS): Nesta implementação o nó origem possui um período pré-estabelecido para realizar o envio das mensagens ou um número de réplicas de cada mensagem que será enviada. O seu comportamento se dá com uma jorrada de mensagens codificadas. Sua aplicação é interessante para o caso onde deseja-se a transferência da informação de maneira estatisticamente garantida, mas que deseja-se que outras aplicações tenham a possibilidade de utilizar os recursos da rede. Além disso, pode ser empregada em redes previsíveis, onde a taxa de perda de pacotes seja conhecida, podendo assim configurar o valor da quantia de cópias que se fazem necessário o envio. Um dos exemplos é quando se está transmitido um vídeo para várias pessoas, atualização de aplicativos ou algum tipo de informação onde pode-se admitir algum atraso maior e uma confiabilidade estatística média. Outro exemplo são pessoas num estádio de futebol que estão recebendo a narração ou imagens ou ainda os melhores momentos de outro jogo que esteja acontecendo simultaneamente, ou até mesmo do próprio jogo que está sendo assistido no estádio. Os prós desse método são a possibilidade de uso da rede para várias aplicações, a não saturação dos buffers dos nós intermediários (dependente do tempo definido para a abertura ou da quantidade de réplicas a serem enviadas) e uma alta taxa de entrega das informações (conf. figura 5). Dentre os contras estão a carga na rede que, se não for bem manipulada a abertura da fonte, os buffers dos relays acabam sendo saturados. O gasto de recursos é ainda dependente dessa definição da abertura da fonte, porém este quesito não pode ser definido como um ponto a favor e nem contra, pois caso a abertura da fonte seja definida para um tempo relativamente pequeno, esse ponto é uma vantagem, por outro lado, se esse tempo for muito grande, isso torna-se uma desvantagem. Esta é a implementação mais versátil do protocolo, onde a aplicação teria o maior controle sobre ele, podendo manipular seu comportamento de acordo com as necessidades de cada aplicação. Ideal para redes sobre uma gerência centralizada que pode facilmente agendar o tempo das disseminações dos objetos para maximizar o uso da rede.

PTTA COM CONTROLE DA FONTE E MANTENDO O USO DOS RECURSOS: Nesta forma de operação, o protocolo de transporte controla a geração de mensagens baseado

no uso que se deseja da rede. É o tipo de uso do protocolo que seria convencional para as aplicações que encontra-se nos dias atuais em smartphones, tablets, notebooks e netbooks, pois seria utilizado em casos onde tem-se diversos dispositivos interconectados na mesma rede e todos compartilhando seus recursos sem uma gerência centralizada dos recursos. Diversas aplicações estão sendo executadas em diversos dispositivos distintos, isso cria um ambiente onde a economia dos recursos da rede é de suma importância, pois deseja-se que as baterias dos dispositivos móveis durem o maior tempo possível, que os buffers sejam sempre utilizados de maneira otimizada e que a comunicação ocorra com uma certa confiabilidade estatística. Esta abordagem é onde consegue-se um equilíbrio entre taxa de entrega e uso dos recursos da rede, é muito útil para uma rede onde se conhece a taxa de entrega das mensagens e sabe-se o comportamento da rede, pois o uso dos recursos seria controlado pela aplicação. Os prós dessa forma de operação são uma relação entre uso de recursos da rede e taxa de entrega muito boa, o que leva à economia de recursos da rede, uma boa taxa de entrega dos dados (conf. figura 5) e ainda a possibilidade do uso da rede para diversas aplicações simultâneas. Dos pontos contra pode-se citar apenas que a informação não é entregue no menor tempo possível, porém isso para várias aplicações em DTN não é um problema.

PTTA COM CONTROLE DA FONTE POR MODELO DE ENCONTROS: É o tipo de aplicação mais automatizado e autônomo possível. Independente do modelo de encontros utilizado, o controle da fonte é feito automaticamente, podendo-se utilizar os parâmetros mais variados para o ajuste da diversidade de informação que será transmitida pela fonte. Com esta possibilidade, o uso do protocolo vem a ganhar conforme este modelo de encontros é melhorado. Isso é de extrema importância pois torna o protocolo totalmente autônomo. Pode-se utilizar um modelo baseado na mobilidade dos nós que será descrito mais adiante na seção 3.3.1. Pode-se utilizar um modelo da média dos encontros do nó fonte com os seus vizinhos. Ou, a média de encontros que os vizinhos do nó fonte possuem. Ou ainda, pode-se fazer uma média com todos os encontros que todos os nós que o nó fonte encontrar, fazendo assim a estimativa da média de encontros da rede, por exemplo. Além disso, caso surjam trabalhos que melhor descrevam uma rede, pode-se facilmente acoplar esse novo modelo ao protocolo e obter resultados ainda melhores, o que torna o protocolo uma ferramenta totalmente versátil e atualizada (atualizável). Os prós dessa forma residem no fato de ser um controle automático se adaptando à rede em tempo real, sem a necessidade de intervenção por parte da aplicação. Além disso, caso seja implementado um modelo característico o protocolo passa a se comportar conforme este modelo, daí a sua versatilidade. Um ponto negativo dessa forma de operação é que dependendo do modelo utilizado, caso seja utilizado um modelo errado, o comportamento do protocolo não será congruente com a rede onde está aplicado, pois o comportamento dele

depende do modelo aplicado, porém isso seria caracterizado como uma falha do administrador e não do protocolo.

Nas diversas abordagens o funcionamento do protocolo se dará da mesma forma à explicada no algoritmo a seguir, porém o critério de parada "<CONDIÇÃO>" do laço enquanto é o que será alterado em cada uma das abordagens.

Algoritmo 1: Fonte

```
Entrada: Dados
Saída: PacoteCodificado
início
    DivideDadosOriginaisEmNPartes(Dados);
    //Divide os dados em N partes e numera estas
    partes
    enquanto <CONDIÇÃO> faça
        SorteiaGrauDoPacote();
        //Gera X aleatoriamente
        SorteiaPartesEnvolvidas();
        //Seleciona X das N partes aleatoriamente
        RealizaCodificação();
        //Realiza a operação XOR entre as X partes
        selecionadas
        Pacote := MontaPacote();
        //Monta o pacote com PortaDeOrigem, PortaDeDestino,
        Checksum (PREENCHIDO COM 0s), TamMapaPartes,
        MapaDePartes, DadosCodificados
        Checksum := CalculaChecksum(Pacote);
        //Calcula a soma em complemento 1 do pacote de
        16 em 16 bits (2 em 2 bytes)
        Pacote := MontaPacote();
        //Monta o pacote com PortaDeOrigem, PortaDeDestino,
        Checksum (calculado anteriormente),
        TamMapaPartes, MapaDePartes, DadosCodificados
        EnviaParaCamadaInferior(Pacote);
        //Repassa o Pacote para a camada inferior
    fim
fim
```

A Figura 8 apresenta um exemplo do processo descrito no algoritmo anterior. Nesta figura é possível visualizar que os dados foram divididos em *n* partes. Estas partes foram combinadas para formar as "gotas" que representam o resultado da operação *XOR* entre os bits das partes combinadas. Esse resultado é utilizado no campo dados do pacote. Na figura ainda observa-se que existe uma quantidade muito grande de combinações entre as partes para a geração das gotas.

Para montar o pacote é incluído o valor da porta de origem e destino, o *checksum*, o tamanho do mapa de partes, no exemplo, este campo possui o valor 1 pois, o mapa de partes é composto por apenas um conjunto de 32 bits. Por fim, o mapa de partes é montado sendo este composto pelo valor 1 nos bits que representam as partes utilizadas para a geração da gota que será utilizada neste pacote e o campo dados é composto pelo resultado da operação *XOR* entre as partes envolvidas nesta gota (que no exemplo seriam as partes 1 e 2).

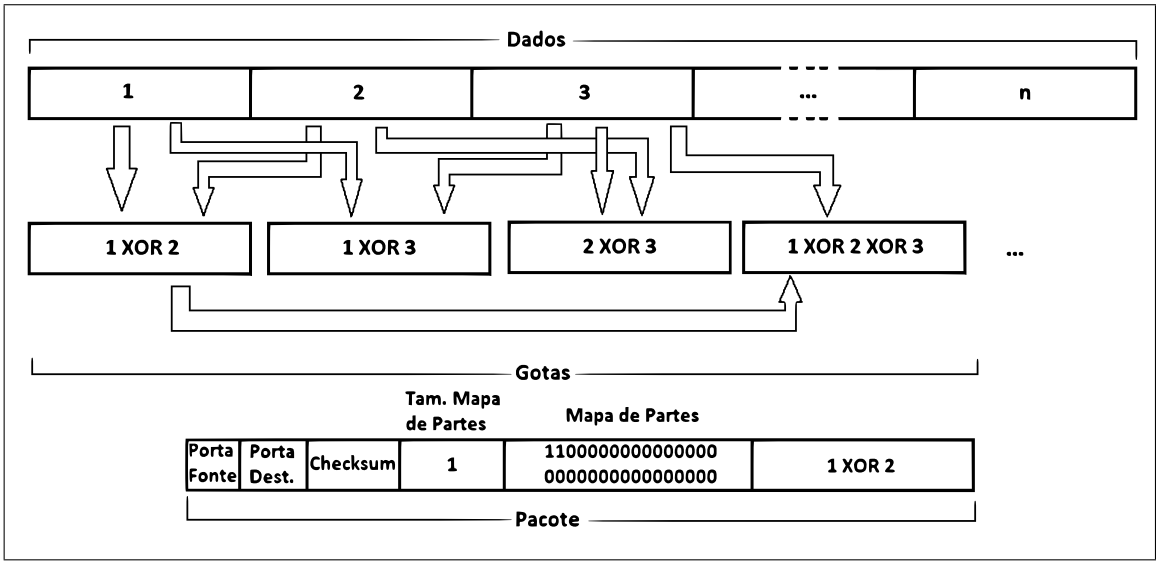

Figura 8: Ilustração do funcionamento do PTTA.

O protocolo de transporte do lado do destino (receptor) possui o seu funcionamento, para todas as abordagens, de acordo com o algoritmo descrito abaixo:

Algoritmo 2: Receptor

```
Entrada: PacotesCodificados
Saída: Dados
início
    K := 0;
    //Contador de pacotes decodificados
    enquanto K < N faça
        para NumPacote := 1, NumPacote<TamanhoDe(Pacotes) passo 1
        faça
            Pacote := Pacotes[NumPacote];
            t := VerificadorMapaDePartes(Pacote);
            //Verifica se Pacote possui apenas um valor 1
            no MapaDePartes e retorna verdadeiro ou falso
            se t então
                Pos := PosDados(Pacote);
                //Retorna a posição do valor 1 do
                MapaDePartes de Pacote
                AtribuiValor(Pos, Pacote, DadosOriginais);
                //Atribui o valor do campo DadosCodificados
                de Pacote para DadosOriginais[Pos]
                K := K + 1;
                para i := 1, i<TamanhoDe(Pacotes) passo 1 faça
                    DecodificaPacotes(DadosOriginais[Pos], Pacotes[i], Pos);
                    //Realiza a operação XOR do valor
                    DadosOriginais(Pos) com o campo
                    DadosCodificados de Pacotes[i] se este
                    possuir o valor 1 na posição Pos do
                    MapaDePartes e altera o valor do bit
                    MapaDePartes[Pos] para 0
                fim
            fim
        fim
    fim
fim
```

3.3 MECANISMO DE CONTROLE DA FONTE

Com o intuito de prever a mobilidade e comportamento de repasse das mensagens em uma DTN, foi realizado em conjunto com o CREATE-NET(*Center for REsearch And Telecommunication Experimentation for NETworked Communities*) um estudo sobra a distribuição da informação em uma rede tolerante a atrasos.

3.3.1 MODELIZAÇÃO MATEMÁTICA

Para tornar possível o desenvolvimento da modelização matemática foram realizadas duas considerações: os nós se movem de acordo com o modelo de mobilidade Random Way Point (JOHNSON; MALTZ, 1996) e; utilizam o protocolo de roteamento chamado Two-hops routing (GROSSGLAUSER; TSE, 2002) (GROSSGLAUSER; TSE, 2001). Estas características são necessárias, por questão de tornar possível a análise do problema. O Random Way Point é um padrão de movimentação aleatório onde é selecionada uma posição ao acaso e o nó caminha em direção a ela com uma velocidade também selecionada aleatoriamente. O protocolo Two-hops se preocupa em repassar a mensagem da origem para um nó intermediário e este repassa a mensagem somente ao destino, o que elimina a recursividade existente em outros protocolos a fim de facilitar o desenvolvimento do modelo matemático. Além disso, utilizando-se do padrão de mobilidade Random Way Point, a análise do padrão de encontros dos nós pode ser descrita levando-se em conta a densidade de "entre encontros" (*Intermeeting Intensity* - II), podendo assim ser modelada por um processo de Poisson.

Nas redes tolerantes ao atraso, a falta de conectividade persistente sugere o uso de redundância para atingir os nós de destino. Uma técnica é armazenar informação nos nós intermediários, chamados de relays, na forma de cópias adicionais da mensagem.

Para a realização do desenvolvimento dos modelos matemáticos foi utilizado o princípio da teoria de filas conforme foi proposto por (HAYDEN et al., 2012) e (BORTOLUSSI; HILLSTON, 2012).

Considera-se uma DTN composta de um nó origem, N-1 nós intermediários e um nó destino (perfazendo N nós, no início do cenário, que não possuem mensagem alguma). Uma mensagem é gerada no nó origem no tempo t=0 e precisa ser repassada para o nó de destino. Neste contexto uma forma simples de modelar o processo de entre encontros é através de um processo puntual onde as chegadas modelam os encontros intermediários que regimentam as oportunidades de transmissão entre os nós DTN. Neste modelo será adotado um processo de Poisson de tal forma que o tempo entre contatos de dois nós é assumido que seja exponencial-

mente distribuído com intensidade $\lambda > 0$.

Para modelar uma DTN deve-se focar no contato entre os nós. No caso do roteamento two-hops, em cada contato entre origem e um nó que ainda não tenha recebido a mensagem, a mensagem é confiada a este nó. Se um nó que não é o origem possui a mensagem e entra em contato com outro nó então este repassará a mensagem se e somente se esse outro nó for o destino. Todavia, este raciocínio pode ser refinado incluindo no modelo a probabilidade de transferência de uma mensagem durante um contato, o que relaciona a distribuição dos tempos de contato.

O tempo de contato entre dois nós móveis é o período de tempo onde esses dois nós possuem a oportunidade de se comunicar, uma vez que eles se encontram no alcance de rádio um do outro. Mesmo existindo uma variedade grande de trabalhos nesta área, a grande maioria das pesquisas assume que o tempo de contato entre os nós é suficiente para trocar a mensagem em cada encontro.

Um olhar mais detalhado sugere que a distribuição do tempo de contato deva ter um impacto no atraso e na capacidade da rede. Em particular, uma vez que o tempo de contato entre dois nós algumas vezes não seja suficiente para transferir uma cópia da mensagem para um nó intermediário, o nó origem deveria dividir a mensagem em K pacotes. Aqui será descrito em detalhes como a fragmentação melhora a eficiência de uma DTN ligando a probabilidade de entrega de uma certa mensagem e o número de pacotes K que o origem precisa dividir a mensagem.

Para proceder com a análise, as aproximações de fluído, que serão apresentadas a seguir, onde se modela a evolução do valor médio de um processo subjacente podem ser encontradas em (HAYDEN et al., 2012) e (BORTOLUSSI; HILLSTON, 2012).

3.3.1.1 TRANSMISSÃO SEQUENCIAL

Na transmissão sequencial a origem entrega pacotes em ordem para os relays, *i.e.*, os pacotes são numerados 1, 2, ..., K, e ao encontrar um nó que possua $j-1$ pacotes, $0 \leq j \leq K$, o origem tentará entregar pacotes a partir de j. Seja $X_i(t)$ a fração dos nós excluindo o origem que possuem os i primeiros pacotes no tempo t. Se no tempo t o origem encontrar um nó que possui os i primeiros pacotes, então esse receberá j pacotes $i+1, ..., i+j$ se o tempo de contato for $t_j \leq T_c < t_{j+1}$; onde t_j denota o tempo necessário para transmitir j pacotes. Desta forma, a aproximação de fluído para esse processo pode ser escrita como

$$\begin{aligned}
\dot{X}_0(t) &= -\lambda \overline{p}_1 X_0(t) \\
\dot{X}_i(t) &= \lambda \sum_{j=0}^{i-1} X_j(t) p_{i-j} - \lambda \overline{p}_1 X_i(t) \;\; i = 1,..,K-1 \\
\dot{X}_K(t) &= -\sum_{j=0}^{K-1} \dot{X}_j(t)
\end{aligned} \tag{1}$$

onde $p_j = \mathbb{P}\{t_j \leq T_c < t_{j+1}\}$ e $\bar{p}_j = \mathbb{P}\{T_c \geq t_j\}$.

Em particular, a primeira equação em (1) possui um argumento $\lambda \overline{p}_1$ que é a frequência na qual um relay que não tem nenhum pacote se encontra com a origem e o contato dura o suficiente para ter ao menos um pacote transferido para o relay. As equações para $i = 1, \ldots, K-1$ seguem usando um argumento similar e a última expressa a dinâmica do estado de absorção K.

A dinâmica correspondente a (1) pode ser derivada como segue. Denota-se $X_0(t)$ a fração de nós que não possuem pacotes no tempo t. Desta forma tem-se

$$X_0(t) = e^{-\lambda \overline{p}_1 t} \mathbb{1}_{\{t \geq 0\}}(t) \tag{3}$$

onde $\mathbb{1}_A(\cdot)$ é um padrão indicando um conjunto de funções A; o sistema (1) pode ser resolvido iterativamente usando a transformada de Laplace $X_i(s) = \mathscr{L}[X_i(t)|s]$,

$$sX_i(s) = \lambda \sum_{j=0}^{i-1} X_j(s) p_{i-j} - \lambda \overline{p}_1 X_i(s) \;\; i = 1,..,K-1$$

notando que $X_0(s) = (s + \lambda \overline{p}_1)^{-1}$.

Usando a transformada de Laplace, tem-se

$$
\begin{aligned}
X_1 &= \lambda p_1 X_0^2 \\
X_2 &= \lambda p_2 X_0^2 + \lambda^2 p_1^2 X_0^3 \\
X_3 &= \lambda p_3 X_0^2 + 2\lambda^2 p_1 p_2 X_0^3 + \lambda^3 p_1^3 X_0^4 \\
X_4 &= \lambda p_4 X_0^2 + \lambda^2 (2p_1 p_3 + p_2^2) X_0^3 \\
&\quad + \lambda^3 3 p_1^2 p_2 X_0^4 + \lambda^4 p_1^4 X_0^5 \\
X_5 &= \lambda p_5 X_0^2 + 2\lambda^2 (p_1 p_4 + p_2 p_3) X_0^3 \\
&\quad + 3\lambda^3 (p_1^2 p_3 + p_1 p_2^2) X_0^4 + 4\lambda^4 p_1^3 p_2 X_0^5 + \lambda^5 p_1^5 X_0^6 \\
X_6 &= \lambda p_6 X_0^2 + \lambda^2 (2p_1 p_5 + 2p_2 p_4 + p_3^2) X_0^3 + \lambda^3 (3p_1^2 p_4 \\
&\quad + 6p_1 p_2 p_3 + p_2^3) X_0^4 + \lambda^4 (4p_1^3 p_3 + 6p_1^2 p_2^2) X_0^5 \\
&\quad + 5\lambda^5 p_1^4 p_2 X_0^6 + \lambda^6 p_1^6 X_0^7 \\
&\quad \cdots
\end{aligned}
\tag{4}
$$

o que leva à expressão geral

$$
X_i(s) = \sum_{j=0}^{i-1} X_0(s)^{j+2} \lambda^{j+1} a_{i,i-j}(p_1, \ldots, p_{i-j}) \tag{6}
$$

Desta forma, é possível derivar de (4) o cálculo iterativo no domínio do tempo ao tomar a transformada inversa como sendo

$$
X_i(t) = \sum_{j=0}^{i-1} a_{i,i-j}(p_1, \ldots, p_{i-j}) \lambda^{j+1} \Theta_{j+2}
$$

Aqui a descrição combinatória da entrega sequencial dos pacotes é codificada pelos coeficientes $a_{i,i-j}$: esses são monômios de grau j dos p_is, onde

$$
a_{i,i-j}(p_1, \ldots, p_{i-j}) = \sum_{\{r : \sum r = i-j\}} \prod p_r \tag{7}
$$

e $a_{i,i-j}$ é a probabilidade de se repassar i pacotes em $i-j$ contatos por exemplo, o coeficiente $a_{6,3} = (3p_1^2 p_4 + 6p_1 p_2 p_3 + p_2^3)$ descreve a probabilidade de transferir um pacote em dois contatos e quatro pacotes em um contato, mais a probabilidade de transferir um, dois e três pacotes em cada contato respectivamente; e o último termo da soma corresponde à probabilidade de transferir exatamente dois pacotes por contato.

Além disso, Θ_i é a i-ésima auto-convolução de $X_0(t)$, i.e., $\theta_0(t) = X_0(t)$ e $\theta_{j+1}(t) = \theta_j * X_0(t)$ e isso leva também a $\theta_j(t) = \frac{t^{j-1}}{(j-1)!}X_0(t)$.

Para se entender os coeficientes $a_{i,i-j}$ é possível remeter ao conceito de combinação de *partições* e *partições ordenadas* de um inteiro n em k partes. Em geral, a partição de n em k partes é a solução de $n = x_1 + \ldots + x_k$, para $x_1 \geq x_2 \geq \ldots \geq 1$. Por exemplo, $(3,2,1)$ é uma partição ordenada de $n = 6$ em 3 partes. Para qualquer solução dada é possível determinar o número correspondente de partições não ordenadas como $k!/\sum_{r=1}^{n} h_r!$, onde h_r é o número de ocorrências do inteiro r nas partições.

Retornando em (6), é necessário expressar a estrutura combinatorial da entrega das mensagens uma vez que esta é fragmentada. Neste caso, os coeficientes $a_{i,i-j}$ correspondem às *partições ordenadas* do inteiro i em $i-j$ partes, uma vez identificado p_i com o inteiro i, i.e., o número de pacotes. Por exemplo, para $i = 6$, as partições em 2 partes são 2 do tipo $\{1,5\}$, 2 do tipo $\{2,4\}$ e apenas uma do tipo $\{3,3\}$.

Seja $v(n,k)$ o número de partições ordenadas n em k partes e $p(n,k)$ o número de partições de n em k partes. Na Tabela 1 estão enumeradas as partições e as partições ordenadas até $n = 4$. Na literatura podem ser encontrados diversos algoritmos para determinar as partições de um inteiro em k partes (ZOGHBI; STOJMENOVIĆ, 1998).

n	k	p(n)	partição	partição ordenada	$v(n,k)$
1	1	1	(1)	(1)	1
2	1	2	(2)	(2)	1
	2		(1,1)	(1,1)	1
3	1	4	(3)	(3)	1
	2		(2,1)	(2,1) (1,2)	2
	3		(1,1,1)	(1,1,1)	1
4	1	5	(4)	(4)	1
	2		(3,1)	(3,1)(1,3)	3
			(2,2)	(2,2)	
	3		(2,1,1)	(2,1,1)(1,2,1)(1,1,2)	3
	4		(1,1,1,1)	(1,1,1,1)	1

Tabela 1: O número de partições e o valor $v(n,k)$ para pequenos valores de k.

O número de termos que aparece no polinômio $a_{i,i-j}$, no caso geral cresce de acordo com o número de partições; os coeficientes na frente de cada monômio nos p_is é o número de partições ordenadas. Através da observação, é possível construir um algoritmo para montar a expressão de (6) por exemplo, $a_{5,3}$ diz respeito a partições de 5 em 3 partes, e corresponde às seguintes partições não ordenadas $(3,1,1)$ e $(2,2,1)$. Usando a correspondência acima, os dois monômios $p_1^2 p_3$ e $p_1 p_2^2$ irão aparecer em $X_5(s)$, ambos com coeficientes $3 = 3!/2!$.

Desta forma para prosseguir com os cálculos, será levada em conta esta relação do número de partições

Lema 3.3.1. *Sendo $n \geq k \geq 1$, então $v(n,k) = \binom{n-1}{k-1}$*

Demonstração. Por indução em n. O fato de ser verdade para $n = 2$. Agora, assume-se que também será para $n-1$, e para todo $k \leq n-1$, então

$$v(n,k) = \sum_{h=1}^{n-k+1} v(n-h,k-1) = \sum_{h=1}^{n-k+1} \binom{(n-2)-(h-1)}{k-2}$$
$$= \sum_{h=0}^{n-k} \binom{(n-2)-h}{k-2} = \sum_{r=k-2}^{n-2} \binom{(n-2)-r}{r} = \binom{n-1}{k-1}$$

onde o último passo provém da identidade binomial $\sum_{r=M}^{N} \binom{r}{M} = \binom{N+1}{M+1}$. □

Nota-se que do Lema 3.3.1, o número de monômios distintos que são levados em conta nos termos à direita de (6) são 2^{i-1}.

Finalmente, uma expressão simplificada para (4) pode ser obtida no caso de uma distribuição geométrica, i.e., quando $p_i = (1-p)p^i$, $i = 1,2,\ldots$, para $0 \leq p \leq 1$. Neste caso, para escrever a dinâmica, é suficiente usar o número das partições ordenadas em (6) e observar que a expressão será

$$X_i(s) = p^i \sum_{j=0}^{i-1} v(i,i-j)[\lambda(1-p)]^{j+1} X_0(s)^{j+2} \tag{9}$$

A partir de (9), no domínio de tempo a expressão para $i = 1,\ldots,K-1$:

$$X_i(t) = p^i X_0(t) \sum_{j=0}^{i-1} v(i,i-j) \frac{[\lambda(1-p)]^{j+1} t^{j+1}}{(j+1)!}$$
$$= p^i X_0(t) \sum_{j=0}^{i-1} \binom{i-1}{j} \frac{[\lambda(1-p)t]^{j+1}}{(j+1)!} = p^i X_0(t) \widetilde{X}_i(t) \tag{10}$$

Em seguida, serão explicitadas as expressões para o caso de um número de pacotes pequeno.

$$
\begin{aligned}
\widetilde{X}_1 &= \lambda(1-p)t \\
\widetilde{X}_2 &= \left(\lambda(1-p)t + \frac{[\lambda(1-p)]^2t^2}{2}\right) \\
\widetilde{X}_3 &= \left(\lambda(1-p)t + 2\frac{[\lambda(1-p)]^2t^2}{2} + \frac{[\lambda(1-p)]^3t^3}{6}\right) \\
\widetilde{X}_4 &= \left(\lambda(1-p)t + 3\frac{[\lambda(1-p)]^2t^2}{2} + 3\frac{[\lambda(1-p)]^3t^3}{6}\right. \\
&\quad \left. + \frac{[\lambda(1-p)]^4t^4}{24}\right) \\
\widetilde{X}_5 &= \left(\lambda(1-p)t + 4\frac{[\lambda(1-p)]^2t^2}{2} + 6\frac{[\lambda(1-p)]^3t^3}{6} + \right. \\
&\quad \left. 4\frac{[\lambda(1-p)]^4t^4}{24} + \frac{[\lambda(1-p)]^5t^5}{120}\right)
\end{aligned}
\tag{12}
$$

Nesta derivação é relevante ter uma expressão na forma fechada para a dinâmica da fração dos nós que possuem todos os K pacotes, chamados, X_K: combinando as expressões anteriores é possível derivá-la

$$
\begin{aligned}
X_K(t) &= 1 - X_0(t)\left[1 + \sum_{r=1}^{K-1} p^r \sum_{j=1}^{r} \binom{r-1}{j-1} \frac{[\lambda(1-p)t]^j}{j!}\right] \\
&= 1 - X_0(t)\left[1 + \sum_{j=1}^{K-1} \frac{[\lambda(1-p)t]^j}{j!} \sum_{r=j}^{K-1} p^r \binom{r-1}{j-1}\right] \\
&= 1 - X_0(t)[1 + F_K(p)]
\end{aligned}
\tag{14}
$$

onde foi utilizado novamente o fato que $\sum_{l=k}^{n} \binom{l}{k} = \binom{n+1}{k+1}$ e definido

$$
F_K(p) = \sum_{j=1}^{K-1} \frac{[\lambda(1-p)t]^j}{j!} \binom{K-1}{j} \tag{16}
$$

3.3.1.2 PROBABILIDADE DE SUCESSO

Neste momento o interesse é na probabilidade de sucesso da mensagem completa, i.e., a probabilidade na qual a transmissão sequencial pode entregar todos os K pacotes para o destino dado um deadline. Seja T_K o atraso da entrega da mensagem, i.e, o tempo desde o primeiro pacote ser gerado pela fonte, até o último pacote K ser entregue ao destino. A distribuição

(CDF) do atraso T_K é denotado por $D_X(t) = \mathbb{P}(T_K \leq t)$. Seja $y_d(t)$ uma variável aleatória que representa o número de pacotes recebidos pelo destino no intervalo $[0,t]$. É possível expressar a probabilidade de sucesso no recebimento da mensagem em t como

$$\begin{aligned}
&D_X(t+dt) - D_X(t) = \mathbb{P}(t < T_K \leq t+dt) \\
&= \sum_{i=0}^{K-1} \mathbb{P}(t < T_K \leq t+dt, y_d(t) = i) \\
&= \sum_{i=0}^{K-1} \mathbb{P}(T_K \leq t+dt | y_d(t) = i, T_K > t) \cdot \\
&\quad \cdot \mathbb{P}(y_d(t) = i | T_K > t) \mathbb{P}(T_K > t) \\
&= \sum_{i=0}^{K-1} dt\lambda(1 + NX_K(t))\bar{p}_{K-i} \frac{NX_i(t)}{N - NX_K(t)} (1 - D_X(t)) \\
&= dt\lambda N\Big(\frac{1}{N} + X_K(t)\Big) \sum_{i=0}^{K-1} \frac{X_i(t)\bar{p}_{K-i}}{1 - X_K(t)} (1 - D_X(t))
\end{aligned}$$

Nos cálculos anteriores, $\mathbb{P}(T_c \leq t+dt | y_d(t) = i, T_K > t)$ representa a probabilidade do destino receber todos os pacotes condicionado com o fato dele já possuir i pacotes no tempo t. Isso ocorre se o destino encontra um nó que tenha todos os pacotes e o contato dure o suficiente para repassar $K - i$ pacotes. A probabilidade que o destino tenha i pacotes no tempo t é representada pelo termo $\mathbb{P}(y_d(t) = i | T_K > t)$; então $\mathbb{P}(y_d(t) = i | T_K > t)$ é a probabilidade do destino estar entre os nós que possuem i pacotes no tempo t (deve-se lembrar que existem N nós).

Assim é possível obter um conjunto de equações diferenciais (ODE) separáveis

$$\frac{dD_X(t)}{dt} = \lambda N\Big(\frac{1}{N} + X_K(t)\Big) \sum_{i=0}^{K-1} \frac{X_i(t)\bar{p}_{K-i}}{1 - X_K(t)} [1 - D_X(t)] \tag{18}$$

estas equações podem ser integradas da forma

$$\int_0^t \frac{dD}{1-D} = \lambda N \int_0^t \left[\frac{1}{N} + X_K(s)\right] \sum_{i=0}^{K-1} \frac{X_i(s)\bar{p}_{K-i}}{1 - X_K(s)} ds$$

Finalmente, é possível extrair a expressão na forma fechada de $D_X(t)$ integrando por partes e notando que $\dot{X}_K(s) = \lambda \sum_{i=0}^{K-1} X_i(s)\bar{p}_{K-i}$ para rearranjar os cálculos seguintes

$$
\begin{aligned}
D_X(t) &= 1-\exp\left\{-\lambda N\sum_{i=0}^{K-1}\int_0^t ds\left[\frac{1}{N}+X_K(s)\right]\frac{X_i(s)\bar{p}_{K-i}}{1-X_K(s)}\right\}\\
&= 1-\exp\left\{-N\int_0^t \frac{\left[\frac{1}{N}+X_K(s)\right]\dot{X}_K(s)}{1-X_K(s)}ds\right\}\\
&= 1-\exp\left\{N\big(X_K(t)-X_K(0)\big)+(N+1)\log\left(\frac{1-X_K(t)}{1-X_K(0)}\right)\right\}
\end{aligned}
$$

Onde foi utilizado $\int \frac{x}{1-x}dx = -x-\log(1-x)$.

O resultado principal da análise anterior é que utilizando um protocolo de entrega sequencial, a probabilidade de sucesso $D_X(t)$ em qualquer instante de tempo, não depende da trajetória de cada pacote, i.e., nos X_is. Mas, depende apenas do estado final de $X_K(t)$. Desta última observação é possível verificar, com um cálculo simples, que a expressão acima é crescente na variável X_K. Desta forma, é possível afirmar

Lema 3.3.2. *$D_X(t)$ é uma função apenas de $X_K(t)$ e é crescente em X_K.*

Como uma consequência do Lema 3.3.2, quando deseja-se caracterizar a probabilidade de sucesso como uma função da distribuição do número de pacotes que podem ser transmitidos durante um contato, pela propriedade da monotonicidade é possível focar no efeito de tal distribuição na variável X_K.

3.3.1.3 FRAGMENTAÇÃO

A seguir será desenvolvida a análise de X_K como uma função de K. Ao olhar para (14), deveria estar sendo estudado o impacto da distribuição dos pacotes transmitidos por contato na dinâmica de X_K. Porém, desde que se assume a distribuição geométrica, é preciso estudar o impacto do parâmetro p, i.e., a probabilidade de sucesso na transmissão de um pacote em um contato; será escrito $p = p_K$ para relembrar que esta probabilidade é uma função do número de fragmentos gerados a partir da mensagem original e será assumido que:

Suposição 3.3.1. *p_K não é decrescente com K: $p_{K+1} \geq p_K$ para $K = 1,2,\ldots$*

Óbvio, a probabilidade de entrega para um pacote menor em um contato apenas não pode ser menor do que aquela para entregar um pacote maior neste contato, o número de pacotes maior para serem entregues ao destino acaba se tornando uma penalidade. No caso extremo quando p_K é uma constante, a probabilidade de entrega da mensagem é decrescida quando

comparada ao caso quando a mensagem não é fragmentada. Todavia, esse é um caso trivial quando não existe benefício em fragmentar a mensagem.

Geralmente, a suposição feita anteriormente significa que é esperado que $p = p_K$ não é uma função decrescente de K, concordando com a questão intuitiva que é mais fácil transmitir um pacote apenas durante um contato se o tempo de transmissão é menor. I.e, tipicamente é esperado que o efeito da fragmentação seja benéfica para entregar mais pacotes em contatos isolados, por isso a dependência de p em K não deve ser uma constante.

Por questões de notação, na sequência, será denotado X_K a fração de nós que contém K pacotes quando exatamente K fragmentos foram criados da mensagem original: todavia, partindo de (14) deve-se levar em conta o fato de X_K depender de K e p_K. As observações feitas anteriormente são levadas em conta a seguir

Teorema 3.3.1. *i. Se $p_K < p_{K+1}$, então $X_{K+1} > X_K$*
ii. Se $p_K = p_{K+1}$, então $X_K > X_{K+1}$

Demonstração. i.) por inspeção temos (14).
ii.) Para provar a afirmação, será adotado o argumento variacional usando uma função linear interpolada da expressão original conforme esta aparece em (14). Particularmente, define-se a transformação de $\overline{X} : [0,1] \to [0,1]$

$$\overline{X} : y \to \overline{X}_K(y) = 1 - e^{-xp(y)}\Delta_k y - e^{-xp(y)}\left[1 + \sum_{r=1}^{K-1} p(y)^r \sum_{j=1}^{r} \binom{r-1}{j-1} \frac{[\lambda(1-p(y))t]^j}{j!}\right]$$

onde faz-se, $x = \lambda\tau$, $\Delta_k = p_{K+1}\frac{x^K(1-p_{K+1})^K}{K!}$. Neste caso, define-se

$$p(y) = p_K + \delta_{p_K} y, \quad \text{where} \quad \delta_{p_K} = p_{K+1} - p_K > 0$$

Observa-se que $\overline{X}_K(0) = X_K(t)$ e $\overline{X}_K(1) = X_{K+1}(t)$: faz-se a prova que a transformação é crescente no parâmetro y de tal forma que $X_K(t) = \overline{X}_K(0) < \overline{X}_K(1) = X_{K+1}(t)$.

De fato, basta observar que

$$\begin{aligned}\frac{d}{dy}\overline{X}_K(y) &= \delta\frac{d}{dp}X_K(p) - \frac{d}{dy}\Big(e^{-p(y)x}\Delta_k y\Big) \\ &= \delta\frac{d}{dp}X_K(p) + \Delta_k e^{-p(y)x}\big(-1+x\delta y\big) \qquad (21)\end{aligned}$$

e assim pode-se derivar

$$\begin{aligned}&X_{K+1}(t) - X_K(t) = \overline{X}_K(1) - \overline{X}_K(0) \\ &= \int_0^1 \delta\frac{d}{dp}X_K(p)dy + \Delta_k e^{-p_K x}\int_0^1 e^{-x\delta y}\big(-1+x\delta y\big) \\ &\geq \Delta_k e^{-p_K x}\int_0^{x\delta} e^{-v}\big(-1+v\big) = 1 - x\delta e^{-x\delta} > 0\end{aligned}$$

A primeira inequação segue do fato que $\frac{d}{dp}X_K(p) \geq 0$, onde a inequação final pode ser escrita uma vez que observa-se que para $x \geq 0$, a função xe^{-x} atinge seu máximo em $x = 1$, onde seu valor é $e^{-1} < 1$. Uma vez que o raciocínio é independente do número de pacotes K, isso conclui a prova. □

Nota 3.3.1. *A prova do Teorema 3.3.1 provê a visão do impacto da fragmentação: existem dois efeitos opostos que podem ser visualizados em* (14)*. O primeiro é um efeito local e o outro é global. O efeito global é um aumento polinomial no termo multiplicativo que aparece na soma: isso representa o aumento do número de pacotes e tende a diminuir a probabilidade de entrega de mensagem simplesmente porque um pacote a mais tem de ser entregue. Por outro lado, o efeito local é o aumento da probabilidade de transmissão de um pacote durante um contato, pois este pode durar menos, o que provoca uma diminuição do termo exponencial que multiplica a soma. Basicamente, o Teorema. 3.3.1 diz que quando o caso ii estabiliza, o efeito local sobrepõe o global, enquanto que no caso i o efeito global domina.*

Agora, combinando Lema 3.3.2 e Corolário 3.3.1, obtém-se um resultado que caracteriza a eficiência do mecanismo de fragmentação.

Teorema 3.3.2. *Seja* $p^* = \sup\{p_K, K = 1,2,\ldots,\infty\}$ *e denote* $K^* = \inf\{K : \ p_K = p^*\}$*, então*

$$D_X(K^*) = \sup\{D_X(K) : K = 1,2,\ldots,\infty\}$$

A partir do resultado acima, entende-se que o sup é alcançada somente no caso em que o processo entre encontros é tal que existe um número finito K^* tal que $p_{K^*} = p_K$ para todo

$K \geq K^*$, i.e., acima K^* o sucesso de uma única transmissão de pacotes durante um contato está saturado em algum valor máximo de p^*.

Nesse caso, isso também corresponde ao número ideal de pacotes através da qual a mensagem deve ser fragmentada. Denota-se T_m o tempo de transmissão da mensagem, e seja $T_m(1+K\zeta)$ o tempo de transmissão da mensagem quando fragmentada. O termo $\zeta > 0$ leva em conta a sobrecarga, o que é, possivelmente, devido ao tempo de transmissão e para as operações do protocolo, e.g., o intervalo e/ou procedimentos de estabelecimento da comunicação.

Relembrando que no caso geométrico $p = \overline{p}_1 = \mathbb{P}\{T_c \geq t_1\}$ então

$$p = \mathbb{P}\left\{T_c \geq \zeta + \frac{T_m}{K}\right\} = 1 - F_{T_c}\left(\zeta + \frac{T_m}{K}\right)$$

onde $F_{T_c}(\cdot)$ é a CDF do tempo de contato. Assim, se $K^* < \infty$, é possível escrever o número de pacotes ideal K^* como sendo

Corolário 3.3.1. *Dado o overhead de cada contato $\zeta \geq 0$, a fragmentação ideal*

$$K^* = \min\{K : p_K = 1 - F_{T_c}(\zeta)\} \tag{24}$$

A fragmentação ótima é a menor que satura a probabilidade de sucesso ao valor que corresponde ao overhead da transmissão em um contato.

3.3.1.4 CARACTERIZAÇÃO ASSINTÓTICA

Do fato de K^* ser finito extrai-se o seguinte resultado:

Teorema 3.3.3. *K^* é finito*

Demonstração. Para K crescente, a expressão para um grande número de pacotes como em (14) converge para um valor finito. De fato,

$$\lim_{K\to\infty} X_K(t) = 1 - \lim_{K\to\infty} a_K \cdot b_K \tag{25}$$

onde $a_K = e^{-xp_K}$ e $b_K = 1 + F_K(p_K)$, onde seja $x = \lambda t$ por questão de notação. O limite existe: de fato, $a_K = e^{-xp_K} \to e^{-xp^*}$. Mas, b_K converge e assim é possível escrever

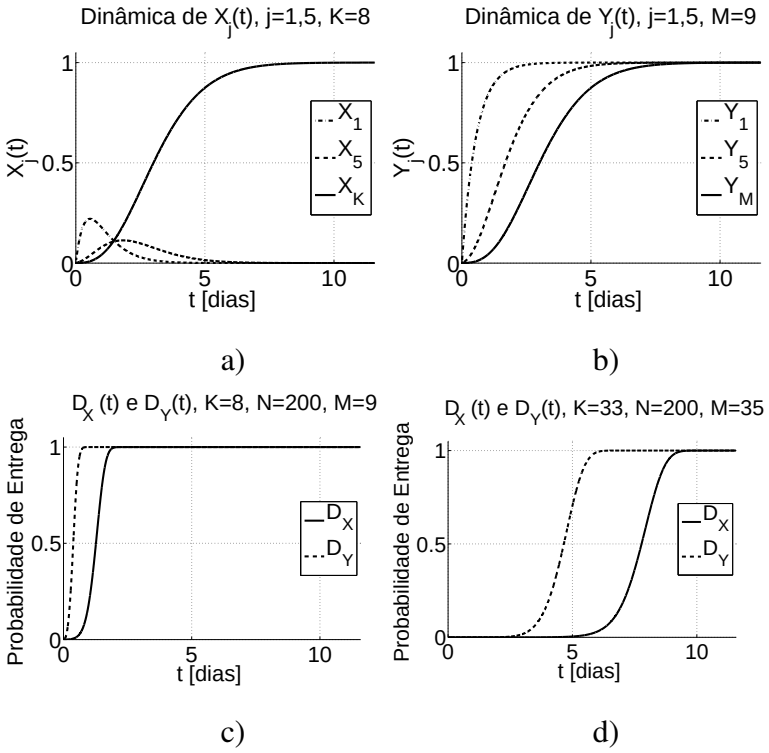

Figura 9: (a) Dinâmica de $X_j(t)$, (b) Dinâmica de $Y_j(t)$ para $j = 1,5,8$ e (c) Probabilidade de entrega de mensagens $D_X(t)$ e $D_Y(t)$ para $K = 8 = M-1$. (d) Probabilidade de entrega de mensagens $D_X(t)$ e $D_Y(t)$ para $K = 33 = M-2$.

$$
\begin{aligned}
F_K(p_K) &= \sum_{j=1}^{K-1} \frac{[\lambda(1-p)t]^j}{j!} \sum_{r=j}^{K-1} p^r \binom{r-1}{j-1} \\
&\leq \sum_{j=1}^{K-1} \frac{[\lambda(1-p)t]^j}{j!} \sum_{r=j}^{\infty} p^r \binom{r-1}{j-1} \qquad (26) \\
&= \sum_{j=1}^{K-1} \frac{[\lambda(1-p)t]^j}{j!} p \sum_{l=j}^{\infty} p^l \binom{l}{j-1} \qquad (27) \\
&= \sum_{j=1}^{K-1} \frac{[\lambda(1-p)t]^j}{j!} \left(\frac{p}{1-p}\right)^j \to e^{px} - 1 \qquad (28)
\end{aligned}
$$

desta forma $\lim_{K\to\infty} X_K(t) \leq 1 - e^{-xp^*} e^{p^* x} = 0$. Segundo o Teorema 3.3.2 e a assumindo que p_K não é decrescente, segue que $K^* < \infty$. □

3.3.2 CÓDIGOS FONTANAIS

Agora, assume-se que a fonte usa pacotes redundantes adicionais para melhorar a eficiência da DTN. Usando códigos fontanais, para todo f, o destino é capaz de decodificar os K pacotes com pelo menos probabilidade de $1-f$ caso tenha recebido pelo menos $M := K\log(\frac{K}{f})$ pacotes codificados. Escrevendo de uma forma diferente, isso significa que qualquer $M := K\log(K/f)$ pacotes codificados permitem decodificar os K pacotes originais com pelo menos a probabilidade $P_s = 1-f$ por em média $K\log(1/f)$ operações (MACKAY, 2003).

Usando códigos fontanais, o nó de origem é capaz de liberar pacotes recém-gerados em cada contato com nós intermediários, de modo que a ordem sequencial de pacotes torna-se irrelevante neste caso: em particular, isto significa que a análise do mecanismo de transmissão sequencial ainda se mantém, mas a decodificação é feita independentemente da ordem em que os pacotes foram recebidos no destino.

Define-se o estado da rede, a distribuição dos pacotes codificados em nós intermediários por uma M-tuple $(X_1,..,X_M)$, onde $X_i(t)$, $i = 0,..,M-1$ denota o número de nós com i pacotes codificados e $X_M(t)$ denota a fração de nós que têm mais de M pacotes codificados. Em seguida, apresenta-se a mesma aproximação de fluido padrão (com base na análise de campo médio) feito no primeiro caso, mas altera-se os valores K para M.

Agora se deseja escrever o número de nós que têm pelo menos i pacotes em função do número de nós que têm exatamente i pacotes para obter a forma fechada como segue. Pelas equações anteriormente elaboradas, temos que:

$$\begin{aligned} Y_0 &= Y_1 + X_0 \\ Y_h(t) &= Y_{h+1}(t) + X_h(t) \\ Y_{M-1}(t) &= Y_M(t) + X_{M-1}(t) \\ Y_M(t) &= X_M(t) \end{aligned}$$

Desta forma, como $X_M(t)$ foi calculado na equação (14) é possível realizar o cálculo recursivo de trás para frente e desta maneira encontrar todos os valores Y_i.[2] Sendo T_M o atraso na entrega da mensagem, i.e, o tempo desde que o primeiro pacote é gerado na origem, até o momento em que o M-ésimo pacote codificado é entregue ao destino.

[2]Implicitamente assume-se que B é o tamanho do buffer de um nó, então $B > M$.

A distribuição (CDF) do atraso T_M é denotada por $D_Y(t) = \mathbb{P}(L_M \leq t)$. Sendo $y_d(t)$ uma variável aleatória que representa o número de mensagens codificadas recebidas pelo destino durante $[0,t]$. $Y_i(t)$ denota o número de nós que possuem ao menos i pacotes, i.e., $Y_i(t) = \sum_{j=i}^{M} X_j(t)$. Daí segue que $X_i(t) = Y_i(t) - Y_{i+1}(t)$. A probabilidade de sucesso é calculada de forma semelhante ao caso não codificado.

$$
\begin{aligned}
& D(t+dt) - D(t) = \mathbb{P}(t < L_M \leq t+dt) \\
= & \sum_{i=0}^{M-1} \mathbb{P}(t < T_M \leq t+dt, y_d(t) = i) \\
= & \sum_{i=0}^{M-1} \mathbb{P}(T_M \leq t+dt | y_d(t) = i, T_M > t). \\
& \mathbb{P}(y_d(t) = i | T_M > t) \mathbb{P}(T_M > t) \\
= & dtN\lambda \sum_{i=0}^{M-1} \Big(\frac{1}{N} + Y_{M-i}(t)\Big) \bar{p}_{M-i} \frac{X_i(t)}{1 - X_M(t)} (1 - D(t)) \\
= & dtN\lambda \sum_{i=0}^{M-1} \bar{p}_{M-i} \Big(\frac{1}{N} + Y_{M-i}(t)\Big) \frac{(Y_i(t) - Y_{i+1}(t))}{1 - Y_M(t)}. \\
& (1 - D(t))
\end{aligned}
$$

$$
\begin{aligned}
dD_Y &= \sum_{i=0}^{M-1} \mathbb{P}(t < T_M \leq t+dt, y_d(t) = i) \\
&= dtN\lambda \sum_{i=0}^{M-1} \bar{p}_{M-i} \Big(\frac{1}{N} + Y_{M-i}(t)\Big) \frac{(Y_i(t) - Y_{i+1}(t))}{1 - Y_M(t)}.
\end{aligned}
$$

onde vale relembrar que $\bar{p}_i = \mathbb{P}(T_c \geq t_i)$.

Assim

$$
\begin{aligned}
\frac{dD_Y(t)}{dt} &= N\lambda \sum_{i=0}^{M-1} \bar{p}_{M-i} \Big(\frac{1}{N} + Y_{M-i}(t)\Big). \\
& \qquad \frac{(Y_i(t) - Y_{i+1}(t))}{1 - Y_M(t)} (1 - D_Y(t))
\end{aligned}
$$

Então integra-se

$$\int_0^t \frac{dD_Y}{1-D_Y} =$$
$$N\lambda \sum_{i=0}^{M-1} \int_0^t \bar{p}_{M-i}\Big(\frac{1}{N} + Y_{M-i}(s)\Big) \frac{(Y_i(s) - Y_{i+1}(s))}{1-Y_M(s)} ds$$

A expressão separável é integrada para obtenção da expressão final

$$D_Y(t) = 1 - \exp\Big(N\lambda \sum_{i=0}^{M-1} \int_0^t ds\, \bar{p}_{M-i}\Big(\frac{1}{N} + Y_{M-i}(s)\Big) \cdot$$
$$\cdot \frac{(Y_i(s) - Y_{i+1}(s))}{1-Y_M(s)}\Big)$$

onde no caso geométrico $\bar{p}_{M-i} = p^{M-i}$.

3.3.3 RESULTADOS NUMÉRICOS

Nesta seção será realizada uma caracterização numérica dos modelos obtidos anteriormente. Foi reportado em Figura 9 o comportamento da transmissão sequencial e do esquema utilizando códigos fontanais previstos pelo modelo para dois casos diferentes, i.e., $K = 8$ e $K = 33$ e para o mesmo valor $p = 0.4$. Em particular, na Figura 9a) é possível observar o comportamento transiente da fração dos nós infectados que contém $j = 1,5$ e $K = 8$ pacotes para a entrega sequencial (X_is) e na Figura 9b) a dinâmica prevista para as variáveis Y_i correspondentes ao utilizar os fountain codes. Como é possível observar, todos os X_is para $0 < i < K$ possuem o seu pico e depois decrescem conforme o esperado (a quantidade de nós que possuem i mensagens reduz com o passar do tempo para os is pequenos e aumenta conforme i tende para K); ao contrário, as variáveis Y_i são monotonicamente crescentes. Mesmo sendo o número final de nós infectados iguais $X_M = Y_M$, o esquema codificado provê uma melhora acentuada: o ganho fundamental é devido ao fato de que o sistema de encaminhamento não restringe o repasse de pacotes para acontecer sempre ordenadamente e, portanto, permite a reconstrução da mensagem muito mais rápida no nó destino.

Em suma, é possível comparar o comportamento da dinâmica no caso de $K = 8$ e $K = 33$, Figura 9c) e Figura 9d), respectivamente: claramente, a partir do Teorema 3.3.2, uma vez que foi utilizado $p_8 = p_{35} = 0.4$, o aumento no número de pacotes tem um efeito prejudicial na probabilidade de entrega. Mas pode-se ver que o ganho relativo à codificação, combinado com a não necessidade de ordenamento dos pacotes, fornece efetivamente um ganho relativo maior para um maior número de pacotes.

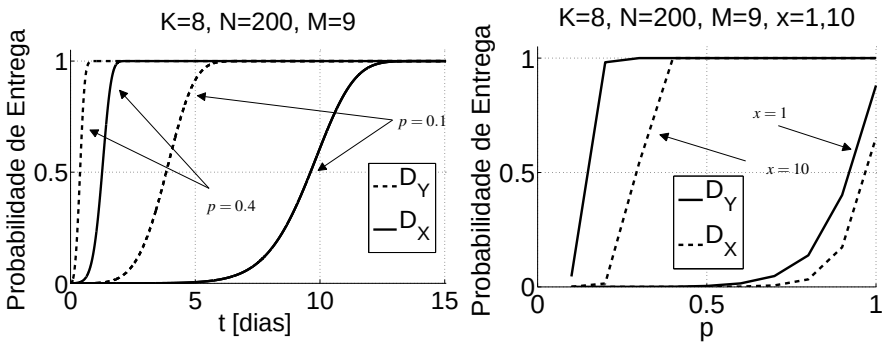

Figura 10: a) Dependência de D_X e D_Y na probabilidade de transmissão por pacote $p = 0.1$ e $p = 0.4$, respectivamente.b) Dependência de D_X e D_Y na probabilidade de entrega p para dois valores de $x = \lambda\tau$

O efeito de p na probabilidade de entrega é mostrado na Figura 10a): o aumento de p de 0.1 para 0.4 configura uma aceleração da dinâmica, que por sua vez reflete na probabilidade de entrega. Nota-se mais uma vez como o ganho de D_Y frente a D_X em termos da probabilidade de entrega é muito maior para valores pequenos de p, o que decorre do fato de que a entrega é desordenada no caso codificado. Finalmente, o efeito do aumento de p é exibido para um valor de $x = \lambda\tau$: nesse caso, é possível observar uma transição brusca da probabilidade de entrega como uma função de p, i.e., a probabilidade de transmissão de pacote por contato. Este resultado assemelha-se aos efeitos da transição de fase mostrados em (ALTMAN; De Pellegrini, 2011), onde o parâmetro era, na verdade, o controle de encaminhamento usado no nó de origem, a fim de reduzir o número de nós infectados e economizar recursos de rede. No entanto, nesse contexto os nós eram modelados tendo buffer para apenas um pacote.

Com base neste estudo realizado, pode-se utilizar os valores de D_Y e p para definir os critérios de parada e a quantidade de redundância que serão inseridos na transmissão para este tipo de rede. Esses valores podem ser utilizados nas equações que serão descritas na sequência, para estimar os valores de α (diversidade de informações) e t (tempo para fechamento da fonte), visto que o atraso é o tempo necessário para a transmissão ocorrer e a quantidade de redundância enviada (α) é o inverso da taxa de entrega das informações.

Ao se utilizar os códigos fontanais é possível visualizar que, como a geração de mensagens codificadas pode ser realizada de maneira contínua, é fácil saturar a rede com mensagens diferentes enchendo os *buffers* dos nós intermediários e desperdiçando muitos recursos como: energia e processamento. Para evitar o uso ineficiente dos recursos por parte do protocolo de

transporte proposto, uma forma de conter a geração das mensagens foi desenvolvida.

Inicialmente, para se efetuar o controle da geração das mensagens foi realizado um estudo da quantidade de mensagens que trafegavam na rede.

Conforme descrito anteriormente, com a utilização de códigos fontanais, não importam quais foram os pacotes recebidos, o que importa é apenas que o número de pacotes recebidos seja suficiente para que a decodificação ocorra. Percebe-se porém, que o custo de manter mensagens por muito tempo trafegando na rede não é a melhor opção devido ao tráfego paralelo, onde manter mensagens na rede pode significar mensagens em zonas isoladas, que não atingem o destino e acabam por desperdiçar *buffer* e energia dos *relays*. O problema então não é o tamanho físico do buffer mas sim o tempo e energia para enviar estas mensagens. Por isso o método adotado para se resolver este problema foi ajustar o TTL (*Time To Live* ou tempo de vida) dos pacotes e o intervalo de geração das mensagens. Vale ressaltar que muitas mensagens no buffer significa que será necessário um grande tempo de contato para transmiti-las.

Para não sobrecarregar a rede e desperdiçar recursos, foi usado como referência o uso da rede quando os dados são enviados sem o uso do PTTA, afim de modelar equações que irão reger este controle. Para controlar os recursos da rede foi manipulada a geração das mensagens para que não fossem utilizados recursos adicionais da rede. Para isso foi levado em consideração o TTL das mensagens no envio sem a utilização dos códigos fontanais e o intervalo de geração dessas mensagens, isto é a quantidade de informação presente na rede no tempo. Além disso, foi analisado o número de mensagens de um fluxo que trafegava na rede em cada instante de tempo de simulação e foi modelada uma equação que visa reproduzir o formato de tráfego (quantidade de mensagens na rede) próximo ao obtido sem a utilização do PTTA. A equação obtida para controlar a quantidade de dados enviados é a equação 36.

$$\Delta g_P = \frac{\psi_p}{M} \tag{36}$$

onde ψ_p é o tempo de vida (em segundos) das mensagens ao se enviar os dados com o PTTA, M é o número máximo de mensagens (desta transmissão) que será permitido coexistir na rede num instante de tempo (o qual é baseado no uso da rede sem a utilização do PTTA) e Δg_P é o intervalo de tempo existente entre a geração de uma mensagem e outra.

Outro critério importante é o momento do fechamento da fonte. Este momento é definido com base no tempo em que as mensagens enviadas sem o PTTA ficam trafegando na rede. Desta forma, não se estará utilizando recursos adicionais da rede na transmissão. Para este fechamento foi modelada outra equação (37):

$$t = \psi_r - (M \times \Delta g_r) \tag{37}$$

onde t é o tempo no qual a fonte será fechada e é dado em segundos, ψ_r é o tempo de vida das mensagens sem o uso do PTTA, M é o número máximo de mensagens (desta transmissão) que será permitido coexistir na rede e Δg_r é o intervalo de geração das mensagens na transmissão sem o PTTA.

Para obter uma equação mais geral, melhorada e um pouco mais intuitiva, algumas alterações nas equações anteriormente descritas foram realizadas. Uma variável chamada de "diversidade de informações" (α_{PTTA}) foi incluída, a qual representa o número de pacotes diferentes enviados pela aplicação de origem na rede.

O mecanismo proposto controla os recursos de rede, limitando a sobrecarga e desperdício de recursos. Ele ajusta a taxa de geração segmento do PTTA λ_{PTTA} para moldar a carga da rede e manter o consumo de recursos abaixo do limiar de referência. Para atingir este objetivo, o TTL dos pacotes de transporte ψ e a taxa de geração de pacotes λ, *i.e.* a carga da rede, são contabilizados. Observa-se que tais valores são ambos locais e conhecidos em cada transmissão DTN.

Eq.(38) foi modelada para reproduzir a carga da rede, tendo em conta a diversidade de informações. Diversidade significa aqui o número de diferentes pacotes enviados por uma fonte de aplicação (nó origem) na rede.

$$M_{Max} = \begin{cases} (\lambda \times \alpha) \times \psi & \text{se } \psi \leq \Delta t \\ (\lambda \times \alpha) \times \Delta t & \text{caso contrário} \end{cases} \tag{38}$$

onde M_{Max} é o número máximo de pacotes diferentes enviados pelo nó origem na rede para transmitir um dado conteúdo, λ é a taxa de geração de pacotes, ψ é o TTL dos pacotes (em segundos), Δt é o tempo de geração dos pacotes e α é a taxa de diversidade de informação, i.e. quando o PTTA não é usado $\alpha = 1$.

A descontinuidade da função é devido ao caso em que não há mensagens suficientes a serem geradas (enviadas) durante o TTL ψ. Nesse caso, o tempo de geração de Δt é menor do que o TTL de uma mensagem e este é o fator limitante do M_{Max}. Pois caso essa descontinuidade não fosse levada em consideração, seriam geradas mais mensagens enquanto as outras ainda não expiraram devido ao valor do TTL, o que aumentaria a carga na rede e tornaria o método ineficiente.

Para limitar a carga da rede e, consequentemente, o desperdício dos recursos, o meca-

nismo proposto utiliza a Eq.38: que representa a carga máxima da rede gerada por uma fonte de aplicação, i.e., o número máximo de pacotes diferentes que trafegam na rede originado por uma única transmissão de conteúdo. Nesta equação, os atributos locais λ, ψ, Δt e $\alpha = 1$ são conhecidos e o valor unitário é devido ao fato de que é necessário enviar, pelo menos, uma cópia de cada mensagem a fim de ter a possibilidade de receber todas as mensagens. Esses são atributos locais do aplicativo sem PTTA.

O resultado apresentado na Eq.39, une esses parâmetros para descrever a relação entre λ_{PTTA} que representa a taxa de geração de pacotes do PTTA e o valor correspondente de α_{PTTA}, que representa a diversidade informação do PTTA. O melhor α_{PTTA} é dependente da topologia da rede. O TTL, usado nos pacotes que transportam os segmentos do PTTA ψ_{PTTA}, está representado na Eq.40 e correlaciona ψ_{PTTA}, ψ, e α_{PTTA}.

$$\lambda_{PTTA} = \frac{M_{Max}}{\psi_{PTTA} \times \alpha_{PTTA}} \tag{39}$$

$$\psi_{PTTA} = \frac{\psi}{\alpha_{PTTA}} \tag{40}$$

Outro ponto importante é saber quando parar a geração de segmentos do PTTA, porque na abordagem de códigos fontanais, o que se tem é um gerador de códigos contínuo. Este tempo de parada, chamado de TTL_{fluxo}, é definido com base no tempo em que o último pacote gerado Δt será descartado pelo TTL (tempo de expiração). Esta relação é exibida na Eq.41.

$$TTL_{fluxo} = \Delta t + \psi \tag{41}$$

Para obter o tempo de parada adequado (da geração de segmentos), é necessário utilizar o mesmo TTL_{fluxo} para o caso com PTTA. Usando a equação 41 sem mudar o TTL_{fluxo} é possível calcular o Δt_{PTTA}, i.e., o tempo de parada da geração de segmentos PTTA, conforme apresentado na Eq.42. Vale relembrar que após esse período, não serão utilizados recursos de rede adicionais para gerar novos dados codificados.

$$\Delta t_{PTTA} = \Delta t + \psi - \psi_{PTTA} \tag{42}$$

4 AVALIAÇÃO DA PROPOSTA

Apesar deste trabalho tratar da camada de transporte é importante descrever os protocolos da camada de rede (protocolos de roteamento) que serão utilizados nos testes. O uso desses protocolos é importante para mostrar a independência do protocolo de transporte desenvolvido perante os diversos protocolos de roteamento. Os protocolos descritos a seguir são os mais renomados e utilizados na literatura.

Epidêmico

O protocolo de roteamento epidêmico é um dos mais populares e utilizados em DTNs devido a sua alta taxa de entrega. O seu funcionamento é estocástico e utiliza-se da inundação para atingir o nó destino. Para entender como ele funciona pode-se fazer um comparativo com uma epidemia (a isso se deve o nome de epidêmico). Numa epidemia, uma pessoa infectada por um vírus, por exemplo, propaga este vírus para todas as pessoas com quem ela entra em contato. Fazendo assim com que todas essas pessoas fiquem infectadas e agora quando qualquer pessoa se aproximar de qualquer uma dessas infectadas, ela também será infectada e assim sucessivamente. Neste protocolo de roteamento o "vírus", da epidemia, não é na realidade um vírus e sim a informação que se deseja propagar. Como a epidemia, em geral, acaba atingindo uma grande parte da população, a informação, neste protocolo de roteamento, acaba atingindo a maioria dos nós e consequentemente o destino que era o objetivo principal (VAHDAT; BECKER, 2000).

First Contact

O nome *First Contact* significa primeiro contato. Este protocolo de roteamento recebe este nome devido ao seu princípio de funcionamento. Ou seja, o nó de origem, ao contrário do que ocorre com o protocolo epidêmico, armazena a informação que deseja transferir para o nó de destino e apenas repassa esta informação quando entra no alcance do nó de destino. Portanto, ele só transmite a informação diretamente para o destinatário dos dados, não sobrecarregando a rede, economizando recursos dos nós intermediários, porém, com uma taxa de entrega menor, visto que se ele nunca entrar em contato com o destino, estas informações nunca serão entregues (SHAH et al., 2003) (SMALL; HAAS, 2005) (ZHAO et al., 2005) (JAIN et al., 2004a)

(JAIN et al., 2004b).

Spray and Wait

O protocolo do tipo *Spray and Wait* é um protocolo estocástico, como o protocolo epidêmico, porém que visa economizar recursos. O seu funcionamento se divide em duas fases, a fase *spray* e a fase *wait*, que originam o seu nome. Na fase *spray*, o nó origem, baseado numa estimativa do número de nós da rede, seleciona o número de réplicas (L) da mensagem que ele irá criar. Este nó de origem cria as L mensagens e repassa para $L-1$ nós, sendo que uma das mensagens é mantida no próprio nó origem. Na fase *wait*, se o destino não recebeu a mensagem na fase *spray*, cada um dos nós que contém a mensagem só a entregará ao destino, caso ele seja encontrado (SPYROPOULOS et al., 2005).

Ainda existe uma variação do protocolo *spray and wait* chamada *binary*. Nesta variação, o nó origem instancia L cópias da mensagem M original e caso o nó X tenha recebido $n > 1$ cópias da mensagem M, quando o nó X encontrar com o nó Y, que não possui nenhuma cópia da mensagem original, X irá transferir $n/2$ cópias da mensagem para o nó Y. Esse processo se repete, até que cada nó fique com apenas 1 réplica de cada mensagem quando esta réplica somente será repassada ao destino (SPYROPOULOS et al., 2005).

PROPHET

O protocolo PROPHET (*Probabilistic ROuting Protocol using History of Encounters and Transitivity*) estima a probabilidade de entrega das mensagens utilizando o histórico de encontros com cada nó conhecido (LINDGREN et al., 2003). Uma métrica com nome de *Delivery Predictability* é computada para cada nó conhecido, o valor deste varia no intervalo de 0 a 1. Onde $P_{(a,b)}$, por exemplo, significa o valor de *Delivery Predictability* do nó a para o nó b. A equação 43 descreve o cálculo de $P_{(a,b)}$ e é calculada para cada novo encontro entre os nós a e b:

$$P_{(a,b)} = P_{(a,b)anterior} + (1 - P_{(a,b)anterior}) \times P_{inicial}, \tag{43}$$

onde $P_{(a,b)anterior}$ é o valor de $P_{(a,b)}$ anterior ao novo contato entre a e b, $P_{inicial}$ é uma constante definida na inicialização e $P_{(a,b)}$ é o novo valor calculado.

Quando ocorre o encontro entre dois nós a e b, o nó a encaminha a mensagem m destinada ao nó c somente se o valor de $P_{(b,c)} > P_{(a,c)}$. Além disso, quanto mais vezes os nós a e b se encontrarem, mais próximo de 1 será o valor de $P_{(a,b)}$, porém se estes nós permanecerem desconectados por algum tempo, $P_{(a,b)}$ tem seu valor reduzido de acordo com a equação 44:

$$P_{(a,b)} = P_{(a,b)anterior} \times \gamma^k, \tag{44}$$

onde γ é a constante de envelhecimento (definida na inicialização) e k é o tempo decorrido desde a última vez em que a métrica *Delivery Predictability* foi envelhecida. Este tempo pode ser controlado de maneiras diferentes e deve ser definido dependendo da aplicação e do atraso desejado (LINDGREN et al., 2003).

Além disso, a *Delivery Predictability* tem ainda uma **transitividade**. Esta transitividade é baseada nos encontros entre os nós, supõe-se que o nó a frequentemente encontra-se com b e o nó b encontra-se frequentemente com c, então, provavelmente o nó c é uma boa escolha para encaminhar mensagens destinadas ao nó a. A equação 45 mostra como esta transitividade afeta a *Delivery Predictability* (LINDGREN et al., 2003).

$$P_{(a,c)} = P_{(a,c)anterior} + (1 - P_{(a,c)anterior}) \times P_{(a,b)} \times P_{(b,c)} \times \beta, \tag{45}$$

onde $\beta \in [0,1]$ é uma constante de escala que determina o impacto da transitividade sobre a *Delivery Predictability* (LINDGREN et al., 2003).

Maxprop

A essência do maxprop está em uma lista de pacotes armazenados dos seus pares baseado em um custo atribuído a cada destino. Este custo é uma estimativa da probabilidade de entrega e esta lista é chamada pelo protocolo como *delivery likelihood*. Este protocolo dá preferência para as conexões mais recentes e evita o recebimento em duplicata de informações. A estimativa da probabilidade de entrega que está presente na *delivery likelihood* é calculada da maneira explicada a seguir (BURGESS et al., 2006).

Supõe-se o conjunto de nós na rede sendo s. Cada nó, $i \in s$, mantém a probabilidade de encontrar o nó $j \in s$. Estima-se que a probabilidade ${f_j}^i$ como sendo a probabilidade do nó i se conectar com o nó j. Inicialmente para todos os nós ${f_j}^i$ é atribuído como sendo $\frac{1}{|s|-1}$, o que torna a soma de todas as probabilidades igual a 1, portanto esta lista está normalizada. Quando o nó j é encontrado, o valor de ${f_j}^i$ é incrementado em 1, e todos os valores de f são re-normalizados (no caso divididos por 2). Usando este método, também chamado de média incremental, os nós vistos sem frequência (raramente) terão valores pequenos no decorrer do tempo, já o último nó encontrado terá o maior valor (BURGESS et al., 2006).

Com o valor da lista dos outros nós, o nó local calcula o custo $c(i, i+1, ..., d)$ para cada caminho possível para o destino d com comprimento de até n saltos. O custo para um

caminho usando os nós $i, i+1, ..., d$ é a soma das probabilidades de cada conexão do caminho não ocorrer, que é estimada como sendo um menos a probabilidade de cada ligação ocorrer, conforme a equação 46 (BURGESS et al., 2006).

$$c(i, i+1, ..., d) = \sum_{x=i}^{d-1} [1 - (f_{x+1}^{x})]. \tag{46}$$

Two Hops

Two Hops traduzido para o português significa dois saltos. Este protocolo recebe este nome, justamente por ser limitado a dois saltos a propagação das mensagens. Em outras palavras, o nó origem repassa para qualquer nó que este encontrar, como no roteamento epidêmico, porém, este nó que recebeu a mensagem da origem, só repassará a mensagem caso encontre com o nó de destino, perfazendo o segundo salto (GROSSGLAUSER; TSE, 2002) (GROSSGLAUSER; TSE, 2001).

Este protocolo é o mais utilizado em modelizações matemáticas, por existirem técnicas que simplificam a descrição do problema. Isso ocorre pois as combinações possíveis de repasse entre os nós é limitada a dois saltos, quando na realidade o roteamento epidêmico, por exemplo, possui uma forma muito mais complexa devido a possibilidade do repasse das mensagens de todos e para todos os nós existentes na rede (GROSSGLAUSER; TSE, 2002) (GROSSGLAUSER; TSE, 2001).

Além disso, pelo fato de limitar os saltos em dois, a carga na rede é diminuída, pois para que todos os nós intermediários recebam a mensagem, isso implica que todos eles se encontrem com o nó de origem, o que pode, eventualmente, não ocorrer (GROSSGLAUSER; TSE, 2002) (GROSSGLAUSER; TSE, 2001).

4.1 DESCRIÇÃO DOS CENÁRIOS UTILIZADOS

A avaliação de desempenho da proposta foi realizada utilizando o simulador The ONE (KERÄNEN et al., 2009) nos cenários descritos a seguir e usando os protocolos de roteamento citados anteriormente.

4.1.1 CENÁRIO 1 (REDE DEDICADA)

Primeiramente, o cenário de Helsinki (Figura 11) foi utilizado. Foram inseridos 126 nós divididos em ônibus, carros, pedestres e *tramways*. Cada um desses tipos possuía suas

próprias velocidades, característica de movimento, caminho permitido para o seu movimento e tamanho de *buffers*. O caminho permitido para o movimento de cada um dos tipos é importante por existirem ruas com sentido único, onde os pedestres podem se locomover em ambos os sentidos mas os carros não. Ainda existem caminhos que, por exemplo, apenas os *tramways* podem se deslocar. Além disso, todos esses nós operam com uma interface de comunicação com taxas de transmissão de 250KBps.

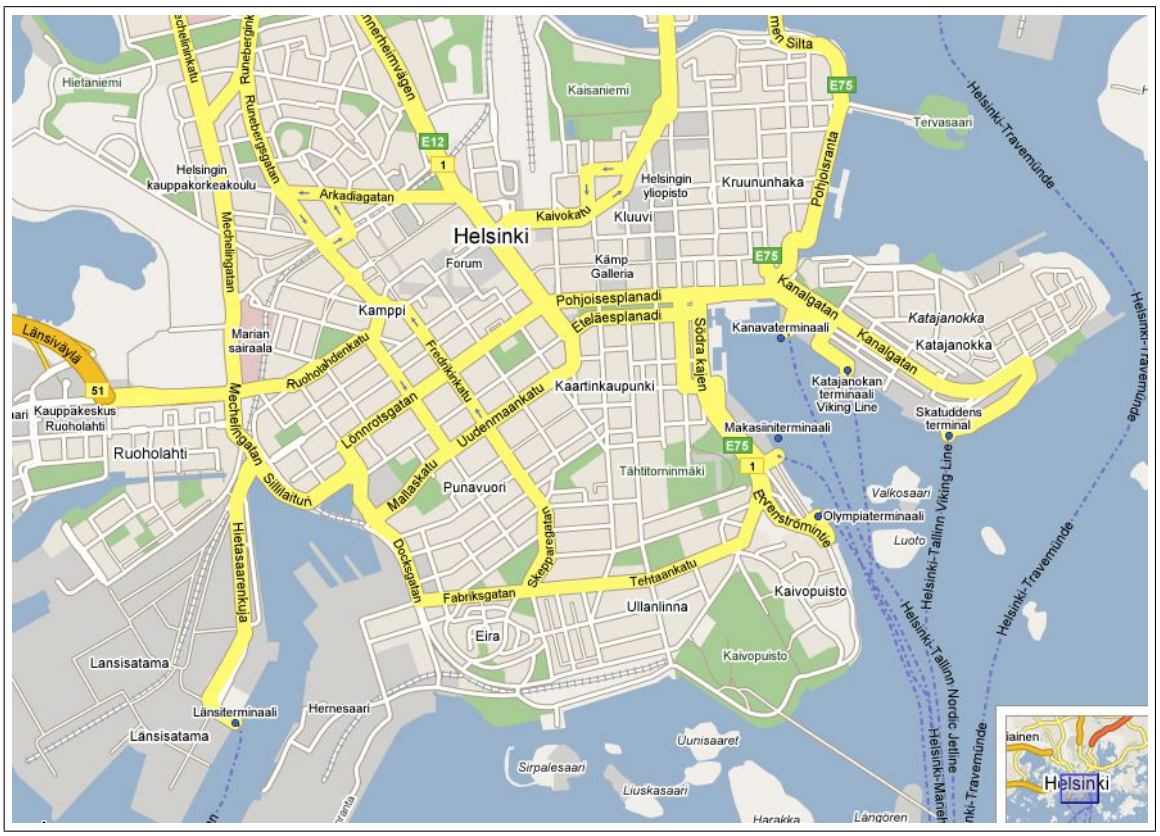

Figura 11: Mapa de Helsinki utilizado na simulação (KERÄNEN et al., 2009).

Neste cenário em cada simulação havia apenas um nó gerador de mensagens (origem) e todas eram destinadas a um mesmo hospedeiro (destino), simulando a transferência de dados entre dois nós. Estes nós foram selecionados aleatoriamente e a cada simulação esta seleção se deu com base na semente passada por parâmetro, o que permite a reprodução dos resultados. A simulação foi executada até que toda a informação pudesse ser decodificada no receptor ou o tempo limite da simulação fosse atingido. Nesta simulação não havia tráfego concorrente, ou seja, uma situação hipotética do melhor caso para qualquer transmissão. Por isso, uma rede dedicada, ou seja, com somente um fluxo de dados, foi simulada. Como todos os nós estão à disposição para transmitirem apenas as mensagens deste fluxo, as perdas são pequenas.

Vários tamanhos de dados foram testados. Para simular o envio de pequenos arquivos foi utilizada uma informação de 5MB o que poderia ser uma música em MP3. Para uma

quantidade intermediária de dados foi utilizado um tamanho de 10MB simbolizando um vídeo pequeno. Para arquivos médios foi utilizada uma informação de 50MB simbolizando um aplicativo, algumas fotos ou até mesmo arquivos pessoais. E, para grandes informações foi utilizado um conjunto de dados de 100MB representando a transferência de arquivos de Backup. Para cada um desses tamanhos foram realizadas 100 simulações com sementes aleatórias distintas. Com isso, todos os resultados apresentados serão médias dos resultados das 100 simulações com intervalos de confiança de 95%.

Os pacotes que trafegavam na rede tiveram seu tamanho estipulados em 500KB. Além disso, neste primeiro cenário foi utilizado o protocolo de roteamento epidêmico (VAHDAT; BECKER, 2000) conforme implementado no simulador.

Neste cenário, os testes foram subdivididos em dois. Na primeira parte foi realizada a transferência dos dados sem o uso do protocolo de transporte desenvolvido, também chamada "sem FC", ou seja, as informações foram segmentadas e enviadas pela rede utilizando-se do protocolo de rede epidêmico (cada fragmento foi enviado uma única vez, com TTL - *Time To Live*) infinito, fazendo com que as mensagens não sejam expiradas. Na segunda parte foi efetuada a transferência com o uso do protocolo de transporte desenvolvido, também chamados "com FC", onde as informações foram codificadas e em seguida enviadas através da rede sem fechamento da fonte (geração infinita de pacotes codificados) e utilizando-se do mesmo protocolo de rede (epidêmico) do primeiro passo e com TTL infinito. Em todos os testes os parâmetros de entrada eram iguais com relação às configurações da rede.

A Figura 12 apresenta o resultado da percentagem de informações recebidas em função do tempo para os diferentes tamanhos de dados testados (5MB, 10MB, 50MB e 100MB), os intervalos de confiança foram omitidos por motivo de clareza.

Em todos os casos testados, os resultados com o uso do protocolo de transporte proposto obtiveram um melhor resultado em relação aos que não utilizaram o protocolo, significando que com as técnicas de códigos fontanais se obteve uma maior eficiência na entrega das informações, mesmo no melhor caso. Isso ocorre porque ao utilizar o protocolo de transporte descrito não importa quais pacotes foram recebidos para realizar a decodificação das informações, no momento em que o destino receber a quantidade suficiente de segmentos, as informações originais serão recuperadas. Por outro lado, quando não se utiliza do protocolo descrito, as informações originais só podem ser inteiramente recebidas se todas as partes da informação segmentada forem recebidas. Neste aspecto, no cenário onde não é utilizado o protocolo de transporte proposto, pode ocorrer de um dos fragmentos da informação ficar "preso" em alguma área da rede que venha a ficar incomunicável. Isso faria com que a informação inteira ficasse

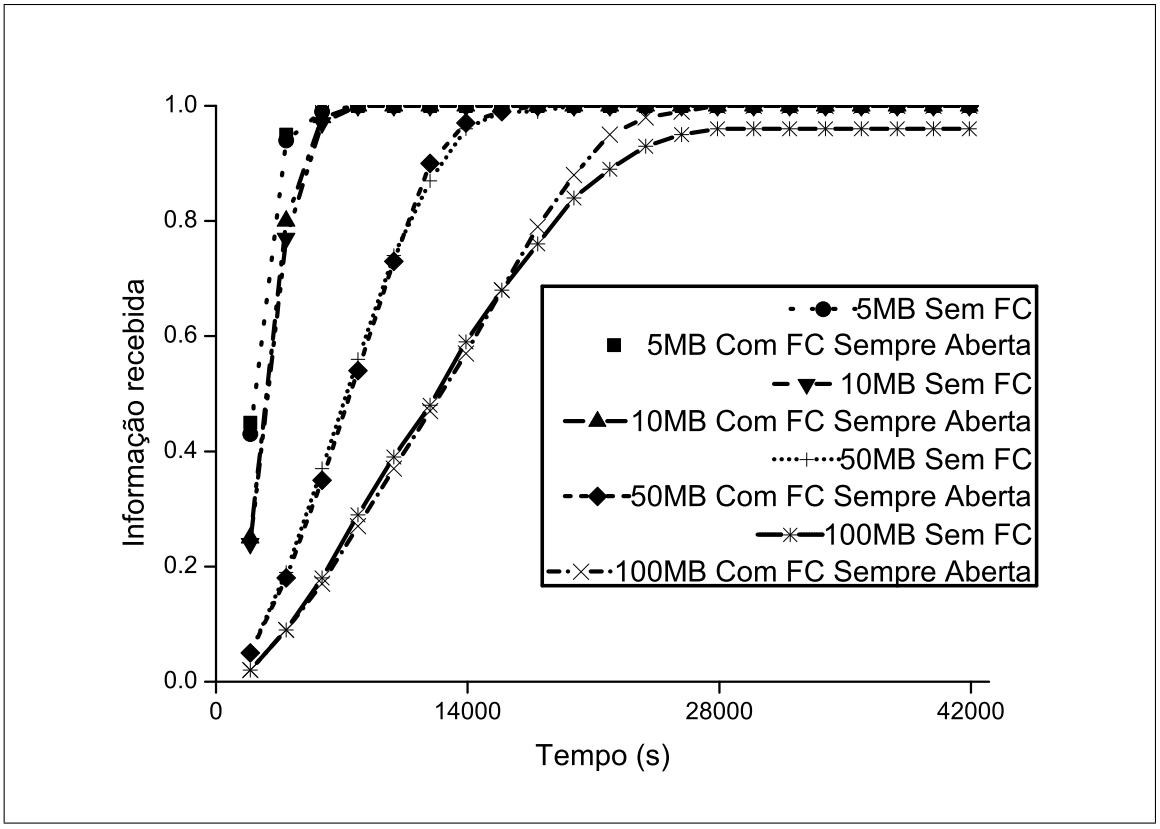

Figura 12: Quantidade de informações recebidas por tempo de simulação.

inutilizada até que este fragmento fosse recebido, como ocorreu nas simulações de números 52, 53, 55, 56, 57, 58, 59, 62 e 63 para os dados com tamanho de 100MB e sem o uso dos códigos fontanais, lembrando que as mensagens foram enviadas com TTL infinito, o que significa que elas estavam presentes na rede. Isso justifica o motivo da Figura 12 para tamanho de 100MB e sem o uso do protocolo de transporte apresentar um máximo que não atinge 100% no recebimento das informações. É importante comentar que quando se utiliza os códigos fontanais, este problema não impossibilita o recebimento da informação, pois as informações deste fragmento estarão disponíveis em outros pacotes como redundância.

Ainda na Figura 12 é possível perceber que para arquivos pequenos, como os de 5MB, a diferença no tempo gasto para se entregar as informações nas duas situações testadas é de cerca de 5% já a mesma diferença para arquivos maiores como os de 50MB e 100MB foi de cerca de 10,5% e 11,5%, respectivamente, lembrando que este é o melhor caso, pois não existe tráfego concorrente.

Após ter sido realizado o teste sem o fechamento da fonte foram observadas quais eram as últimas mensagens recebidas pelos destinos para se verificar a possibilidade de fechamento da fonte com o objetivo de reduzir a quantidade de pacotes que trafegam na rede. Este resultado é apresentado na Tabela 2.

Tabela 2: Identificador da última mensagem recebida em média e com intervalo de confiança de 95% no teste com a fonte sempre aberta.

Tamanho	Mínimo	Médio	Máximo
5MB	46	51	56
10MB	77	83	90
50MB	332	346	360
100MB	643	663	684

A partir dos resultados da Tabela 2 foram realizados novos testes com o fechamento da fonte utilizando-se o valor máximo do intervalo de confiança por se ter 95% de garantia que a média encontra-se abaixo desse valor. O resultado deste experimento foi praticamente o mesmo, porém com um número de mensagens geradas bem menor. Este resultado pode ser observado na Tabela 3.

Tabela 3: Número de pacotes criados em média com fechamento.

Tamanho	Sempre Aberta	Temporizada	Diferença
5MB	120	56	53,3%
10MB	174	90	48,3%
50MB	500	360	28,0%
100MB	866	684	21,0%

Após, foi realizada a comparação dos resultados obtidos pelos testes utilizando a fonte sempre aberta e a fonte temporizada. Os resultados são apresentados na Figura 13. Como é possível perceber, a variação no tempo de entrega dos dados foi pequena, quase imperceptível através do gráfico. Isso é prova que o protocolo de transporte é robusto e não precisa ficar enviando mensagens o tempo todo para poder realizar a entrega dos dados originais, o que ocasionaria muita perda de recursos.

Após isso, como a alteração foi pequena, foi realizado mais um teste utilizando um tempo menor para o fechamento da fonte. Esse tempo foi estipulado com base no limite inferior da média da última mensagem recebida no teste com a fonte temporizada. Novamente, foi utilizada como base a média com intervalo de confiança de 95% e o resultado está apresentado na Tabela 4.

Agora a diferença quanto ao número de mensagens criadas na rede no caso da fonte sempre aberta e neste segundo teste com o fechamento da fonte foi de 69,2%, 60,0%, 38,2% e 29,0% menos mensagens criadas na rede, para os tamanhos 5MB, 10MB, 50MB e 100MB, respectivamente.

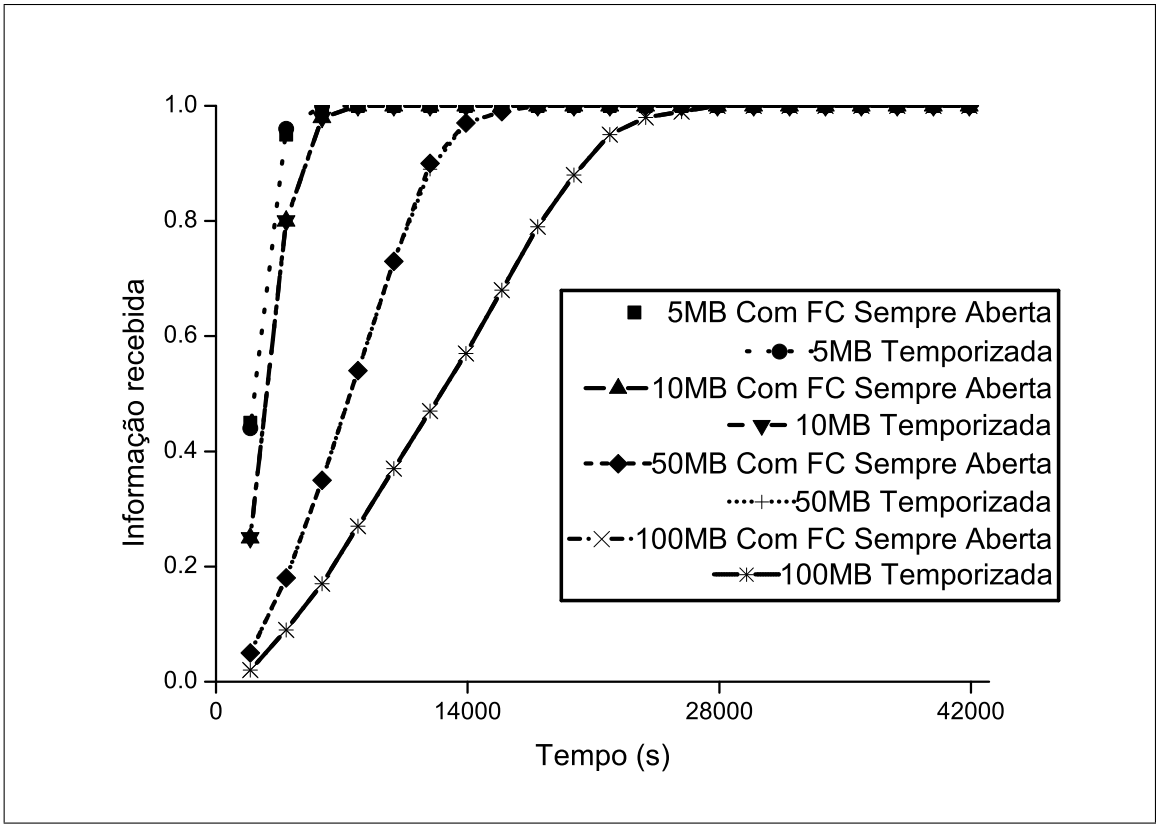

Figura 13: Comparativo da quantidade de informações recebidas por tempo de simulação onde a fonte está sempre aberta e para a fonte temporizada.

Tabela 4: Número de pacotes criados no segundo teste com o fechamento da fonte.

Tamanho	Mínimo	Médio	Máximo
5MB	37	40	55
10MB	69	72	75
50MB	309	316	324
100MB	615	626	636

Foi verificado que com o fechamento da fonte neste segundo teste o ganho no tempo de entrega das mensagens foi ligeiramente maior como pode ser verificado na Figura 14 com os gráficos para os testes sem códigos fontanais e para esta segunda fonte temporizada, onde a maior diferença é visível nos gráficos de 50MB de dados.

Os resultados, até agora, mostram que o uso do protocolo de transporte é promissor. Isso pode ser verificado pelo ganho obtido no tempo de entrega das informações, como mostrado nas Figuras 12 e 14. Com esse ganho é possível verificar um aumento na taxa de entrega, uma vez que em todos os testes realizados, com o uso dos códigos fontanais, foram entregues 100% das informações originais, já nas simulações sem o uso do PTTA, isso não aconteceu, frisando que este é o melhor caso.

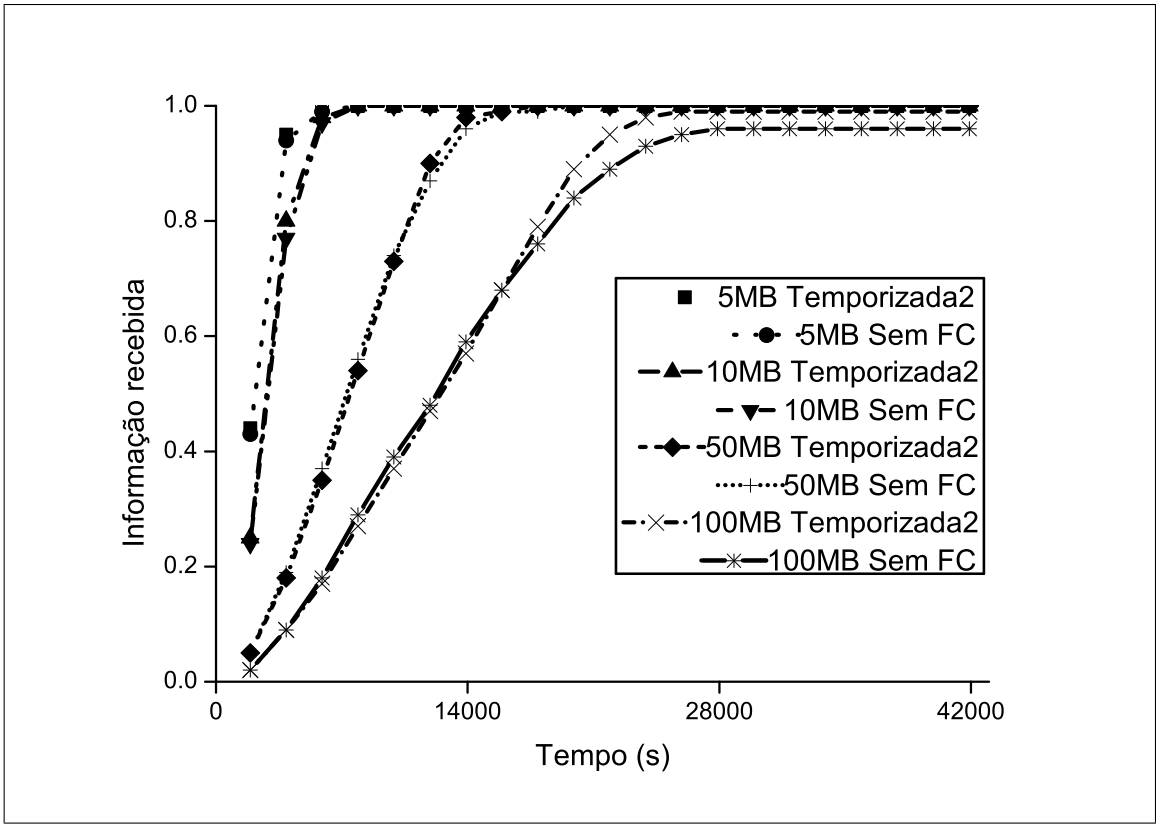

Figura 14: Informações recebidas por tempo de simulação para o caso sem códigos fontanais e para a segunda fonte temporizada.

Os resultados deste primeiro cenário e descritos anteriormente foram publicados em (ALBINI et al., 2011b).

Numa outra etapa, para continuar a avaliar o protocolo de transporte desenvolvido foram realizados testes utilizando o tempo de vida das mensagens (TTL), tamanhos de buffer e quantidade de informações redundantes geradas (tempo de fechamento da fonte) diferentes. Estes testes foram divididos nas mesmas duas partes. A primeira sem o uso do protocolo de transporte. E a segunda com o protocolo de transporte. Nestes testes foi utilizado o cenário 1 e com o protocolo de roteamento Epidêmico.

Na Figura 15 é mostrado o desempenho onde os nós possuem 5MB de buffer, exceto os tramways que possuem 50MB. Dois valores diferentes de Tempo de vida (Time To Live - TTL) dos pacotes foram testados. Um de 60 minutos (1 hora) e outro de 300 minutos (5 horas).

A quantidade de dados redundantes enviados é a característica principal deste gráfico. Nota-se que quando utiliza-se 5MB de tamanho de buffer sem usar o protocolo de transporte, a quantidade máxima de dados recebidos ficou sempre abaixo da linha dos 60%. Por outro lado, quando utilizou-se o protocolo de transporte e foram enviados 700% de informação (equivalente a enviar 7 vezes os dados originais) a quantidade de dados recebidos sempre ficou acima da

linha dos 95%. Isso significa que em muitas das 100 simulações, a informação completa seria recuperada. Porém, 700% de informação enviada sugere que o overhead gerado na rede seria muito grande (cerca de 6 vezes maior, visto que 700% de informação enviada na rede significa 7 vezes o envio da mesma informação que sem o uso do protocolo seria enviado apenas 1 vez). Mas fazendo uma breve reflexão a respeito ao reduzir o TTL de 300 para 60 minutos está sendo reduzido o tempo que as mensagens ocupam a rede em 80%, ou seja, está sendo usado apenas 20% ($\frac{60}{300} = 0.2$) da rede que seria usada com um TTL de 300 minutos. Então, conclui-se que seria possível enviar 500% (porque $\frac{300}{60} = 5$) de informação obtendo o mesmo uso da rede (pois os pacotes expirarão antes com um TTL de 20%). Portanto, estão sendo gerados 20% dos 200% restantes de informação redundante (pois são enviados 700% de informação no total, onde destas 500% não modificam o uso da rede devido ao TTL reduzido e sobram os outros 200% de informação). Por isso, quando envia-se 600% de informação redundante (perfazendo 700% de informação no total, sendo 100% da informação original e 600% de redundância) o overhead é de 40% (ou seja, $700\% \times 20\% = 140\%$) do que o que era gerado quando o TTL era 300 minutos e o protocolo de transporte não havia sido utilizado. Com tudo, foram obtidas melhores taxas de entrega com o uso do PTTA.

Para garantir que o protocolo de transporte entregaria sempre toda a informação original, foi mantido o envio de dados redundantes na rede. É possível observar que quando foram enviados 1100% de dados redundantes, é possível dizer que nas 100 simulações os dados foram recuperados integralmente. Isso significa que tem-se um overhead de 120% na rede, mas em todas as simulações o nó de destino receberá 100% a informação original.

A Figura 16 representa o desempenho onde os nós possuem 6MB de tamanho de buffer. Novamente os dois valores de TTL foram testados e quantidades diferentes de dados redundantes enviados. É possível notar que com 6MB de buffer e sem o uso do protocolo de transporte a quantidade máxima de dados recebidos foi 65% para o TTL de 300 minutos e 61% para o de 60 minutos. Porém, quando se utiliza o protocolo de transporte e envia-se 500% de dados redundantes (um total de 600% de dados porque 100% é a quantidade real de dados mais 500% de redundância), a quantidade de dados recebidas ficou sempre superior à linha dos 95%. Isso mostra que várias das 100 simulações tiverem a totalidade dos dados recuperados. Então, fazendo a mesma análise realizada para o caso dos 5MB de buffer, quando envia-se 500% de dados redundantes com o TTL igual a 60 minutos, o overhead é de apenas 20% e são obtidas melhores taxas de entrega. Para garantir que o protocolo de transporte entregasse a totalidade dos dados em todas as simulações realizadas, foi mantido o envio de dados redundantes na rede. Após isso, foi possível notar que quando são gerados 1100% de dados redundantes é possível dizer que em todas as 100 simulações foram recuperadas a informação completa. Isso significa

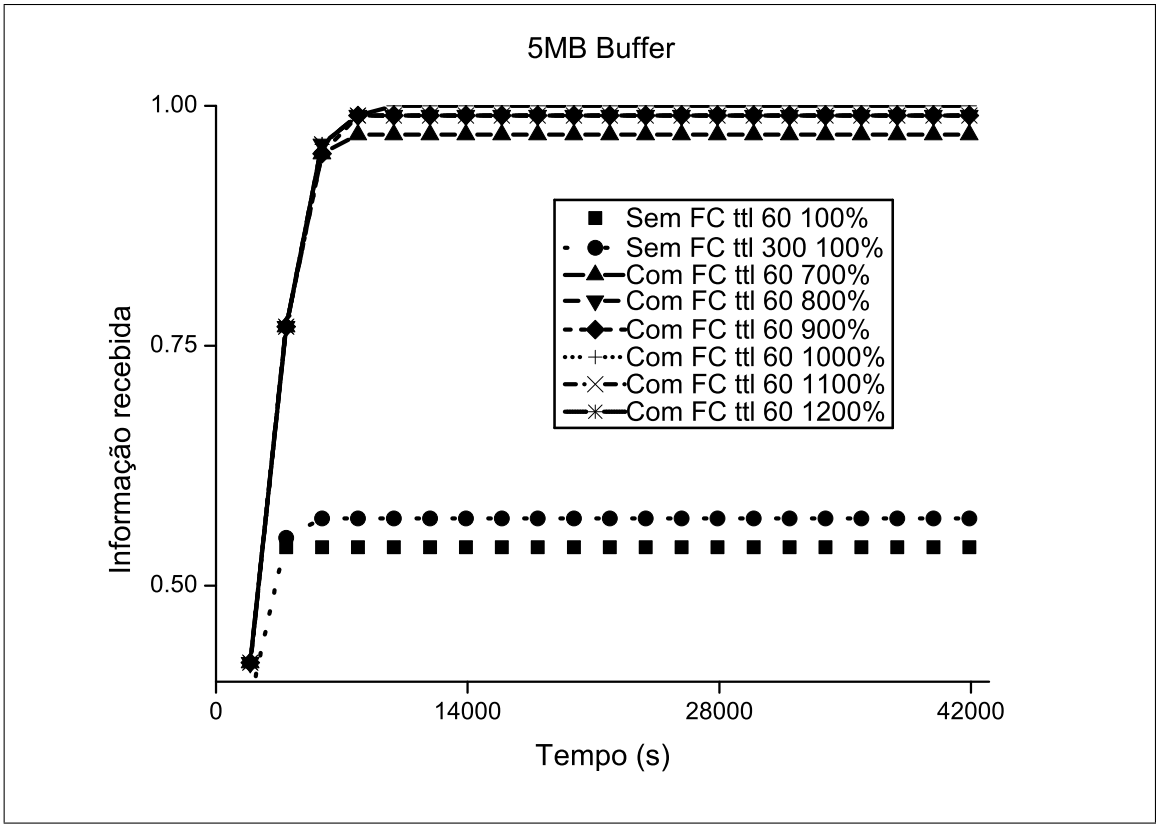

Figura 15: Quantidade de informações recebidas por tempo de simulação com buffer de 5MB.

que tendo o mesmo overhead do obtido no caso do buffer com 5MB (120%) agora 100% de informação pode ser recuperada em todas as 100 simulações.

A Figura 17 exibe o desempenho na qual os nós tinham 7MB de tamanho de buffer. De novo, os dois TTL foram testados. Para os buffers de 7MB sem o uso do protocolo de transporte a máxima quantidade de dados recebidos foi 73% para o TTL de 300 minutos e de 68% para o de 60 minutos. Todavia, usando o protocolo de transporte e enviando 500% de informação redundante (um total de 600% de dados porque 100% corresponde à informação original mais 500% de dados redundantes) a quantidade de dados recebidos ficou novamente acima da linha dos 95%. Isso significa que em várias das 100 simulações executadas a informação completa foi recuperada. Então, realizando a mesma análise feita para os casos anteriores (5MB e 6MB de buffer) quando se envia 500% de dados redundantes com o TTL igual a 60 minutos, se está gerando apenas 20% de overhead a mais do que aquele gerado no caso dos 300 minutos de TTL sem o protocolo de transporte e ainda se obtém melhores taxas de entrega. Para garantir que o protocolo de transporte sempre conseguisse recuperar todos os dados neste cenário, foi mantido o envio de informações redundantes na rede. Após foi possível observar que quando gerados 800% de dados redundantes todas as 100 simulações conseguiram recuperar todo o dado original. Isso significa que com um overhead de 80% na rede foram entregues 100% dos

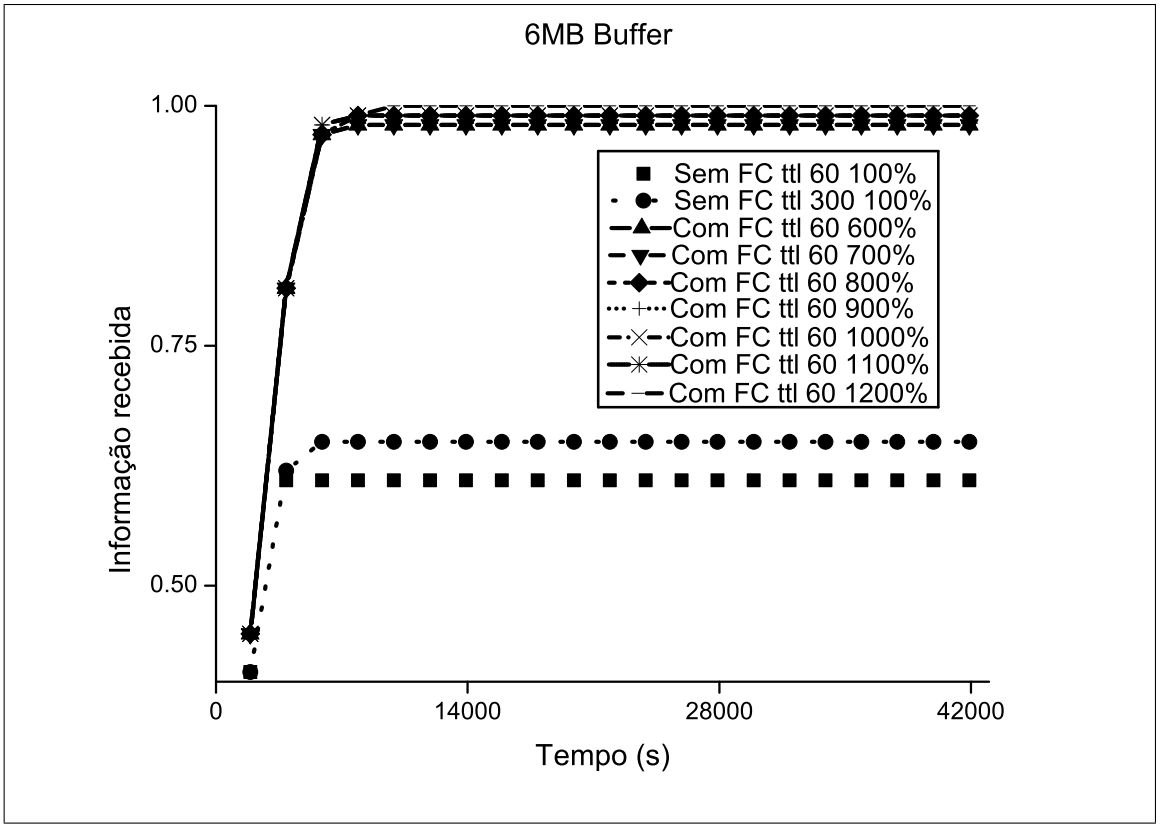

Figura 16: Quantidade de informações recebidas por tempo de simulação com buffer de 6MB.

dados no nó de destino em todas as 100 simulações.

Na Figura 18 se tem o desempenho da rede onde os nós possuem 8MB de tamanho de buffer, com exceção dos tramways que tem tinham 50MB. Foram testados os dois TTLs. Nesta imagem percebe-se que quando se utilizou 8MB de tamanho de buffer sem o protocolo de transporte, o máximo de dados recebidos chegou a 81% para 300 minutos de TTL e 74% para o de 60 minutos. Enquanto ao se utilizar o protocolo de transporte e enviando 400% de informação redundante (um total de 500% de dados pois 100% representa a informação original mais 400% de redundância) a quantidade de dados recebida ficou sempre acima da linha dos 90%. Isso significando que em várias das 100 simulações foi recebida a totalidade dos dados. Portanto, realizando a mesma análise efetuada para o caso dos 5MB de buffer, ao se enviar 400% de dados redundantes com o TTL igual a 60 minutos, não está sendo gerado overhead, ou seja o uso da rede se mantém como ao transmitir 100% de informação com TTL de 300 minutos sem o protocolo de transporte e obtém-se melhores taxas de entrega. Para se obter sempre a recuperação total dos dados, foi mantido o envio de informações redundantes na rede e como mostra a Figura 18 gerando 800% de dados redundantes é possível dizer que em todas as 100 simulações os dados foram completamente recuperados. Isso significa que com um overhead de 80% na rede é possível dizer que nesta rede os nós conseguem recuperar toda a informação.

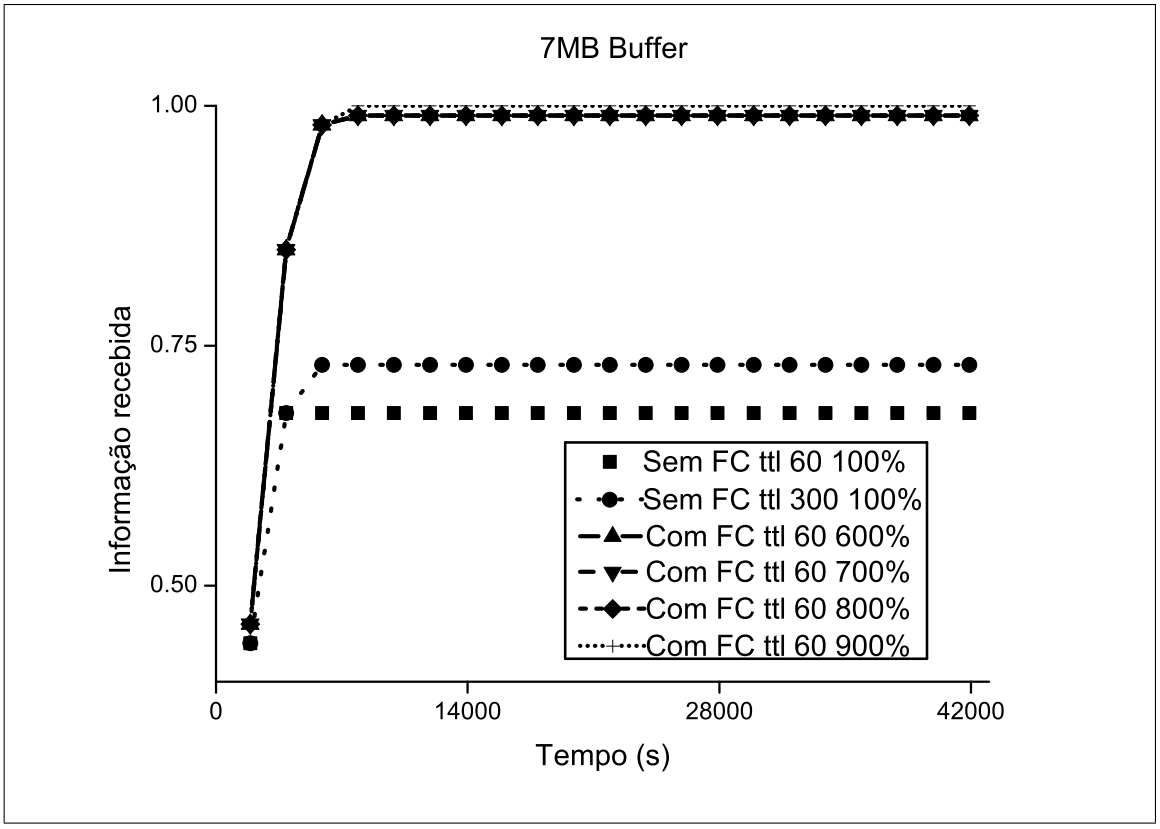

Figura 17: Quantidade de informações recebidas por tempo de simulação com buffer de 7MB.

Foi realizado o mesmo para os nós com tamanho de buffer de 9MB e o resultado é exibido na Figura 19. Mais uma vez, os dois TTLs foram testados. Agora utilizando buffers de 9MB de tamanho e sem o protocolo de transporte, a maior quantidade de dados recebida foi de 92% para mensagens com TTL de 300 minutos e 80% para o de 60 minutos. Por outro lado, ao utilizar o protocolo de transporte e enviando 400% de informações redundantes (um total de 500% de dados, pois 100% são as informações reais e mais 400% de informações redundantes), a quantidade de informações recebidas, na média, foi de 99%. Isso significa que em quase todas as 100 simulações a informação completa pode ser recuperada. Então ao se realizar a mesma análise realizada para os outros tamanhos de buffer, quando envia-se 400% de dados redundantes com TTL igual a 60 minutos, não estão sendo gerado overhead algum excedendo o uso da rede na transmissão sem o protocolo de transporte e com TTL de 300 minutos e ainda assim o protocolo de transporte consegue melhores taxas de entrega. Para garantir que o protocolo de transporte pudesse recuperar 100% das informações em todas as simulações realizadas, foi mantido o envio de informações redundantes. E, agora, é possível observar que com o envio de 500% de informação redundante é possível dizer que em todas as 100 simulações os dados foram completamente recuperados, isso significa que com um overhead de 20% se obteve êxito na totalidade dos experimentos.

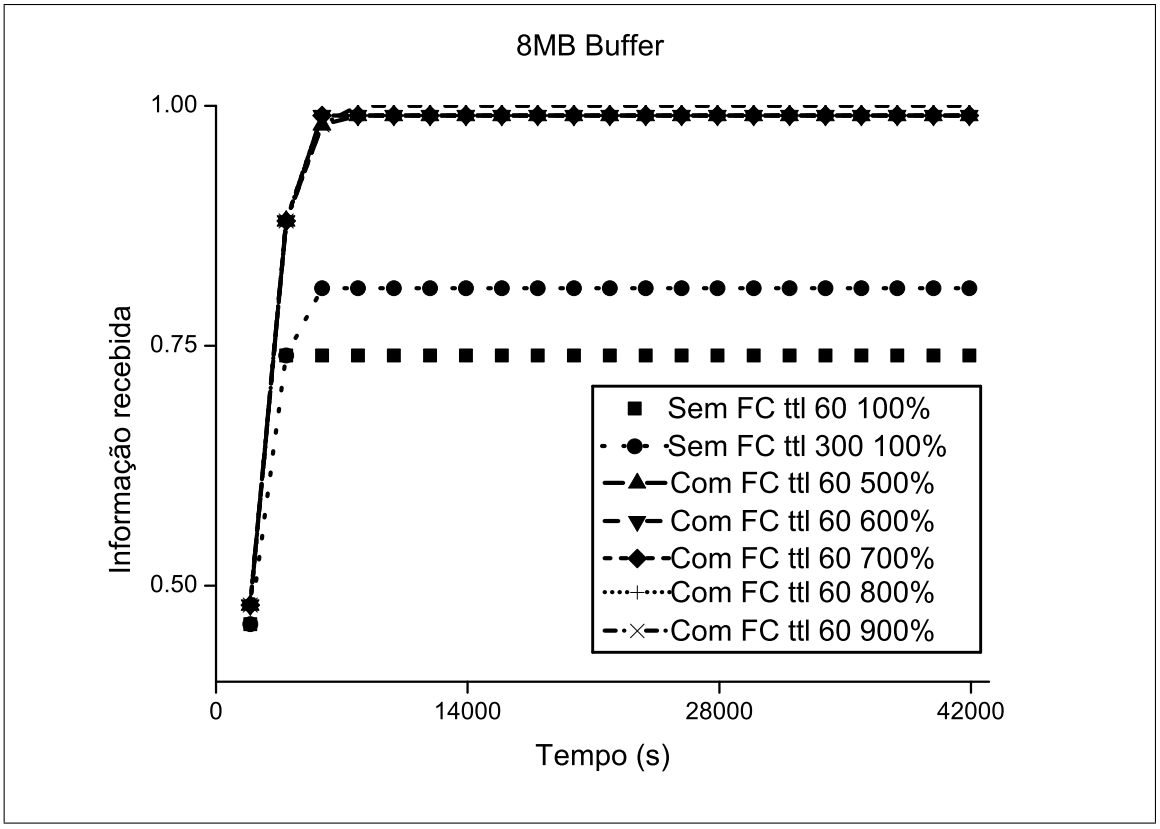

Figura 18: Quantidade de informações recebidas por tempo de simulação com buffer de 8MB.

Das figuras anteriores é possível observar uma relação entre a quantidade de dados redundantes enviados, com o tamanho dos buffers para se atingir 100% de dados recuperados em todas as simulações e a quantidade de overhead gerado comparado com o das mensagens com TTL de 300 minutos. O resumo desta relação é exibida na tabela 5.

Tabela 5: Quantidade de informação redundante necessária para cada tamanho de buffer e quantidade de overhead gerado comparado com o cenário utilizando TTL a 300 minutos.

Tamanho de Buffer	Informação redundante	Overhead Gerado
5MB	1100%	120%
6MB	1100%	120%
7MB	800%	80%
8MB	800%	80%
9MB	500%	20%

Em todos os casos testados os resultados com o uso do protocolo de transporte obteve melhor desempenho que aqueles onde o protocolo não foi utilizado. Isso implica que com as técnicas de códigos fontanais uma maior eficiência na entrega das informações é obtida. Isso se dá porque, quando se utiliza o protocolo descrito, não importam quais os pacotes recebidos para realizar a decodificação da informação, a hora em que o destino receber quantidade suficiente de

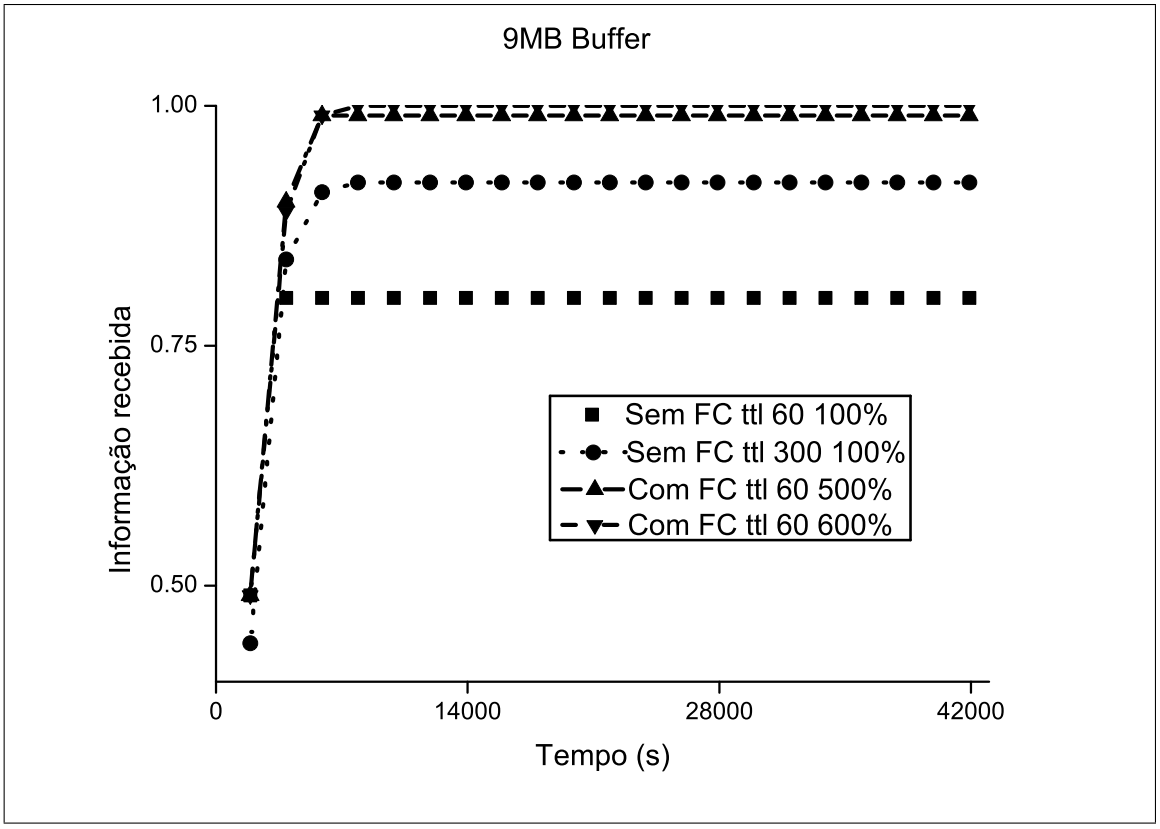

Figura 19: Quantidade de informações recebidas por tempo de simulação com buffer de 9MB.

segmentos, a informação original poderá ser recuperada. Por outro lado, quando o protocolo de transporte não é utilizado, a informação original só poderá ser completamente recebida se todas as partes da informação segmentada forem recebidas. Mas, pode ocorrer de algum fragmento ficar "preso" em alguma área da rede que venha a ficar desconexa e a informação pode nunca chegar ao destino.

É importante notar que de acordo com a tabela 5, o máximo de informações redundantes que foi necessário enviar para se ter uma taxa de entrega de 100% em todas as simulações efetuadas, foi de 1100%, lembrando que 100% é a quantidade de mensagens originais, então 1100% significa que foram enviadas 10 vezes a quantidade de dados originais, esta quantidade de 10 vezes será importante para o controle da fonte que será explicado adiante.

Os resultados até esta parte do cenário 1, foram publicados em (ALBINI et al., 2011a).

4.1.2 CENÁRIO 2 (REDE COMPARTILHADA)

Além do cenário de Helsinki, foi utilizado um outro cenário chamado aqui RWP (Random WayPoint), devido ao padrão de movimentação dos nós. Este cenário conta com a mesma área do cenário 1, porém sem delimitação de ruas e caminhos permitidos, e com a mesma quan-

tidade de nós na rede e com a velocidade sendo selecionada aleatoriamente entre 0,5 e 1,5 m/s e sem pausa entre os pontos aleatórios selecionados para a movimentação.

Ao mesmo tempo em que foi utilizado este novo cenário, também foi inserido tráfego concorrente, onde outros nós da rede enviavam dados ao mesmo tempo em que o origem e destino estavam tentando se comunicar com o uso do protocolo de transporte proposto. Além disso, foi testado o comportamento do protocolo de transporte utilizando-se diversos protocolos de roteamento. O intuito destes testes com os diversos protocolos de roteamento é verificar a dependência, se é que ela existe, entre o protocolo de transporte e os protocolos de roteamento utilizados.

Para que a comparação fosse válida, os mesmos testes realizados no cenário 2, foram realizados utilizando o cenário 1.

Agora os testes foram divididos em três fases.

Primeiramente foi realizada a transferência dos dados sem o uso do protocolo de transporte, também chamado sem PTTA, ou seja, as informações foram segmentadas e enviadas pela rede utilizando-se dos protocolos de roteamento Epidêmico, Prophet, Maxprop, Spray and wait e First contact (cada fragmento foi enviado uma única vez, com TTL - *Time To Live* igual a 300 minutos).

Em seguida foi efetuada a transferência com o uso do protocolo de transporte proposto (PTTA), onde as informações foram segmentadas, codificadas e em seguida enviadas através da rede. Nesta parte, primeiro foram enviadas as informações sem fechamento da fonte (geração contínua e infinita de dados codificados, como utilizado no cenário 1) e utilizando todos os mesmos protocolos de roteamento citados anteriormente.

E por último, foi feita uma análise dos pacotes recebidos e foi realizada a simulação com fechamento da fonte de uma maneira ideal, que necessitaria conhecer o comportamento da rede para que se fechasse a fonte quando o último pacote necessário para a decodificação de todos os dados enviados fosse gerado chamado de PTTA ideal. A partir daí foi observado que a quantidade de informação redundante trafegando na rede, aumentava muito a quantidade de mensagens na rede, o que ocasiona desperdício de recursos, pois as mensagens eram geradas continuamente. Por isso, foram refeitas estas simulações com um TTL 10 vezes menor (30 minutos) que o valor utilizado no caso onde o protocolo de transporte não era utilizado (que era de 300 minutos). Este valor 10 vezes menor foi escolhido por se observar que utilizando o PTTA, era necessário enviar cerca de 10 vezes mais pacotes na rede para se obter a totalidade da informação decodificada (vide tabela 5). Assim, diminuindo o tempo que cada pacote ficava

na rede, estaria economizando os recursos da rede de uma maneira simplista.

Após as simulações com o TTL de 30 minutos, foi analisada a quantidade de informações na rede, e foi observado que apenas na média os recursos seriam economizados proporcionalmente, mas nos primeiros instantes de tempo, o uso dos recursos ficaria maior do que no caso onde não se utilizasse o protocolo de transporte. Por isso foi adotada uma política para estabilizar o uso da rede e este ficar o mais parecido possível com aquele que é necessário para o envio das informações sem o uso do PTTA. Esta política foi chamada de "PTTA" CF - Controle da Fonte. Para isso foi usada a equação (36) a fim de encontrar o intervalo de geração das mensagens e a equação (37) para encontrar o tempo que a fonte de códigos fontanais na camada de transporte fica ativa.

$$\begin{aligned} \psi_p &= 30min \times 60 = 1800s \\ M &= 20 \\ \lambda_{PTTA} &= \frac{\psi_p}{M} \\ \lambda_{PTTA} &= \frac{1800}{20} \\ \lambda_{PTTA} &= 90s \end{aligned}$$

$$\begin{aligned} \psi_r &= 300min \times 60 = 18000s \\ \lambda_r &= 25s \\ t &= \psi_r - (M \times \lambda_r) = 18000 - (20 \times 25) = 17500s \end{aligned}$$

Em todas as simulações foram utilizados os dois cenários descritos anteriormente. Em todos os testes os parâmetros de entrada eram iguais com relação às configurações da rede, ou seja, ambos possuem o mesmo tamanho, as mesmas sementes aleatórias, o mesmo número de nós, entre outros.

Os resultados obtidos das simulações com o protocolo epidêmico são exibidos na Figura 20.

Na Figura 20 é possível observar que no cenário *Random WayPoint* utilizando-se o protocolo de roteamento Epidêmico e enviando as informações sem o uso do PTTA, a quantidade de informação recebida foi insignificante, pois a média com intervalo de confiança de 95% ficou em zero, mostrando que quase na totalidade das 100 simulações não foram recebidas as informações enviadas. Ao contrário quando se utilizou o PTTA, a taxa média de entrega

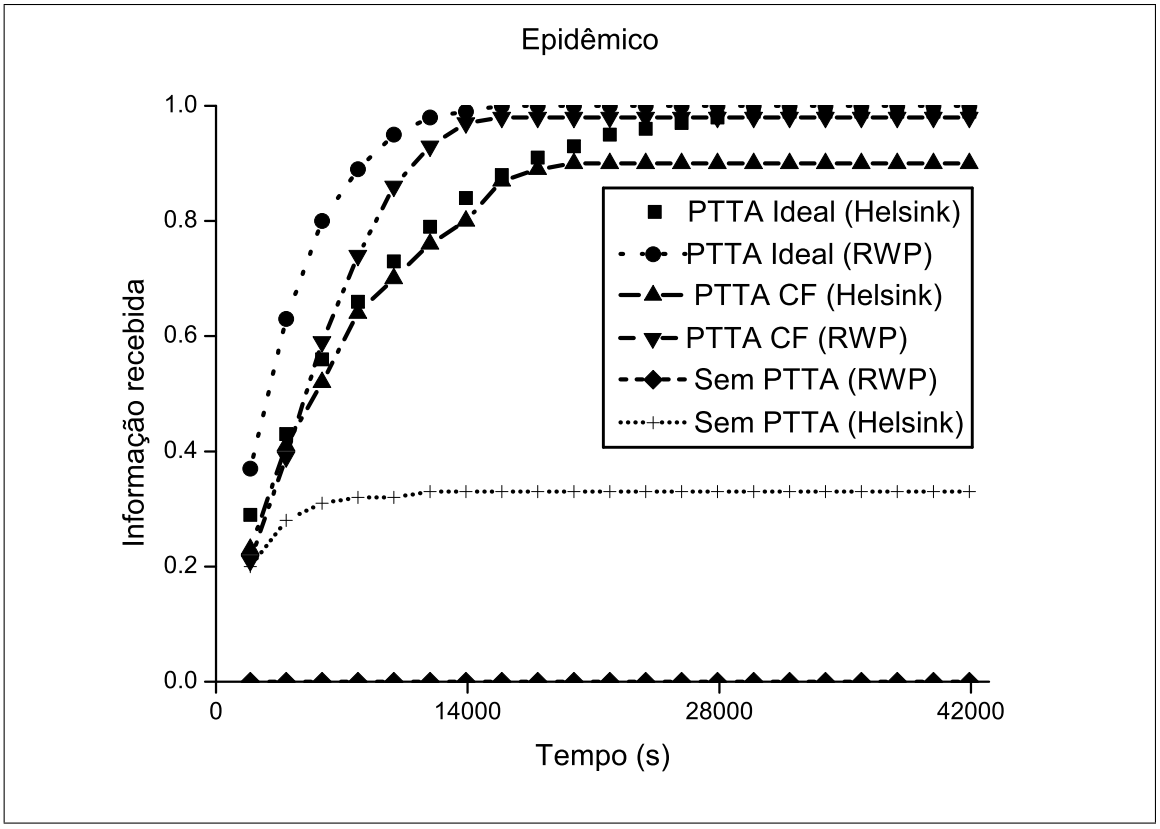

Figura 20: Sucesso de entrega utilizando o protocolo de roteamento epidêmico.

foi a melhor, dentre os casos comparados nesta figura, como pode-se observar. Além disso, no cenário de Helsinki foi atingida uma taxa de entrega de cerca de 90% no pior dos casos com o protocolo de transporte, contra cerca de 30% sem o PTTA (ganho de 60%).

A Figura 21 mostra que utilizando-se o protocolo de roteamento Maxprop e enviando as informações sem o uso do PTTA, foi obtida a melhor taxa de entrega para as informações dentre todas as simulações sem o PTTA. Porém, mesmo este resultado não foi superior aos resultados com o uso do PTTA. Quando utiliza-se o protocolo de transporte, a taxa média de entrega torna-se muito melhor como pode-se observar pela comparação dos gráficos na Figura 21. Além disso, no cenário de Helsinki foi conseguido uma taxa de entrega de pouco mais de 90% no pior dos casos com o protocolo de transporte, contra cerca de 55% sem ele (ganho de 35%). Outro fato interessante neste gráfico é a ordem das linhas sem o uso dos FC, onde o pior dos casos foi para o cenário de Helsinki e o melhor foi para o cenário RWP. Este foi o único caso onde Helsinki perdeu para o RWP, nos outros foi verificada a ordem inversa.

Na Figura 22 verifica-se que com o protocolo de roteamento Prophet e enviando as informações sem o uso do PTTA, a quantidade de informação recebida fica em torno dos 30% para o cenário RWP e em torno dos 40% para o cenário de Helsinki. Porém, quando se utilizou o PTTA, a taxa média de entrega foi próxima de 100% para o cenário RWP e em torno de 90%

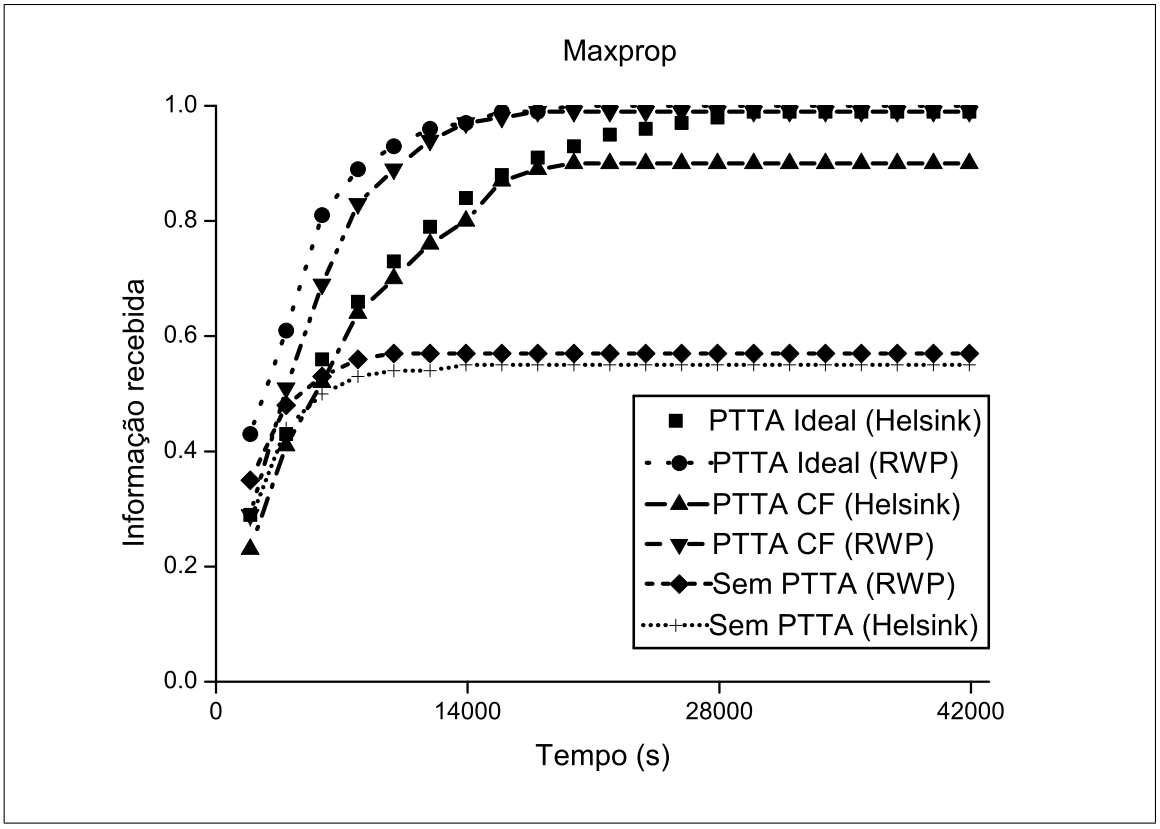

Figura 21: Sucesso de entrega utilizando o protocolo de roteamento Maxprop.

para o cenário de Helsinki. Além disso, no cenário RWP foi obtida uma diferença na taxa de entrega de cerca de 70% a mais do que os casos sem o PTTA, e cerca de 50% para o cenário de Helsinki.

A Figura 23 evidencia o comportamento das informações na rede quando se utiliza o protocolo de roteamento *Spray and Wait*. Nesta figura é muito evidente a inversão ocorrida nos casos sem FC entre os dois cenários, RWP e Helsinki. É perceptível que no começo o *Random WayPoint* sai na frente, mas perto dos 8000 segundos ocorre a inversão e as simulações de Helsinki se sobressaem. A explicação deste fato é dada pelo comportamento da rede em ambos os casos. Como se tem o TTL estipulado em 300 minutos, para os casos sem PTTA, o tamanho dos *buffers* são iguais mas o padrão de encontros não, ao se analisar o comportamento da rede, nota-se que como se tem mais encontros no cenário RWP do que no Helsinki, os *buffers* dos nós logo são preenchidos, o que não ocorre facilmente no cenário de Helsinki, por ocorrerem encontros menos frequentes. Por isso no começo o RWP obtém uma taxa de entrega superior ao cenário de Helsinki, pois as mensagens propagam-se rapidamente, mas após algum tempo, quando os *buffers* estão cheios e começam a ocorrer descartes o quadro se inverte, pois descartam-se mensagens que ainda não foram recebidas e não estão na rede.

Na Figura 24 é visível o impacto do número de encontros entre os nós ao se utilizar

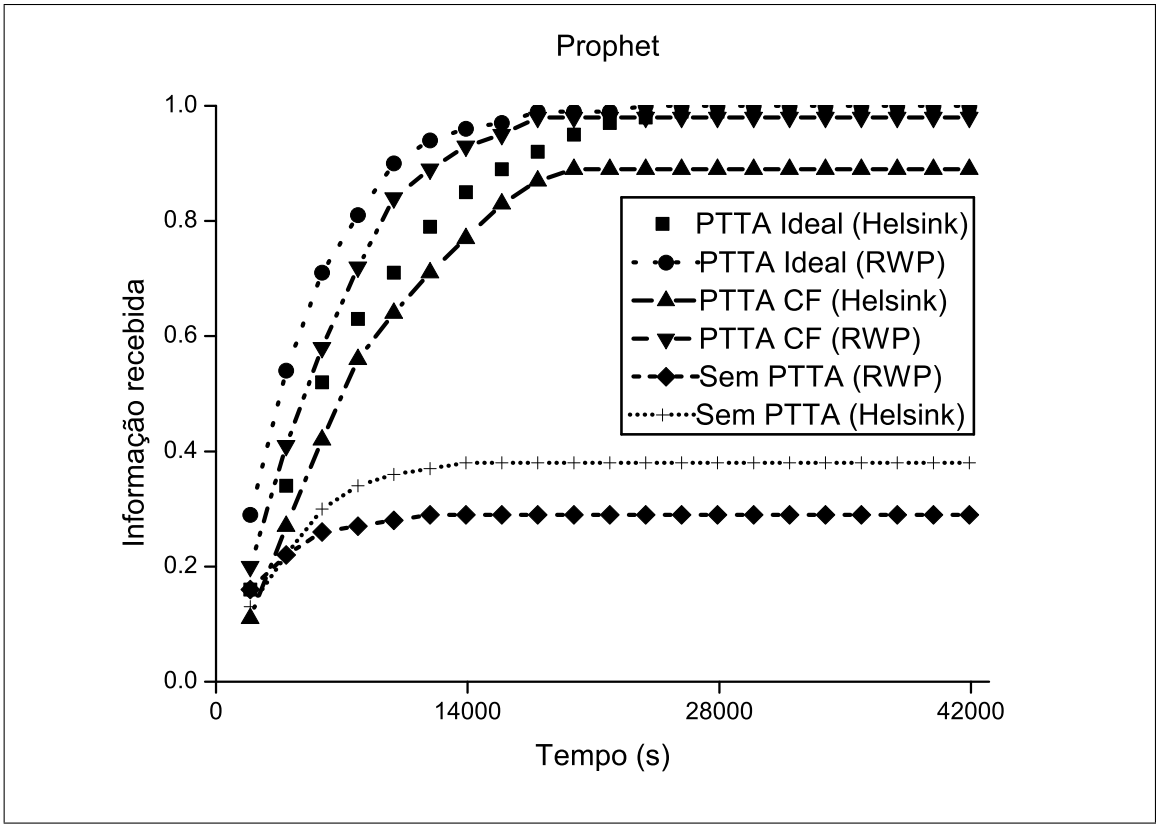

Figura 22: Sucesso de entrega utilizando o protocolo de roteamento Prophet.

o protocolo twohops. Como no cenário RWP a quantidade de encontros entre os nós é maior, devido a inexistência de limitações como ruas e calçadas, a taxa de entrega com o PTTA com o controle da fonte quase atingiu 100%, enquanto quando não se utilizou o protocolo ficou próximo dos 30%. Nos gráficos referentes a Helsinki é possível observar que a propagação da mensagem tornou-se mais lenta, isso porque o número de encontros entre os nós é baixo. Além disso, é possível se fazer o paralelo com o resultado matemático obtido, porém como este refere-se ao momento em que todos os nós recebem todas as mensagens, ele é um pouco defasado com relação ao obtido em simulação, onde apenas o destino é monitorado.

Como é possível verificar em todos os resultados até agora, com o PTTA e com o controle da fonte (CF) proposto, foi possível obter uma boa taxa na entrega dos dados, próximo ao ideal. Porém, não foi o ideal porque, para este fim, seria necessário um canal de retorno, para que o controle fosse feito exatamente no momento em que não fossem mais necessários pacotes para o destino decodificar os dados. Como a proposta é exatamente para que este canal de retorno não exista, observa-se que os resultados foram muito promissores. Para comprovar isso, basta verificar o uso dos recursos da rede, através do tráfego na rede.

Na Figura 25 está sendo exibido o número de pacotes diferentes na rede por tempo de simulação. É muito importante observar que o uso dos recursos da rede com o controle de

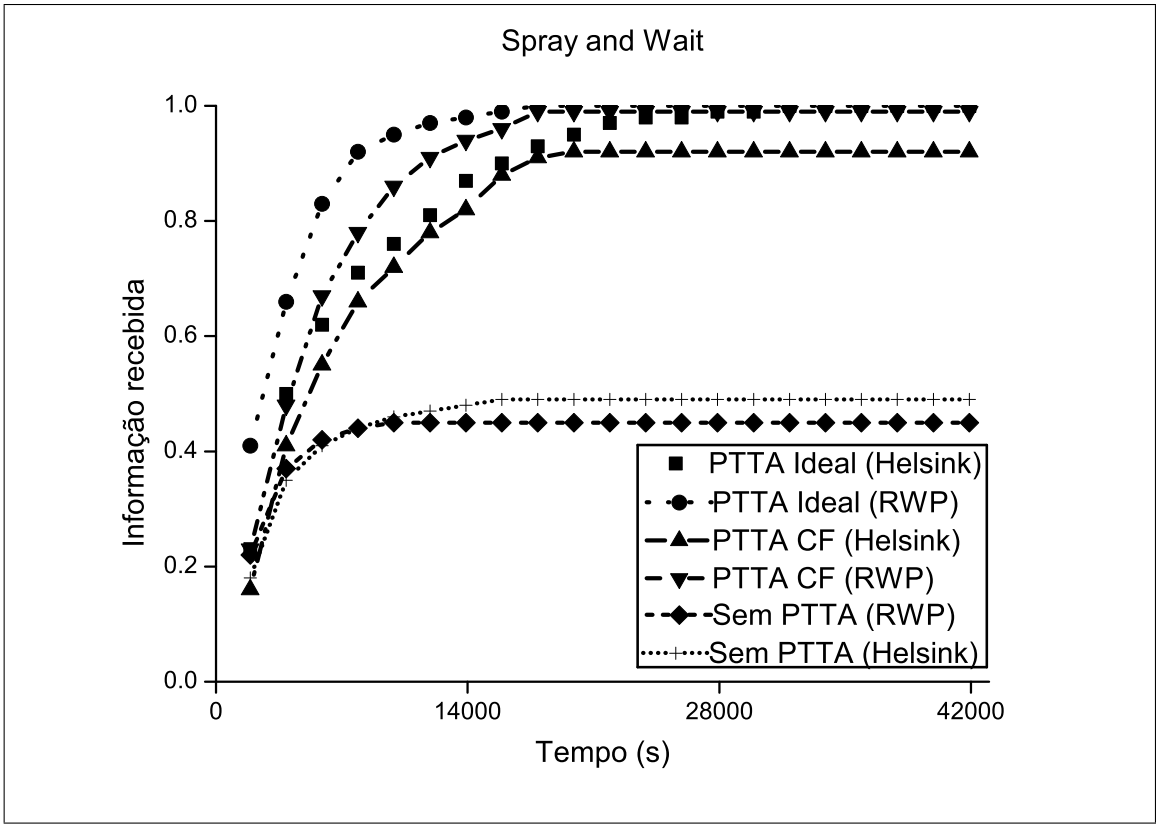

Figura 23: Sucesso de entrega utilizando o protocolo de roteamento Spray and Wait.

geração das mensagens proposto, se deu de maneira equilibrada. Isso é afirmado ao se comparar com o uso da rede nos casos onde o PTTA não era utilizado. Através dessa comparação é possível verificar que a área dos dois gráficos são muito próximas, e para os casos onde foi utilizado o controle da fonte, a área é ainda menor do que no caso onde foram enviadas as mensagens sem o PTTA, o que prova que o uso da rede foi menor e os resultados, como mostrados anteriormente, obtiveram taxas de entrega maiores.

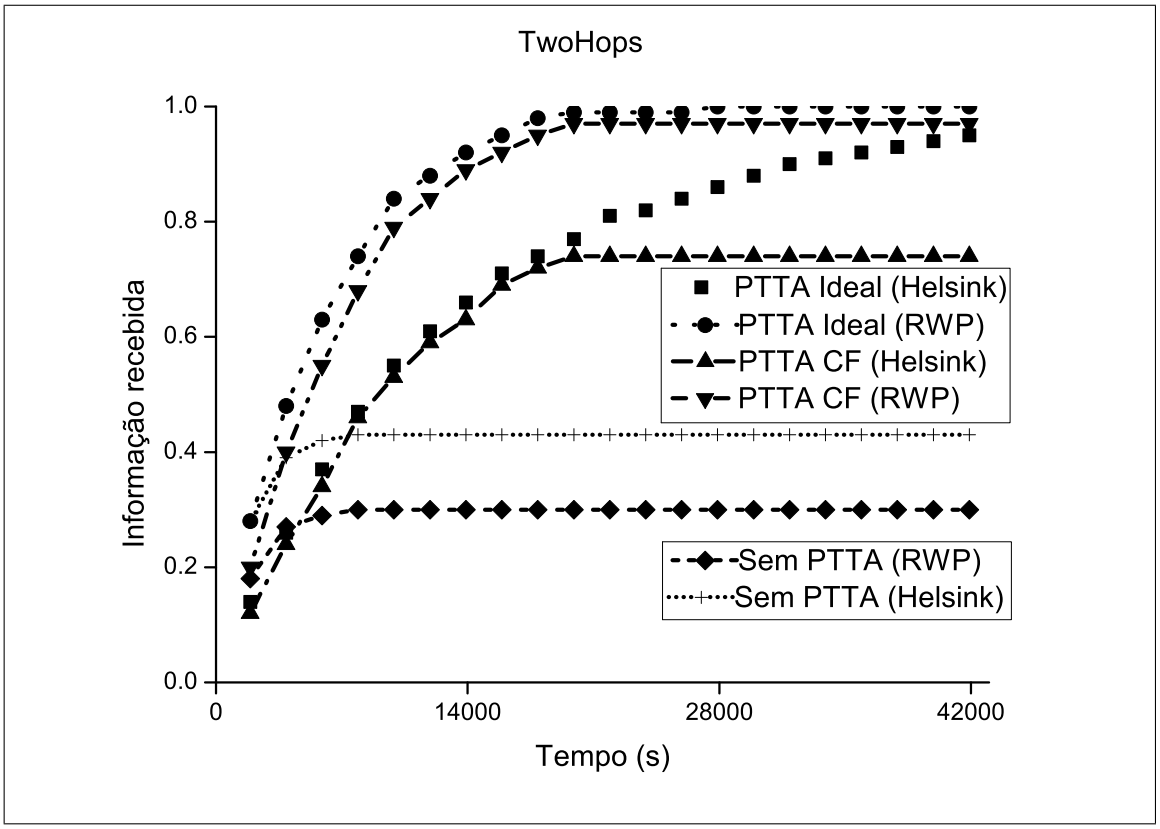

Figura 24: Sucesso de entrega utilizando o protocolo de roteamento Twohops.

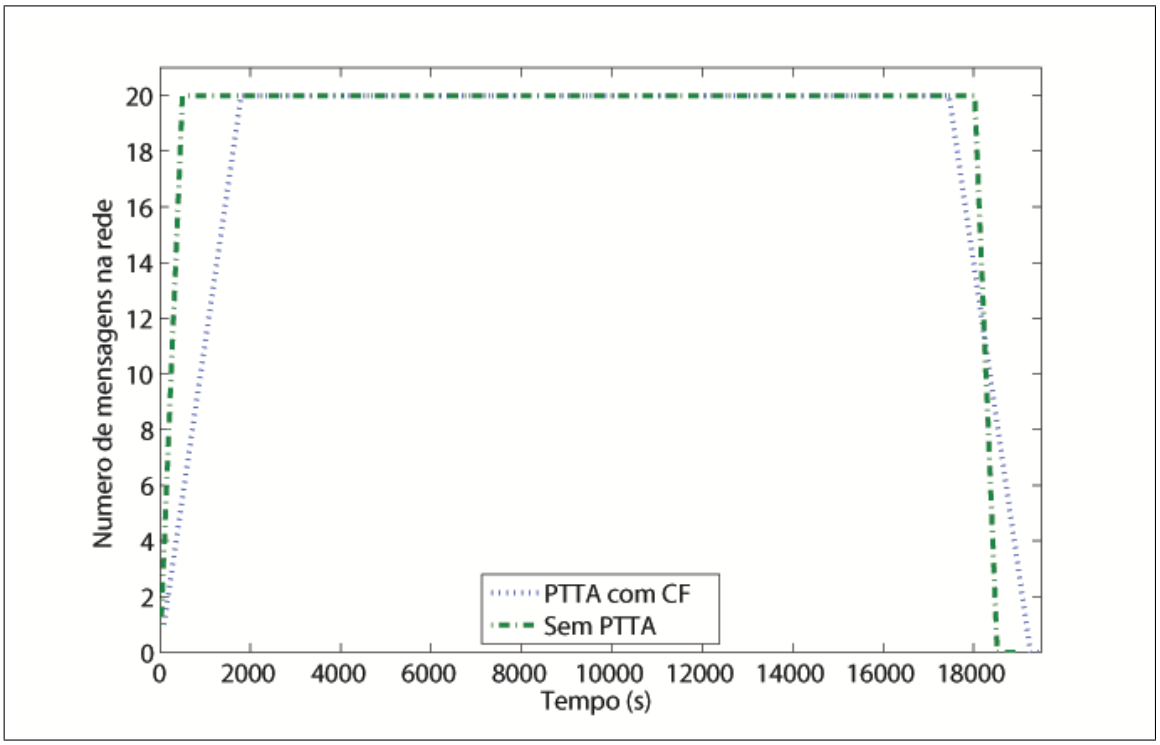

Figura 25: Quantidade de pacotes diferentes do fluxo na rede por tempo de simulação.

5 CONCLUSÕES

Na presente tese foi apresentado um novo protocolo de transporte para redes tolerantes a atrasos e desconexões a fim de viabilizar a distribuição de conteúdo neste cenário de rede oportunista. O protocolo proposto é chamado PTTA (Protocolo de Transporte Tolerante a Atrasos e desconexões) e se utiliza das técnicas dos códigos fontanais para aumentar a taxa de entrega das informações sem o uso de conexão e confirmação no recebimento das informações. A principal motivação para o uso de códigos fontanais foi para alcançar uma otimização nas técnicas de encaminhamento das mensagens. Ainda neste trabalho foi levado em consideração o tamanho dos buffers, o TTL das mensagens e a quantidade de informações redundantes geradas na rede.

Além disso, o trabalho detalhado neste documento descreveu e avaliou um método para o controle da fonte do PTTA. A proposta baseia-se no uso da rede e no controle da geração das mensagens codificadas, usando para isso o TTL, o intervalo de geração das mensagens e a diversidade de informações desejadas, que é basicamente a quantidade de redundância gerada.

As simulações realizadas através do simulador (The ONE) mostraram que o uso do protocolo demonstra vantagens no recebimento dos dados diminuindo o tempo de entrega. Ainda foi possível concluir que, de acordo com os testes realizados, os benefícios são maiores conforme o volume dos dados a serem transmitidos aumenta. As principais vantagens no uso do protocolo proposto são: a não necessidade de confirmação; a ausência de retransmissões de mensagens; a flexibilidade no controle da geração das informações codificadas, e; a necessidade de apenas 5% de informação redundante recebida para se realizar a decodificação.

Assim, baseado nos resultados das simulações apresentados, o protocolo descrito provou que pode alcançar melhores taxas de entrega com um melhor desempenho geral da rede. Os resultados demonstraram que através do conhecimento da rede, os valores obtidos podem alcançar taxas cada vez melhores.

Por conseguinte, foi demonstrado que através do controle da fonte proposto, se obtém um melhor aproveitamento dos recursos da rede, não gerando desperdício e nem custo adicional na transmissão dos dados, enquanto a probabilidade de entrega torna-se significativamente

maior.

Os resultados apresentados avaliaram situações próximas às reais, onde existe tráfego concorrente na rede, o que gera uma maior carga de mensagens trafegando nos nós intermediários e consequentemente o preenchimento de seus buffers. Isso prova que em situações normais o PTTA consegue mostrar sua eficiência na transmissão dos dados. Ainda é importante ressaltar que foram realizados testes com diversos protocolos de roteamento, e que em todos os casos o PTTA mostrou-se superior à transmissão dos dados se comparado com qualquer outro protocolo deste gênero (sem o uso do PTTA), demonstrando assim a sua independência em relação às camadas inferiores.

5.1 PRINCIPAIS RESULTADOS OBTIDOS

Como principais resultados obtidos com a finalização desta tese tem-se:

- Um novo protocolo de transporte para DTN que melhora a distribuição de conteúdo nesse tipo de rede oportunista, através da utilização de códigos fontanais para aumentar a taxa de entrega das mensagens sem que seja necessário a utilização de confirmação das mensagens enviadas;
- Um módulo de controle da fonte dos códigos fontanais para mitigar a geração das mensagens codificadas através do intervalo de geração destas e da diversidade de informações desejada;
- A publicação do artigo: De Pellegrini, F; El-Azouzi, R; Albini, F., "Interplay of Contact Times, Fragmentation and Coding in DTNs", in the Proceedings of WiOPT'13, Tsukuba, Japan;
- A disseminação dos resultados obtidos através da publicação de (i) artigo científico em revisa especializada na área de redes sem fio (Wireless Networks); (ii) artigo científico em congresso internacional na área de redes (GIIS 2011); (iii) artigo científico em congresso nacional na área de Telecomunicações (SBrT 2011).

5.2 TRABALHOS FUTUROS

Os resultados mostram que o uso dos códigos fontanais é uma alternativa interessante para solucionar o problema da não existência de confiabilidade em DTNs. Como trabalhos futu-

ros poderiam ser realizados novos modelos matemáticos para que, com base nesses, o controle da fonte do protocolo de transporte fosse aprimorado.

Além disso, novas métricas poderiam ser utilizadas para calcular a taxa de encontro dos nós e, baseado nisso, controlar a geração das mensagens. Ou ainda, integrar técnicas utilizadas em protocolos de roteamento na escolha do melhor nó intermediário para o encaminhamento das mensagens, no protocolo de transporte. Por exemplo, poderia ser utilizada uma métrica como a proposta em (KALANTARI; LA, 2008) (a "temperatura" do nó origem) e baseado nesta decidir o tempo necessário para a transmissão ocorrer, visto que este parâmetro representa o número de contatos com outros nós da rede. Outra possibilidade seria seguir a sugestão descrita em (SAMUEL et al., 2009), onde é proposta a criação de alguns nós intermediários chamados "super nós" que funcionariam como uma nova fonte de informações codificadas, as quais poderiam ser muito úteis na disseminação em broadcast.

Ainda como trabalhos futuros, sugere-se desenvolver aplicações para dispositivos móveis reais, os quais utilizem o protocolo e assim verificar a viabilidade da exploração comercial do uso deste protocolo em aplicações móveis. Além disso, verificar o possível uso em redes colaborativas, as quais seriam públicas e gratuitas gerando economia e solucionando a comunicação inclusive em áreas remotas.

5.3 PUBLICAÇÕES

Este trabalho foi publicado em:

ALBINI, F. ; MUNARETTO, A. ; FONSECA, M. . Delay tolerant transport protocol - DTTP. In: Global Information Infrastructure Symposium (GIIS), 2011, 2011, Da Nang. IEEE Global Information Infrastructure Symposium (GIIS 2011), 2011. p. 1-6.

ALBINI, F. ; MUNARETTO, A. ; FONSECA, M. . PTTA: Protocolo de Transporte Tolerante a Atrasos. In: Simpósio Brasileiro de Telecomunicações (SBrT) 2011, 2011, Curitiba. XXIX Simpósio Brasileiro de Telecomunicações (SBrT) 2011. Rio de Janeiro: Sociedade Brasileira de Telecomunicações, 2011. p. 1-5.

DE PELLEGRINI, F. ; EL-AZOUZI, R. ; ALBINI, F. . Interplay of Contact Times, Fragmentation and Coding in DTNs, In: Proceedings of WiOPT'13, Tsukuba, Japan.

ALBINI, F. ; MUNARETTO, A. ; FONSECA, M. ; AMORIM, M. D. ; PELLEGRINI, F. . A blind mechanism to improve content distribution in delay/disruption tolerant networks. Wireless Networks, v. 19, p. 1-9, 2013. DOI: http://dx.doi.org/10.1007/s11276-013-0651-4

REFERÊNCIAS

ALBINI, F. L. P.; MUNARETTO, A.; FONSECA, M. Delay tolerant transport protocol - DTTP. **Global Information Infrastructure Symposium (GIIS), 2011**, IEEE, Da nang, Vietnam, v. 1, p. 1 – 6, August 2011. Disponível em: <http://dx.doi.org/10.1109/GIIS.2011.6026709>.

ALBINI, F. L. P.; MUNARETTO, A.; FONSECA, M. PTTA - protocolo de transporte tolerante a atrasos. **XXIX Simpósio Brasileiro de Telecomunicações - SBrT2011**, SBC, Curitiba, PR, Brazil, v. 1, p. 1 – 13, 2011.

ALTMAN, E.; De Pellegrini, F. Forward correction and fountain codes in delay-tolerant networks. **IEEE/ACM Transactions on Networking**, v. 19, n. 1, p. 1–13, February 2011.

ALTMAN, E.; PELLEGRINI, F. D. Forward correction and fountain codes in delay tolerant networks. In: **IEEE INFOCOM 2009 - The 28th Conference on Computer Communications**. IEEE, 2009. p. 1899–1907. ISBN 978-1-4244-3512-8. Disponível em: <http://dx.doi.org/10.1109/INFCOM.2009.5062111>.

BOICE, J.; GARCIA-LUNA-ACEVES, J. J.; OBRACZKA, K. Disruption-tolerant routing with scoped propagation of control information. In: **ICC**. [S.l.]: IEEE, 2007. p. 3114–3121.

BORTOLUSSI, L.; HILLSTON, J. Fluid model checking. **CoRR**, abs/1203.0920, 2012.

BRADEN, R. **Requirements for Internet Hosts - Communication Layers**. IETF, out. 1989. RFC 1122 (Standard). (Request for Comments, 1122). Updated by RFCs 1349, 4379, 5884, 6093, 6298. Disponível em: <http://www.ietf.org/rfc/rfc1122.txt>.

BURGESS, J. et al. Maxprop: Routing for vehicle-based disruption-tolerant networks. In: **In Proc. IEEE INFOCOM**. [S.l.: s.n.], 2006.

BURLEIGH, S. et al. Delay-tolerant networking: an approach to interplanetary internet. **IEEE Communications Magazine**, v. 41, n. 6, p. 128–136, 2003.

BYERS, J. W. et al. A digital fountain approach to reliable distribution of bulk data. In: **Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication**. New York, NY, USA: ACM, 1998. (SIGCOMM '98), p. 56–67. ISBN 1-58113-003-1. Disponível em: <http://doi.acm.org/10.1145/285237.285258>.

BYERS, J. W. et al. A digital fountain approach to reliable distribution of bulk data. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 28, n. 4, p. 56–67, out. 1998. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/285243.285258>.

CERF, V. et al. **Delay-Tolerant Networking Architecture**. IETF, abr. 2007. RFC 4838 (Informational). (Request for Comments, 4838). Disponível em: <http://www.ietf.org/rfc/rfc4838.txt>.

CHAHIN, W. et al. Blind online optimal forwarding in heterogeneous delay tolerant networks. In: **Wireless Days**. [S.l.]: IEEE, 2011. p. 1–6. ISBN 978-1-4577-2027-7.

CHEN, B. B.; CHAN, M. C. Architecture and incentive design of integrated cellular and disruption tolerant networks. In: **INFOCOM Workshops 2009, IEEE**. [S.l.: s.n.], 2009. p. 1 –2.

CHOI, B. J.; SHEN, X. Adaptive exponential beacon period protocol for power saving in delay tolerant networks. In: **Communications, 2009. ICC '09. IEEE International Conference on**. [S.l.: s.n.], 2009. p. 1 –6. ISSN 1938-1883.

CHOU, P. A.; WU, Y. **Network Coding for the Internet and Wireless Networks**. [S.l.], jun. 2007.

CHUAH, M.; XI, Y. Enhanced delivery in disruption tolerant networks using advantaged nodes with directional antenna capability. In: **Military Communications Conference, 2007. MILCOM 2007. IEEE**. [S.l.: s.n.], 2007. p. 1 –6.

CHUAH, M.; YANG, P.; XI, Y. How mobility models affect the design of network coding schemes for disruption tolerant networks. In: **ICDCSW '09: Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops**. Washington, DC, USA: IEEE Computer Society, 2009. p. 172–177. ISBN 978-0-7695-3660-6.

DAI, Y. et al. Cfp: Integration of fountain codes and optimal probabilistic forwarding in dtns. In: **GLOBECOM**. [S.l.]: IEEE, 2010. p. 1–5. ISBN 978-1-4244-5638-3.

DU, J.; KRANAKIS, E.; NAYAK, A. A geometric routing protocol in disruption tolerant network. In: **Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on**. [S.l.: s.n.], 2009. p. 109 –116. ISSN 1545-0678.

DVIR, A.; VASILAKOS, A. V. Backpressure-based routing protocol for dtns. In: KALYANARAMAN, S. et al. (Ed.). **SIGCOMM**. ACM, 2010. p. 405–406. ISBN 978-1-4503-0201-2. Disponível em: <http://dblp.uni-trier.de/db/conf/sigcomm/sigcomm2010.html#DvirV10>.

FALL, K. A delay-tolerant network architecture for challenged internets. In: **SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications**. New York, NY, USA: ACM, 2003. p. 27–34. ISBN 1-58113-735-4.

FALL, K. R.; FARRELL, S. Dtn: an architectural retrospective. **IEEE Journal on Selected Areas in Communications**, v. 26, n. 5, p. 828–836, 2008.

FRANÇOIS, J.-M.; LEDUC, G. Routing based on delivery distributions in predictable disruption tolerant networks. **Ad Hoc Netw.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 7, n. 1, p. 219–229, 2009. ISSN 1570-8705.

GONT, F.; BELLOVIN, S. **Defending against Sequence Number Attacks**. IETF, fev. 2012. RFC 6528 (Proposed Standard). (Request for Comments, 6528). Disponível em: <http://www.ietf.org/rfc/rfc6528.txt>.

GONT, F.; YOURTCHENKO, A. **On the Implementation of the TCP Urgent Mechanism**. IETF, jan. 2011. RFC 6093 (Proposed Standard). (Request for Comments, 6093). Disponível em: <http://www.ietf.org/rfc/rfc6093.txt>.

GROSSGLAUSER, M.; TSE, D. Mobility increases the capacity of ad-hoc wireless networks. In: **INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE**. [S.l.: s.n.], 2001. v. 3, p. 1360–1369 vol.3. ISSN 0743-166X.

GROSSGLAUSER, M.; TSE, D. Mobility increases the capacity of ad-hoc wireless networks. **IEEE/ACM Transactions on Networking**, v. 10, p. 477–486, 2002.

GUO, Z. et al. Adaptive routing in underwater delay/disruption tolerant sensor networks. In: **Wireless on Demand Network Systems and Services, 2008. WONS 2008. Fifth Annual Conference on**. [S.l.: s.n.], 2008. p. 31 –39.

HAYDEN, R.; STEFANEK, A.; BRADLEY, J. T. Fluid computation of passage time distributions in large Markov models. **Theoretical Computer Science**, v. 413, n. 1, p. 106–141, January 2012. Submitted to TCS November 2010. Accepted July 2011. Available online August 2011. Extended version of June 2009 technical report entitled "Fluid passage-time calculation in large Markov models". Disponível em: <http://pubs.doc.ic.ac.uk/fluid-passage-time/>.

HOLLIDAY, P. Nomad a mobile ad hoc and distruption tolerant routing protocol for tactical military networks. In: **ICDCSW '09: Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops**. Washington, DC, USA: IEEE Computer Society, 2009. p. 488–492. ISBN 978-0-7695-3660-6.

IQBAL, M. A. et al. Survey of network coding-aware routing protocols in wireless networks. **J. Network and Computer Applications**, v. 34, n. 6, p. 1956–1970, 2011.

JACOBSON, V.; BRADEN, R.; BORMAN, D. **TCP Extensions for High Performance**. IETF, maio 1992. RFC 1323 (Proposed Standard). (Request for Comments, 1323). Disponível em: <http://www.ietf.org/rfc/rfc1323.txt>.

JAIN, S. et al. Using redundancy to cope with failures in a delay tolerant network. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 35, p. 109–120, August 2005. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1090191.1080106>.

JAIN, S.; FALL, K.; PATRA, R. Routing in a delay tolerant network. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 34, n. 4, p. 145–158, ago. 2004. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1030194.1015484>.

JAIN, S.; FALL, K.; PATRA, R. Routing in a delay tolerant network. In: **Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications**. New York, NY, USA: ACM, 2004. (SIGCOMM '04), p. 145–158. ISBN 1-58113-862-8. Disponível em: <http://doi.acm.org/10.1145/1015467.1015484>.

JOHNSON, D. B.; MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. In: **Mobile Computing**. [S.l.]: Kluwer Academic Publishers, 1996. p. 153–181.

KALANTARI, M.; LA, R. J. A dtn packet forwarding scheme inspired by thermodynamics. In: **CISS**. [S.l.]: IEEE, 2008. p. 1216–1221.

KATTI, S. et al. Xors in the air: practical wireless network coding. In: **In Proc. ACM SIGCOMM**. [S.l.: s.n.], 2006. p. 243–254.

KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. The ONE Simulator for DTN Protocol Evaluation. In: **SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques**. New York, NY, USA: ICST, 2009. ISBN 978-963-9799-45-5.

KIM, S.-H.; HAN, S.-J. Contour routing for peer-to-peer dtn delivery in cellular networks. In: **Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on**. [S.l.: s.n.], 2012. p. 1 –9.

LEE, C. et al. Regional token based routing for dtns. In: **Information Networking, 2009. ICOIN 2009. International Conference on**. [S.l.: s.n.], 2009. p. 1 –5.

LINDGREN, A.; DORIA, A.; SCHELÉN, O. Probabilistic routing in intermittently connected networks. **SIGMOBILE Mob. Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 7, n. 3, p. 19–20, jul. 2003. ISSN 1559-1662. Disponível em: <http://doi.acm.org/10.1145/961268.961272>.

LUBY, M. LT codes. In: **Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on**. [s.n.], 2002. p. 271–280. Disponível em: <http://dx.doi.org/10.1109/SFCS.2002.1181950>.

MACKAY, D. J. **Information Theory, Inference, and Learning Algorithms**. Cambridge, UK: Cambridge University Press, 2003.

MACKAY, D. J. C. **Information theory, inference, and learning algorithms**. 1st. ed. [S.l.]: Cambridge University Press, 2003. Hardcover. ISBN 0521642981.

MACKAY, D. J. C. Fountain codes. **Communications, IEE Proceedings-**, v. 152, n. 6, p. 1062–1068, 2005.

MITZENMACHER, M. Digital fountains: a survey and look forward. In: **Information Theory Workshop, 2004. IEEE**. [s.n.], 2004. p. 271–276. Disponível em: <http://www.eecs.harvard.edu/~michaelm/postscripts/itw2004.pdf>.

NELSON, S.; BAKHT, M.; KRAVETS, R. Encounter-based routing in DTNs. In: **Proceedings of INFOCOM 2009**. [s.n.], 2009. Disponível em: <http://mobius.cs.uiuc.edu/ snelso20/infocom09.pdf>.

NORDEMANN, F.; TONJES, R. Transparent and autonomous store-carry-forward communication in delay tolerant networks (dtns). In: **Computing, Networking and Communications (ICNC), 2012 International Conference on**. [S.l.: s.n.], 2012. p. 761 –765.

POSTEL, J. **User Datagram Protocol**. IETF, ago. 1980. RFC 768 (Standard). (Request for Comments, 768). Disponível em: <http://www.ietf.org/rfc/rfc768.txt>.

POSTEL, J. **Transmission Control Protocol**. IETF, set. 1981. RFC 793 (Standard). (Request for Comments, 793). Updated by RFCs 1122, 3168, 6093, 6528. Disponível em: <http://www.ietf.org/rfc/rfc793.txt>.

POSTEL, J.; REYNOLDS, J. **Telnet Protocol Specification**. IETF, maio 1983. RFC 854 (Standard). (Request for Comments, 854). Updated by RFC 5198. Disponível em: <http://www.ietf.org/rfc/rfc854.txt>.

PUJOL, J.; TOLEDO, A.; RODRIGUEZ, P. Fair routing in delay tolerant networks. In: **INFOCOM 2009, IEEE**. [S.l.: s.n.], 2009. p. 837 –845. ISSN 0743-166X.

RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D. **The Addition of Explicit Congestion Notification (ECN) to IP**. IETF, set. 2001. RFC 3168 (Proposed Standard). (Request for Comments, 3168). Updated by RFCs 4301, 6040. Disponível em: <http://www.ietf.org/rfc/rfc3168.txt>.

RAMOS, M. C. **Variações sobre Códigos LT**. Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 2010.

SAMUEL, H.; ZHUANG, W.; PREISS, B. Dtn based dominating set routing for manet in heterogeneous wireless networking. **Mobile Networks and Applications**, 2009. Disponível em: <http://dx.doi.org/10.1007/s11036-008-0131-8>.

SCOTT, K.; BURLEIGH, S. **Bundle Protocol Specification**. IETF, nov. 2007. RFC 5050 (Experimental). (Request for Comments, 5050). Disponível em: <http://www.ietf.org/rfc/rfc5050.txt>.

SHAH, R. C. et al. Data mules: Modeling a three-tier architecture for sparse sensor networks. In: **IN IEEE SNPA WORKSHOP**. [S.l.: s.n.], 2003. p. 30–41.

SHANNON, C. E. A mathematical theory of communication. **Bell system technical journal**, v. 27, 1948.

SMALL, T.; HAAS, Z. J. Resource and performance tradeoffs in delay-tolerant wireless networks. In: **Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking**. New York, NY, USA: ACM, 2005. (WDTN '05), p. 260–267. ISBN 1-59593-026-4. Disponível em: <http://doi.acm.org/10.1145/1080139.1080144>.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: **Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking**. New York, NY, USA: ACM, 2005. (WDTN '05), p. 252–259. ISBN 1-59593-026-4. Disponível em: <http://doi.acm.org/10.1145/1080139.1080143>.

SPYROPOULOS, T. et al. Routing for disruption tolerant networks: taxonomy and design. **Wirel. Netw.**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 16, n. 8, p. 2349–2370, nov. 2010. ISSN 1022-0038. Disponível em: <http://dx.doi.org/10.1007/s11276-010-0276-9>.

SUN, X. et al. Performance of dtn protocols in space communications. **Wireless Networks**, Springer US, p. 1–19, 2013. ISSN 1022-0038. Disponível em: <http://dx.doi.org/10.1007/s11276-013-0582-0>.

VAHDAT, A.; BECKER, D. **Epidemic Routing for Partially Connected Ad Hoc Networks**. 2000.

WANG, R. et al. Licklider transmission protocol (ltp)-based dtn for cislunar communications. **Networking, IEEE/ACM Transactions on**, v. 19, n. 2, p. 359–368, 2011. ISSN 1063-6692.

WANG, R. et al. Which dtn clp is best for long-delay cislunar communications with channel-rate asymmetry? **Wireless Communications, IEEE**, v. 18, n. 6, p. 10–16, 2011. ISSN 1536-1284.

WANG, R. et al. Ltp aggregation of dtn bundles in space communications. **Aerospace and Electronic Systems, IEEE Transactions on**, v. 49, n. 3, p. 1677–1691, 2013. ISSN 0018-9251.

WANG, Y. et al. Erasure-coding based routing for opportunistic networks. In: **Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking**. New York, NY, USA: ACM, 2005. (WDTN '05), p. 229–236. ISBN 1-59593-026-4. Disponível em: <http://doi.acm.org/10.1145/1080139.1080140>.

WYATT, J. et al. Disruption tolerant networking flight validation experiment on nasa's epoxi mission. In: **Advances in Satellite and Space Communications, 2009. SPACOMM 2009. First International Conference on**. [S.l.: s.n.], 2009. p. 187 –196.

YU, D.; KO, Y.-B. Ffrdv: fastest-ferry routing in dtn-enabled vehicular ad hoc networks. In: **Proceedings of the 11th international conference on Advanced Communication Technology - Volume 2**. Piscataway, NJ, USA: IEEE Press, 2009. (ICACT'09), p. 1410–1414. ISBN 978-8-9551-9138-7. Disponível em: <http://dl.acm.org/citation.cfm?id=1701835.1701938>.

ZENG, Y. et al. Directional routing and scheduling for green vehicular delay tolerant networks. **Wireless Networks**, v. 19, n. 2, p. 161–173, 2013. Disponível em: <http://dblp.uni-trier.de/db/journals/winet/winet19.html#ZengXLV13>.

ZHANG, Q. et al. Network coding for applications in the delay tolerant network (dtn). In: **Proceedings of the 2009 Fifth International Conference on Mobile Ad-hoc and Sensor Networks**. Washington, DC, USA: IEEE Computer Society, 2009. (MSN '09), p. 376–380. ISBN 978-0-7695-3935-5. Disponível em: <http://dx.doi.org/10.1109/MSN.2009.68>.

ZHAO, W.; AMMAR, M.; ZEGURA, E. Multicasting in delay tolerant networks: semantic models and routing algorithms. In: **Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking**. New York, NY, USA: ACM, 2005. (WDTN '05), p. 268–275. ISBN 1-59593-026-4. Disponível em: <http://doi.acm.org/10.1145/1080139.1080145>.

ZOGHBI, A.; STOJMENOVIĆ, I. Fast algorithms for generating integer partitions. **Int. J. Comput. Math.**, v. 70, n. 2, p. 319–332, 1998.