UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

UM MECANISMO PARA O INCENTIVO BASEADO EM CRÉDITOS PARA REDES TOLERANTES A ATRASOS E DISRUPÇÕES

Daniel de Miranda Chaves Christiani

Orientador
Prof. Dr. Carlos Alberto Vieira Campos

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2015

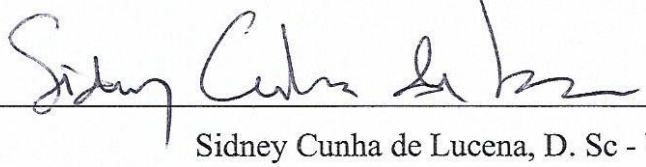
UM MECANISMO PARA O INCENTIVO BASEADO EM CRÉDITOS PARA REDES TOLERANTES A ATRASOS E DISRUPÇÕES

Daniel de Miranda Chaves Christiani

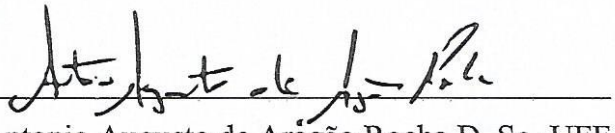
DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓSGRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

Carlos Alberto Vieira Campos, D. Sc - UNIRIO



Sidney Cunha de Lucena, D. Sc - UNIRIO



Antonio Augusto de Aragão Rocha D. Sc -UFF

RIO DE JANEIRO, RJ – BRASIL.

AGOSTO DE 2015

Christiani, Daniel de Miranda Chaves.

C555 Um mecanismo para o incentivo baseado em créditos para redes tolerantes a atrasos e disrupções / Daniel de Miranda Chaves Christiani, 2015.

70 f. ; 30 cm

Orientador: Carlos Alberto Vieira Campos.

Dissertação (Mestrado em Informática) - Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2015.

1. Redes tolerantes a atrasos e desconexões. 2. Protocolo de aplicação sem fio (Protocolo de rede de computador). 3. DiCent.

4. RELICS. I. Campos, Carlos Alberto Vieira. II. Universidade Federal do Estado do Rio de Janeiro. Centro de Ciências Exatas e Tecnológicas. Curso de Mestrado em Informática. III. Título.

CDD – 004.62

Dedico esta dissertação à minha noiva, Karen, cujo apoio e dedicação foram indispensáveis não apenas para a conclusão deste trabalho, mas para toda a minha vida.

Agradecimentos

Agradeço a Deus por ter me dado as forças e ajuda necessárias para que eu concluísse este trabalho. Sem isto, nada seria possível. Agradeço também à minha noiva, Karen, que tem me apoiado, e incentivado por toda esta jornada. Sem o seu amor, e confiança, eu também não teria conseguido terminar este trabalho.

Agradeço ao meu orientador, o professor Carlos Alberto, Beto, que tem me acompanhado desde a minha graduação, sempre me ajudando a conseguir ir mais longe e conquistar novos objetivos. Embora eu sempre tenha trazido diversos problemas à sua porta, ele nunca desistiu de mim, e sempre me ajudou em cada conquista que eu tenho realizado. Agradeço também ao professor Antônio Augusto, o Guto, por todas as reuniões, e orientações, sempre me ajudando e com uma boa ideia quando eu encontrava com dúvidas cerca deste trabalho. Gostaria de agradecer também à professora Morganna, que me deu um apoio incondicional, e tem me ajudado em tudo o que pode, desde os pedidos mais simples até os mais complexos, se dispondo a fazer todo o possível para resolver qualquer tipo de problema que eu muitas vezes trouxe à sua porta. Por fim, gostaria de mostrar meus agradecimentos ao professor Sidney, que me acompanha na UNIRIO desde o meu primeiro período na graduação. São muitos anos de amizade e admiração, que também me incentivaram a continuar as minhas pesquisas em redes de computadores.

Gostaria de agradecer também ao Dr. Yusuf Uddin, pelos códigos do RELICS, que me ajudaram muito na implementação do DiCent e na minha implementação do RELICS. Sem este apoio, eu não teria conseguido terminar os códigos a tempo. E também aos amigos Guilherme Oliveira e Nelson Machado Junior, por também terem oferecido uma ajuda quando precisei.

RESUMO

As redes DTN são redes móveis sem fio caracterizadas por um grande atraso entre as transmissões de mensagens e também por uma grande mobilidade entre os nós, o que causa uma conexão intermitente entre os mesmos. Neste tipo de rede, não é possível garantir, em um determinado intervalo de tempo, que existirá uma conexão fim a fim entre dois nós. Por isso, foram criados diversos protocolos de roteamento, para aproveitar a mobilidade dos nós para realizar a entrega de mensagens. No entanto, alguns nós apresentam um comportamento egoísta, o que pode prejudicar o desempenho da rede. Para evitar isso, foram propostos diversos mecanismos de incentivo, para garantir a colaboração entre os nós.

Nesta dissertação é apresentado o DiCent, uma proposta para um mecanismo de incentivo à colaboração em redes DTN por créditos virtuais. A principal contribuição deste trabalho é o fato do DiCent não utilizar um banco virtual para a distribuição de créditos, mas sim uma abordagem descentralizada, realizada pelos nós da rede.

A avaliação do mecanismo proposto foi realizada através de simulações, utilizando os *traces* de mobilidade do INFOCOM 05 e do RollerNet. Foram utilizados os protocolos de roteamento MaxProp e ProPHET para o encaminhamento de mensagens, e o mecanismo proposto foi comparado com o mecanismo de incentivo RELICS.

Palavras-chave: Redes DTN, DiCent, egoísmo, incentivo, RELICS

ABSTRACT

DTN networks are mobile wireless networks characterized by a large delay between the transmission of messages and also by a great mobility between nodes, which causes an intermittent connection between nodes. In this type of network, we cannot guarantee in a certain period of time, that there will be a close connection between two nodes. Therefore, several routing protocols have been created to take advantage of the mobility nodes to perform message delivery. However, some nodes have a selfish behavior, which could affect the network performance. To avoid this, it has been proposed various incentive mechanisms to ensure collaboration between nodes.

This dissertation presents DiCent, a proposal for a cooperation incentive mechanism in DTN networks using virtual credits. The main contribution of this work is the fact that DiCent does not use a virtual bank for credit distribution, but rather a decentralized approach, carried out by the network nodes.

The evaluation of the proposed mechanism was accomplished through simulations using a trace of mobility form INFOCOM 05 and RollerNet. We used the ProPHET and MaxProp routing protocols for routing messages between nodes, and the proposed mechanism was compared with the incentive mechanism RELICS.

Keywords: DTN networks, DiCent, selfishness, incentive, RELICS

Índice

Índice de Tabelas

Índice de Figuras

Glossário

CRAWDAD – Community Resource for Archiving Wireless Data At Dartmouth

DICENT – DIstributive Incentive

DTN – Delay-Tolerant Network

ONE – Opportunistic Network Environment

MINEIRO – Message-based INcentive mechanism for End-user Improvement of Routing Opportunities

MDR – Multiplicative Decreasing Reward

MuRIS – Multi-Receiver Incentive-Based Dissemination

PROPHET – Probabilistic Routing Protocol using a History of Encounters and Transitivity

RELICS – REaLization of Incentives to Combat Selfishness

SMART – Secure Multilayer Credit-Based Incentive

TTL – Time-To-Live

1 Introdução

Para dar o devido suporte ao avanço da Internet nas últimas décadas, foram criados os protocolos da arquitetura TCP/IP, flexíveis, eficientes, e robustos, mas orientados a conexão que permitem suportar diferentes aplicações em diversos cenários. Porém, para cenários que apresentem longos atrasos e frequentes desconexões, estes protocolos não funcionam, e, por isso, novos protocolos são necessários. Esses cenários são conhecidos como Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks* - DTNs) e um dos seus principais desafios é o roteamento, uma vez que nestes casos é necessário determinar rotas sem o estabelecimento de um caminho fim-a-fim [de Oliveira, 2008].

Este problema de desconexões frequentes pode ocorrer em redes DTNs devido a diferentes fatores, como por exemplo, devido à mobilidade, ocasionada pelas constantes mudanças na topologia da rede, ou pela economia de recursos, assim como pela possibilidade de negação de serviço ou outra característica particular da rede em questão [de Melo, 2011].

Para realizar a transferência de uma informação, esta mensagem deve ser armazenada e encaminhada nó a nó desde a origem até o destino, ou seja, é utilizada uma técnica, conhecida em cenários de redes DTN como armazena-e-encaminha (*store-and-forward*), na qual primeiro a mensagem é recebida integralmente e armazenada para que seja possível seu envio ao próximo nó, que pode ou não ser o destino [de Oliveira, 2008].

Para compensar o gasto de recursos dos dispositivos utilizados nessas redes, são utilizados mecanismos de incentivo, como por exemplo, o pagamento aos nós envolvidos na transmissão da informação como forma de incentivo. Para isso, foi criada uma unidade centralizadora com o objetivo de gerenciar as chamadas moedas virtuais. Porém, essa unidade não condiz com uma das principais características das redes DTN. Uma vez que essas redes sofrem com as frequentes desconexões, a dependência de uma

entidade externa se torna um fator negativo.

1.1 Motivação

A motivação deste projeto vem da necessidade de um mecanismo de incentivo que dispense a necessidade de uma unidade centralizadora externa a uma rede DTN para a utilização de moedas virtuais sem perda de confiabilidade, sendo resistente a possíveis ataques de nós maliciosos.

Embora a existência desta entidade centralizadora pareça a solução para os problemas referentes à segurança do mecanismo de moedas virtuais ela torna a rede DTN, uma rede distribuída, dependente dela, ou seja, caso aconteça algo a esta entidade toda a transmissão de mensagens entre os nós da rede é afetada.

Por outro lado a utilização de moedas virtuais é um eficiente mecanismo de incentivo, sendo necessário um estudo da viabilidade de fazê-lo, sem que aconteça perda da consistência da informação referente a estas moedas, por meio de ataques de nós maliciosos na rede em que a mensagem está sendo transmitida.

1.2 Objetivos

Esta dissertação busca, como seu objetivo principal, propor um novo mecanismo de incentivo baseado em créditos no qual não seja necessária a existência de uma entidade centralizadora. Também será feita uma comparação entre como é feito um encaminhamento de mensagens através de protocolos de roteamento, onde nós egoístas se recusam a encaminhar mensagens, e como os mecanismos de incentivo atuam para melhorar esta entrega, para que seja possível demonstrar o funcionamento e o desempenho do mecanismo proposto. Nesta comparação foram abordadas as métricas referentes não só em relação à fração de entrega, mas também em relação ao atraso médio e a sobrecarga (*overhead*) na entrega destas mensagens.

Para isso primeiro haverá uma conceitualização de uma rede DTN, com suas principais características, como as estratégias de roteamento nesse tipo de rede e os mecanismos de incentivos já existentes para transmissão de mensagens na rede. Depois serão abordados outros conceitos importantes para o entendimento desta proposta como, por exemplo, os conceitos de egoísmo (apresentando também seus efeitos em redes), os conceitos referentes a banco virtual e moeda virtual, além dos tipos de ataques de nós

maliciosos em redes oportunistas. Serão abordados ainda os diferentes tipos de mecanismos de incentivo e como mitigar os efeitos do egoísmo, além do problema de gasto duplo (*Double spending*). Por fim será avaliada a implantação de um mecanismo de incentivo que solucione estes problemas e seja possível a associação do mesmo a qualquer protocolo de roteamento.

1.3 Organização do texto

Além desta introdução o presente trabalho está estruturado da seguinte forma:

- Capítulo II: Fundamentação teórica para dar base ao entendimento do projeto, conceituando os principais conceitos como redes DTNs, quais as estratégias utilizadas pelos protocolos de roteamento nesse tipo de redes, os tipos e ataques sofridos nesses cenários, os mecanismos de incentivo, além das definições relativas à entidade banco virtual e moeda virtual. Também serão apresentados trabalhos relacionados, ou seja, quais são os mecanismos de incentivo já existentes na literatura para estimular nós egoístas a colaborarem durante a transmissão de mensagens na rede.

- Capítulo III: Apresentação do problema que será abordado nesta proposta, quais dificuldades relevantes ao problema deverão ser tratadas, além da proposta de solução, abordando o mecanismo que será proposto, demostrando matematicamente como a solução apresentada é possível, e como ela será implementada, testada e validada.

- Capítulo IV: Apresentação dos resultados obtidos durante as simulações e apresentação do cenário onde estas simulações foram realizadas.

- Capítulo V: Conclusões relativas à proposta e sugestões de trabalhos futuros.

2 Fundamentação teórica e trabalhos relacionados

Neste capítulo serão apresentados, no universo de redes tolerantes a atrasos e disrupções, os principais conceitos relacionados às estratégias utilizadas pelos protocolos de roteamento, além de introduzir a utilização de mecanismos de incentivo e suas categorias, bem como a influência de ataques a estes mecanismos no encaminhamento de mensagens, entre outros conceitos iniciais a fim de gerar uma base suficiente para um melhor entendimento sobre o assunto.

2.1 Redes DTN

Segundo [de Oliveira et al. 2007] redes DTN (*Delay and Disruption Tolerant Networks* - DTNs), [Fall 2003] são redes que as aplicações devem enfrentar um atraso de transmissão significativo, e nem sempre existe um caminho fim-a-fim entre dois nós. Estas redes possuem diversas características que as diferenciam das redes convencionais [de Oliveira et al. 2007] sendo, as principais, citadas por [de Oliveira et al. 2007]:

• **Atrasos longos e/ou variáveis** – esta arquitetura pode ter atrasos desde horas até mesmo dias. O atraso fim-a-fim é determinado através do somatório dos tempos de atraso salto-a-salto, sendo formado por quatro componentes: (1) tempo de espera (tempo de espera de cada nó pelo nó de destino ou pela chegada de um nó intermediário que possa encaminhar as suas mensagens), (2) atraso nas filas (atrasos variáveis que ocorrem nas filas dos nós antes de uma mensagem corrente ser entregue), (3) atraso de transmissão da mensagem e (4) atraso de propagação do sinal (latência) a cada contato entre dois nós [Jones et al., 2005].

• **Frequentes desconexões** – ocasionadas por diferentes motivos, como pela mobilidade que provoca constantes alterações na topologia da rede, por problemas nas condições de comunicação (desvanecimentos), por economia de recursos ou por negação de serviço. Tais eventos resultam em uma conectividade intermitente da rede,

ou seja, não existindo um caminho fim-a-fim entre um nó fonte e um nó de destino, o que pode fazer com que a rede enfrente um ambiente com um atraso significativo.

As redes DTNs são muitas vezes compostas por dispositivos utilizados por pessoas racionais e que nem sempre escolhem colaborar de forma altruística, assim como os dispositivos utilizados nestas redes são portáteis e costumam acompanhar a mobilidade humana, foram desenvolvidos protocolos de roteamento que se aproveitam de características baseadas em um contexto social, como amizade, comunidade e principalmente o egoísmo [Zhu et al. 2013], conceito este que será visto mais adiante neste trabalho.

2.2 Estratégias para o roteamento em redes oportunistas

Redes DTNs utilizam diferentes protocolos de roteamento, desde os mais simples até mesmo os mais avançados, como por exemplo, os protocolos *Spray-and-Wait* [Spyropoulos et al. 2008], Epidêmico [Vahdat et al. 2000], ProPHET (*Probabilistic Routing in Intermittenly Connected Networks*) **[**Lindgren et al 2004**]** entre outros. Em relação à forma como é tratado o roteamento em DTNs são necessários protocolos capazes de tratar os problemas dos atrasos extremamente longos e das frequentes desconexões, características estas já mencionadas, uma vez que os protocolos convencionais não estão aptos a manipular eficientemente a transmissão de dados em DTNs. Um importante fator a ser considerado em redes DTNs está no fato de que em alguns momentos é necessário determinar rotas na rede mesmo que não exista um caminho fim-a-fim entre a fonte e o destino no momento do cálculo da rota [de Oliveira et al. 2007].

Entre os protocolos de roteamento já existentes, vamos tratar primeiramente do protocolo de roteamento Epidêmico, no qual as mensagens enviadas por um nó são replicadas por cada nó que as recebe na rede. Este protocolo gera a maior fração de entrega possível, mas também gera um maior número de mensagens na rede, sobrecarregando-a. Por [de Oliveira et al. 2007] vemos que este protocolo é conhecido como a proposta inicial para redes com características referentes a desconexões e conectividade intermitente [Vahdat e Becker, 2000]. Oliveira et al. [2007] também menciona que este protocolo suporta a entrega eventual de mensagens a destinos arbitrários sem nenhum conhecimento da topologia de rede, sendo necessárias técnicas eficientes para garantir a entrega de mensagens, mesmo quando não há um caminho

totalmente conectado entre a fonte e o destino. Pra tanto, esse protocolo pressupõe que um nó fonte não conhece onde o nó de destino está localizado, além de não saber qual a melhor rota para alcançá-lo. O objetivo é que, com a mobilidade dos nós na rede, seja possível que os mesmos entrem no alcance de transmissão uns dos outros periodicamente e, também, de maneira aleatória. Com isto, a mobilidade dos nós é utilizada como solução para a entrega de mensagens, e não como um problema que precise ser superado na rede. Neste caso, somente a conectividade periódica par-a-par é necessária para garantir a entrega de mensagens eventuais. Listas com informações que identificam as mensagens armazenadas em cada nó são trocadas, no momento em que dois nós iniciam um contato, o que possibilita que o nó determine quais as mensagens existentes no *buffer* do nó vizinho que o mesmo ainda não possui. Depois que as mensagens são identificadas, cada nó solicita o envio das cópias das mensagens que ainda não possui.

O processo de troca de mensagens se repete toda vez que um nó entra em contato com um novo vizinho, permitindo que as mensagens sejam rapidamente distribuídas pelas partes conectadas da rede. Assim, quanto maior o número de cópias de uma mesma mensagem for encaminhado na rede, maior será a probabilidade da mensagem ser entregue e menor será o atraso.

Do trabalho de [Martins, 2014] temos que a única restrição na versão inicial é o tamanho do *buffer* que funcionava como uma fila para armazenar as mensagens com as mais antigas sendo apagadas para dar lugar as mais novas. Existem variações do protocolo Epidêmico original que tentam mitigar os problemas de *buffer* ao limitar o número de nós pelos quais uma mensagem pode trafegar.

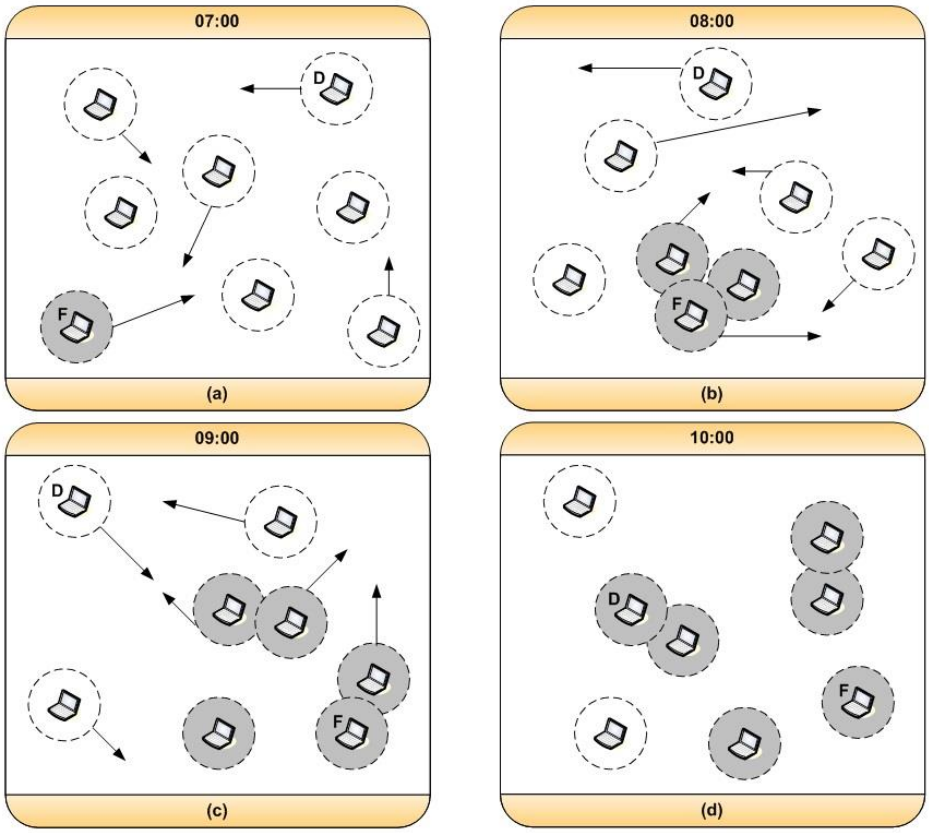


Figura 1: Exemplo de roteamento epidêmico por [de Oliveira et al. 2007]

Para diminuir a sobrecarga na rede, causada pelo grande número de mensagens utilizado para obter uma maior fração de entrega, foi proposto o protocolo Spray-and-Wait [Spyropoulos et al. 2008], como uma evolução do protocolo Epidêmico, sendo uma abordagem que utiliza um número pré-definido de cópias de uma mensagem disseminadas entre os nós que são encontrados pela rede, e depois são armazenadas por estes nós até que encontrem o nó destinatário. Ou seja, neste protocolo existem duas fases distintas, uma inicial conhecida como *Spray* em que o dispositivo gerador da mensagem é quem decide o número de cópias a serem distribuídas pela rede, criando cópias da mesma no dispositivo de origem. Quando uma unidade ou nó recebe uma destas cópias, o protocolo entra na segunda fase, que é denominada de fase de espera (*Wait*). Nesta segunda fase, o dispositivo aguarda o contato com o destinatário para entregar a mesma diretamente.

Outro tipo de protocolo é o PRoPHET, um exemplo de solução baseada na estratégia de inundações, sendo definido como um protocolo de roteamento probabilístico que utiliza um histórico de encontros.

Sendo assim, o ProPHET utiliza uma métrica chamada fração de entrega da mensagem a qual é utilizada para determinar a possibilidade de um nó repassar, ou não, uma mensagem para outro nó durante um encontro.

Segundo [Borah et al 2010] o protocolo de roteamento PRoPHET é preferível nas situações onde alguns dos nós móveis criam padrões de conectividade que não são completamente aleatórios, mas podem ser previstos. Os nós deverão ser capazes de estabelecer uma ligação TCP para a troca de informação. Este protocolo utiliza um mecanismo que é similar aquele utilizado métricas baseadas em vetores, de protocolos de roteamento, em que a métrica pode ser de distância ou custo. Este protocolo será utilizado nesta proposta durante a fase de simulações.

Outro protocolo de roteamento probabilístico é o MaxProp [J. Burgess e Levine 2006],o qual utiliza vários mecanismos para definir a ordem na qual os pacotes deverão ser apagados ou transmitidos, com o objetivo de garantir uma maior taxa de sucesso na entrega e menor atraso médio na entrega das mensagens. Este protocolo se baseia na probabilidade de encontro dos nós, no qual cada nó mantém uma lista com as probabilidades de entrega para os outros nós da rede, atribuindo a estes uma estimativa de probabilidade de entrega. Aqueles que foram criados recentemente possuem maior prioridade. Quando dois pacotes de mensagens possuem destinos com a mesma prioridade, ganha precedência aquele que possuir rota com menor número de saltos [VIEIRA, 2012]. Este protocolo também será utilizado nesta proposta durante a fase de simulações.

Outro conceito importante refere-se aos possíveis comportamentos que os nós envolvidos no encaminhamento de mensagens podem ter. Chen e Chen[2007] estabelecem pelo menos 3 conceitos muito utilizados pela literatura, sendo eles:

- **Nós Colaborativos:** são os nós que compartilham os recursos durante o encaminhamento da mensagem passando as informações verdadeiras.
- **Nós Egoístas:** são os nós que não compartilham recursos durante o encaminhamento de mensagens.
- **Nós Maliciosos:** são nós que corrompem o sistema com informações inválidas durante um ataque a rede.

2.3 Mecanismos de Incentivo

Mecanismos de incentivo são mecanismos com o objetivo de compensar o gasto de recursos e incentivar os nós egoístas a colaborarem com a rede, sendo a maioria deles projetada para cenários *unicast*. Porém, enquanto estes regimes podem efetivamente incentivar os nós egoístas na transmissão de pacotes entre nós, a eficiência de neste tipo de transmissão pode ser abaixo do esperado em cenários *multicast*, que são representados em sistemas como, por exemplo, o *Publish/Subscribe* [Wang et al 2014].

Os mecanismos de incentivo propostos para redes DTN podem ser divididos em três categorias:

1. Abordagem *Tit-for-tat* [Shevade et al. 2008] para forçar a cooperação entre os nós. A relação *Tit-for-tat* foi modelada entre os pares de nós, ou seja, um nó irá transmitir os pacotes de outro nó na mesma medida que tem seus pacotes transmitidos. Esta abordagem permite que nós egoístas maximizem seus próprios desempenhos, sem qualquer degradação significativa do desempenho de todo o sistema. O problema desta abordagem ocorre quando dois nós acabam de se encontrar. Como ambos não possuem conhecimento prévio um sobre o outro, e nenhum pacote foi transmitido entre os mesmos, não é possível estabelecer se o outro nó irá cooperar, e caso ambos os nós decidam esperar que seus pacotes sejam transmitidos primeiro antes de cooperarem, nenhum pacote será transmitido na rede. Shevade et al. [2008] apresenta dois conceitos relacionados aos nós envolvidos na transmissão de mensagens, o conceito de generosidade, que consiste no fato de um nó encaminhar ε pacotes antes de retaliar um possível nó egoísta e o conceito da contrição, mecanismo para evitar que dois nós parem de transmitir pacotes um do outro devido a uma retaliação mútua. Assim, um nó é capaz de perceber que a redução do encaminhamento de suas mensagens foi causada por uma ação egoísta passada, e por isso não retalia de volta.
2. Reputação entre os nós participantes: a ideia está ligada a uma análise do grau de colaboração do nó, ou seja, quanto mais colaborativo o nó, maior a sua reputação [Zhu et al. 2013]. Nessa proposta cada nó mantém o controle dos pacotes que o mesmo tenha enviado a um vizinho particular. Porém, esta técnica acaba por não ser viável em redes DTNs por causa da grande separação espacial

entre o encaminhamento de mensagens sucessivas, uma vez que é significativamente difícil verificar se um determinado pacote foi ainda transmitido ou não [Uddin et al. 2010].

3. Créditos: Segundo [Zhu et al. 2013] os esquemas baseados em crédito visam introduzir uma forma para regular as relações de pacotes de encaminhamento entre os nós diferentes, os quais recebem esta moeda virtual por meio dos envios de pacotes feitos para os outros nós. Para cada pedido de encaminhamento, o banco virtual cobra ao remetente uma quantidade extra desta moeda virtual, e os nós intermediários resgatam suas recompensas em algo que funciona como um banco após a entrega destes pacotes ser bem-sucedida. Esta técnica muitas vezes é utilizada para incentivar os nós egoístas e foi a técnica escolhida para o mecanismo que será proposto neste trabalho. Esta técnica foi escolhida por poder proporcionar uma maior segurança aos nós da rede DTN do que os mecanismos por reputação.

2.4 Moedas e bancos virtuais

Uma moeda virtual pode ser definida como um meio para se realizar transações assim como uma moeda convencional, mas sem todas as características que uma moeda real possui, não estando ligada a nenhuma soberania e sem nenhum respaldo legal em nenhuma jurisdição [Ly 2014]. Nestes mecanismos, o nó que deseja transmitir uma mensagem paga uma quantia de uma moeda virtual aos nós que participaram da retransmissão de sua mensagem. No entanto, estas moedas virtuais são apenas códigos de computador, e poderiam ser facilmente duplicadas, causando o problema conhecido como gasto duplo (*double spending*) e consequentemente tornando a moeda sem nenhum valor. A maioria das moedas virtuais necessita de um banco virtual para gerenciar a troca de moedas e evitar o problema do gasto duplo. Uma moeda virtual, conhecida como BitCoin, propõe um controle descentralizado das moedas, baseado em um bloco de operações conhecido como *block chain*. Este trabalho buscou uma inspiração na moeda virtual BitCoin para criar um mecanismo de incentivo por créditos em redes DTN que funcione de forma descentralizada.

2.4.1 Moeda BitCoin

Esta moeda virtual, introduzida em 2009, existe unicamente em formato eletrônico. Um programa é executado na Internet e todas as transações de BitCoin são registradas em algo que podemos chamar de livro-público. O BitCoin age de forma distribuída, ou seja, não há quem administre o BitCoin, sendo assim, o seu valor é volátil. É importante ressaltar que o algoritmo BitCoin foi programado para libertar *bitcoins* em quantidades decrescentes até que atinja um total de 21 (vinte e um) milhões de *bitcoins* e nenhum *bitcoin* adicional será criado uma vez que este número seja alcançado [Ly 2014].

Desde a sua introdução, esta moeda virtual tem recebido uma crescente popularidade e aprovação, sendo aceita por diferentes empresas *online*. A moeda virtual apresenta muitos benefícios, entre os mesmos a privacidade e a conveniência da moeda, além do fato desta moeda apresentar maior estabilidade financeira do que uma moeda nacional, visto sua instabilidade [Ly 2014].

2.5 Ataques a mecanismos de incentivo

[Nadeem, 2013] divide os ataques à camada de rede em duas categorias principais, chamadas de ataques passivos e ataques ativos, como ilustra a Figura 2.

- **Ataques passivos:** são aqueles que não atingem diretamente o funcionamento do protocolo de roteamento, mas buscam algumas informações valiosas através de análise de tráfego, o que pode comprometer a segurança da rede. Portanto, os mesmos não serão abordados neste trabalho. Os ataques passivos se dividem em espionagem (*eavesdropping*), que ocorre quando um nó atacante escuta as transmissões na rede sem fio para capturar informações, divulgação da localização (*location disclosure*) onde o nó atacante busca descobrir a localização de um determinado nó na rede e análise de tráfego, que é utilizada pelo nó malicioso para descobrir quais são os nós mais importantes da rede.
- **Ataques ativos:** são aqueles que buscam realizar atividades intrusivas como a modificação, injeção, criação, fabricação ou envio dos pacotes de dados ou de roteamento, resultando em várias interrupções para a rede. Alguns desses ataques são causados por uma única atividade de um intruso, e outros podem ser causados por uma sequência de atividades. Os ataques ativos podem ainda ser divididos em ataques por descarte malicioso de pacotes (ocorre quando um nó malicioso pode decidir abandonar esses pacotes em vez de encaminhá-los ao

próximo nó) e ataques de encaminhamento (quando o nó malicioso visa explorar as vulnerabilidades dos algoritmos de roteamento de cooperação).

Uma vez dadas estas definições, é possível notar que os ataques ativos, se compararmos aos os ataques passivos, são mais nocivos à rede, por perturbarem o funcionamento da rede, além de serem graves o suficiente para até mesmo interromper tal funcionamento ou degradar o desempenho da rede significativamente, como no caso de ataques de negação de serviço, portanto tomaremos como foco neste trabalho a prevenção aos ataques ativos da rede, mas precisamente, aos ataques de encaminhamento [Nadeem, 2013] que serão descritos a seguir.

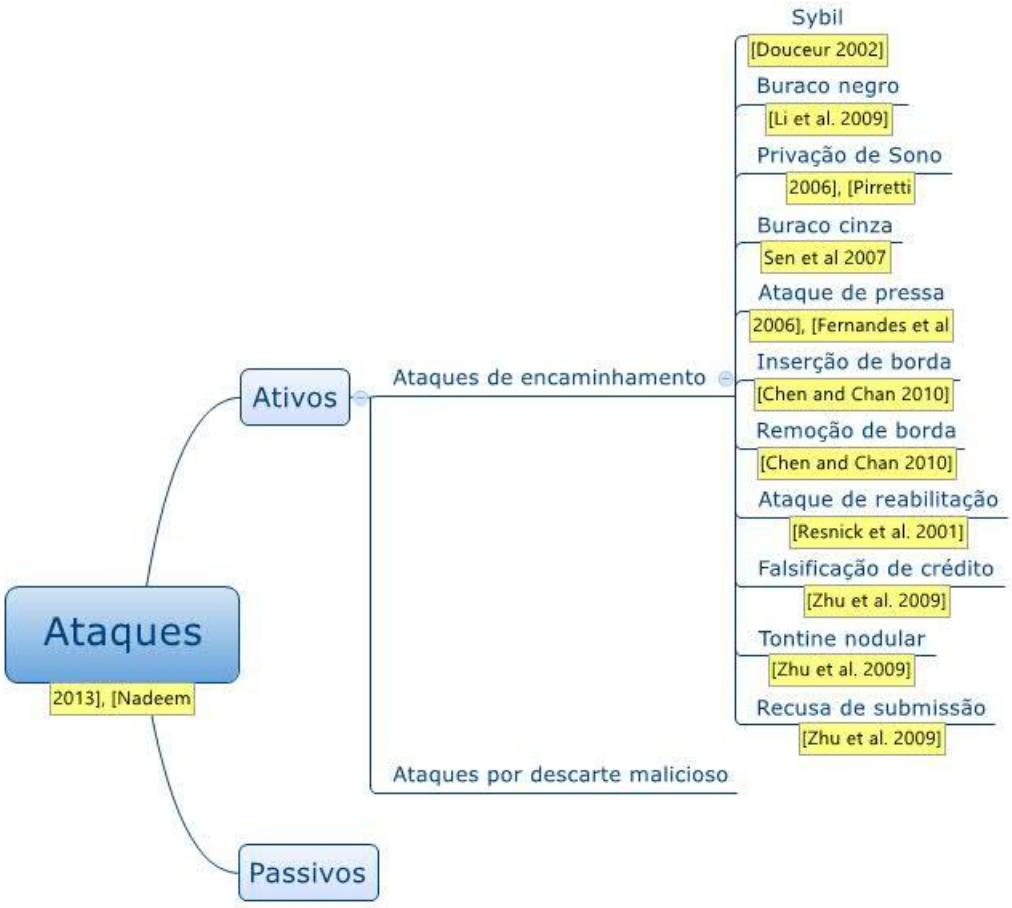


Figura 2: Classificação de Ataques à rede

- **Sybil**

Ataques Sybil foram introduzidos pela primeira vez no contexto de redes *peer-to-peer* como uma forma de ataque esgotamento de recursos, e, posteriormente, analisada em cenários de redes sem fio. O termo foi utilizado para identificar a falsificação de identidades múltiplas em um sistema. Em ataques Sybil, um nó malicioso reivindica um grande número de identidades do cliente, seja através da personificação de outros nós

legais ou alegando identidades falsas, ou seja, o nó egoísta cria outros nós virtuais para então se beneficiar dos mecanismos de incentivo e cooperação em redes oportunistas. Através da negação de serviço, este ataque coloca seriamente em risco a disponibilidade de serviços de rede para sistemas sem fio [Douceur 2002], [Xiao, 2009].

- ***Black hole* (Ataque buraco negro)**

Tipo especial de ataque que geralmente ocorre nos protocolos reativos. Este ataque ocorre quando nós maliciosos atraem os pacotes de dados por anunciar falsamente uma nova rota (mais curta) para o destino, porém, após atraí-los, descarta os pacotes [Rashmi, 2014].

O nó malicioso pode fornecer métricas forjadas através de pacotes de roteamento falsos para que outros nós utilizem rotas que possuam o nó atacante e com isso o nó malicioso pode descartar ou utilizar os pacotes capturados para lançar outros ataques mais sofisticados [Li et al. 2009].

- ***Sleep deprivation* (Ataque de privação do sono)**

É um ataque direcionado ao sistema de conservação de energia do dispositivo. O ataque consiste em não deixar o dispositivo entrar em modo de baixo consumo de energia, de uma maneira que parece ser legítimo; com isso, mantem o nó ativo para reduzir drasticamente o tempo de vida do dispositivo alvo. Infelizmente, este ataque é difícil de detectar, uma vez que o mesmo é efetuado apenas através da utilização de interações aparentemente inocentes [Pirretti, 2006]. Ou seja, é um ataque em que um invasor interage com o nó de uma maneira que parece ser legítima, porém o objetivo desta interação é manter o nó alvo com o modo de conservação de energia desativado [Nadeem, 2013].

- ***Grey hole* (Ataque de encaminhamento seletivo ou buraco cinza)**

Este ataque é um caso especial do ataque *Black Hole*, na qual um intruso captura pela primeira vez as rotas, tornando-se parte delas na rede (como acontece com o ataque *Black Hole*), para poder descartar os pacotes seletivamente [Nadeem, 2013].

- ***Rushing attack* (Ataque da pressa)**

Este ataque busca explorar o controle de sobrecarga utilizado nos pacotes de controle, através de uma requisição de roteamento antecipada. Normalmente apenas uma requisição é utilizada para cada rota descoberta [Nadeem, 2013]. Um invasor pode explorar essa propriedade, espalhando pacotes de solicitação de rota rapidamente em toda a rede para suprimir quaisquer pacotes de solicitação de rota legítimos posteriores e acrescentar rotas que o nó malicioso faça parte. Ou seja, uma vez que o nó atacante

envia um *Route Request* de forma mais rápida aos demais nós da rede, o mesmo faz com que todas as respostas passem pelo nó atacante. Para que seja o primeiro a responder, e assim descartar as demais respostas provenientes dos outros vizinhos [Fernandes et al, 2006].

Além dos ataques identificados no trabalho de Nadeem [2013], existem outros ataques importantes de serem mencionados, sendo os mesmos:

- ***Edge insertion* e *Edge hiding* (Ataques por inserção de borda e remoção de borda)**

Tipos especiais de ataque Sybil identificados por [Chen and Chan 2010]. No ataque de *Edge insertion* (Inserção de borda), o nó malicioso pode criar uma conexão fantasma com outros nós Sybil para receber uma recompensa maior (ou seja, o mesmo pode ser recompensado duas vezes).

O ataque por *Edge hiding* ocorre quando um no egoísta, ao invés de repassar a mensagem para outro nó que teria uma chance maior de entregar a mensagem, ou poderia entrega-la mais rápido, deixa de enviar uma mensagem deliberadamente, para não dividir a recompensa, e ganhar um incentivo maior [Chen and Chan 2010].

- ***Whitewashing* (Ataque de reabilitação)**

Ataque em sistemas que usam mecanismos de incentivo por reputação, [Resnick et al. 2001], faz com que, para evitar uma reputação negativa, o nó atacante troque a sua identidade. O nó desiste de todas as transações, mas o mesmo abandona e retorna à rede repetidas vezes utilizando novas identidades para não sofrer consequências de uma má reputação.

- **Falsificação de crédito**

Apresentado em [Zhu et al. 2009] consiste no fato de um nó egoísta em inserir um crédito falso na moeda em camadas, para que possa se beneficiar do crédito extra, tendo, ou não, participado da transmissão da mensagem. Com isto o nó receberia uma quantidade de crédito como recompensa por uma retransmissão.

- **Tontine nodular**

O ataque consiste em um nó malicioso tentar remover uma ou mais camadas da moeda, removendo a recompensa dos nós participantes e ficando com uma parte maior da recompensa [Zhu et al. 2009].

- **Recusa de submissão**

Como redes DTN não possuem uma conexão fim-a-fim, os nós da rede envolvidos na transmissão não tem a informação sobre o andamento da transmissão da mensagem. Por isso, para acontecer a distribuição da recompensa é necessário que a mensagem chegue ao destino, ou seja, a rede depende do último nó que transmite a mensagem para distribuir a recompensa. O ataque acontece neste momento, o nó malicioso se recusar a submeter a mensagem e com isso, os outros nós podem ficar sem a sua recompensa. O último nó então receberia a sua recompensa diretamente do nó emissor [Zhu et al. 2009].

2.6 Mecanismos de incentivo baseados em crédito:

Nesta Seção serão abordados mecanismos de incentivo baseados em crédito, conforme descrito na Seção 2.3, SMART, MobiCent, MuRIS.

2.6.1 SMART:

Criado em 2009, o SMART (*Secure Multilayer Credit-Based Incentive*) [Zhu et al. 2013] é um esquema de incentivo baseado em crédito e propõe a utilização de criptografia como prova de envio, e a utilização de um banco virtual para gerenciar o pagamento e a recompensa. Este esquema propõe o uso de uma moeda composta de camadas, em que cada camada é assinada por um dos nós que participarem na retransmissão para definir quem irá compartilhar a recompensa. [Zhu et al 2009] define a primeira camada como uma camada de base, gerada com o objetivo de indicar as características do pagamento como o valor, as condições de remuneração, e outras políticas de recompensa. Durante todo o processo cada nó intermediário irá gerar uma nova camada de base sobre as camadas anteriores, anexando uma assinatura digital não falsificável. Essa nova camada (camada de endosso), na qual o nó de encaminhamento concorda em fornecer serviço de encaminhamento sob as exigências CoS (*Class of Service* – Classe do Serviço) pré-definidas, será recompensado de acordo com a política definida futuramente. Com esta camada é fácil de rastrear o caminho de propagação e determinar cada nó intermediário, verificando a assinatura destes. Na fase da recompensa e da cobrança, se as condições de remuneração forem satisfeitas, cada nó que participou do encaminhamento irá compartilhar o crédito.

Os problemas com este mecanismo estão na necessidade de um banco virtual para controlar a cobrança e a recompensa dos nós. E na ausência da informação acerca da

cobrança dos nós, mencionando apenas que a recompensa deve ser distribuída igualmente entre os nós participantes. Devido á necessidade de criptografar as assinaturas, o SMART também acaba gerando uma complexidade extra, relacionada ao custo para se criptografar as assinaturas a cada salto. Por fim, o SMART é incapaz de proteger a rede contra outros ataques ao mecanismo de incentivo que foram descobertos mais tarde, como a inserção e remoção de borda.

2.6.2 MobiCent

O MobiCent [Chen and Chan 2010] é um mecanismo de incentivo baseado em crédito virtual que funciona em cima de qualquer protocolo de roteamento e não necessita de nenhum mecanismo para a detecção de nós egoístas, já que os mesmos não conseguem se beneficiar com nenhum tipo de ataque. É demonstrado neste trabalho que todos os nós acabam por trabalhar cooperativamente, atingindo o Equilíbrio de Nash.

A recompensa a ser paga aos nós que participarão da transmissão é baseada num jogo de Vickery-Clarke-Groves, onde o cliente pagaria o maior lance para cada participante do caminho vencedor. Para evitar o ataque de inserção de borda, o MobiCent utiliza uma recompensa decrescente multiplicativa (MDR - *Multiplicative Decreasing Reward*) onde a recompensa recebida por cada nó é diminuída pela metade para cada nó extra. O MobiCent possui dois mecanismos para impedir o ataque de *Edge hiding*, que é utilizado de acordo com o objetivo do cliente. Caso o cliente queira o menor custo, é acionado o algoritmo min-Cost. Caso o objetivo seja o menor *delay*, será utilizado o algoritmo min-Delay [Chen and Chan 2010].

O principal ponto positivo do MobiCent está na sua capacidade de garantir que todos os nós irão cooperar, já que não irão obter nenhum benefício os nós que se comportarem de forma egoísta. Outro ponto interessante é que o MobiCent não trabalha com o roteamento de pacotes, podendo coexistir com qualquer protocolo de roteamento. Um dos principais problemas desse mecanismo é que o mesmo também necessita de um agente externo confiável (ou seja, um banco virtual), e também foi modelado com nós fixos para ajudarem na transmissão [Chen and Chan 2010].

2.6.3 MuRIS

O MuRIS (*Multi-Receiver Incentive-Based Dissemination*) é um mecanismo de incentivo baseado em crédito, que também busca selecionar os caminhos que podem chegar a vários assinantes da maneira mais eficiente, através de um esquema de

compartilhamento de informações que permite que os nós utilizem informações mantidas localmente sobre encontros e caminhos de entrega parciais para determinar se os mesmos devem encaminhar informações recebidas para outros nós [Wang et al. 2014]. Este mecanismo utiliza um algoritmo parecido com o do MobiCent para obrigar os nós a trabalharem cooperativamente. No entanto, o grande ponto positivo do MuRIS é que o mesmo é um mecanismo de incentivo para cenários *multicast*. Este mecanismo utiliza de informação histórica para construir métricas para ajudar no encaminhamento de nós para obter os melhores resultados em ambientes com arquitetura do tipo *Publisher*/*Subscriber* para que o conteúdo a ser disseminado seja encaminhado para um grande número de nós interessados, com baixo *delay* (atraso médio) e *overhead* (sobrecarga). [Ning et al. 2013]. Além do fato de que esse mecanismo ainda necessita de uma entidade central para as transações com a moeda utilizada como incentivo, o fato dele ter sido modelado para cenários onde múltiplos nós receberiam um conteúdo das mensagens o torna inadequado para cenários *unicast*. Outro ponto negativo do MuRIS é o fato do mesmo possuir um roteamento próprio incluído no mecanismo de incentivo, voltado para cenários *multicast*. Isto impede que o mecanismo possa ser avaliado com outros protocolos de roteamento, e assim avaliar o desempenho deste mecanismo com outras propostas apresentadas pela literatura, em conjunto com um mesmo protocolo de roteamento.

2.7 Mecanismos de incentivo baseados em reputação

Nesta Seção serão apresentados dois mecanismos de incentivo por reputação conforme descrito na Seção 2.3: O RELICS e o MINEIRO.

2.7.1 RELICS

O protocolo RELICS (*REaLization of Incentives to Combat Selfishness*) [Uddin et al. 2010] é um mecanismo de incentivo baseado em reputação, que tem por objetivo recompensar os nós de uma rede DTN baseado em seu gasto de energia. Nesse mecanismo, um nó só tem suas mensagens encaminhadas caso já tenha encaminhado uma mensagem de outro nó. Conforme o nó encaminha mais mensagens, seu *rank* em relação a outros nós vai subindo, o que permite que o mesmo envie mais mensagens. Isso tem por objetivo impedir ataques de *whitewashing*, já que caso o nó entre com uma nova identidade, não poderá enviar muitas mensagens, e voltará ao *rank* inicial.

Mensagens de nós com *ranks* mais altos possuem uma maior prioridade em relação às mensagens de nós com *ranks* mais baixos.

Esse mecanismo permite que os nós limitem o seu uso de recursos de maneira proporcional com o seu gasto energético a partir da rede, uma vez que esse projeto define energia como a razão principal por trás do comportamento egoísta dos nós, sendo a primeira proposta de um mecanismo de incentivo consciente em termos energéticos para DTNs [Uddin et al. 2010].

Assim como o MobiCent, o RELICS pode ser utilizado em conjunto com qualquer protocolo de roteamento [Uddin et al. 2010].

2.7.2 MINEIRO

Proposta apresentada mais recentemente, no trabalho de [Mota et al, 2015], este mecanismo visa detectar e evitar os nós egoístas com base na origem das mensagens recebidas utilizando somente informações locais [Mota et al, 2015].O MINEIRO (*Message-based INcentive mechanism for End-user Improvement of Routing Opportunities*) constrói uma tabela de reputação dos nós, baseando-se apenas em informações locais, na origem da mensagem recebida e sobre qual nó está encaminhando esta mensagem, evitando assim entidades certificadoras centrais.

Esta proposta é baseada em teoria dos jogos, e difere de outras propostas devido ao fato de não ser necessário que um nó tenha conhecimento prévio de outros nós ou que seja feita a troca de chaves públicas [Mota et al, 2015].

2.8 Comparação dos mecanismos de incentivo

Para avaliar os pontos fortes e fracos de algumas propostas encontradas na literatura, foi feita uma análise com base nas características apresentadas pelos autores dos artigos apresentados. O objetivo desta análise é apresentar os problemas e limitações dos mecanismos de incentivo existentes, e motivar a proposta de solução apresentada neste trabalho. A Tabela 1 apresenta as informações encontradas sobre os mecanismos de incentivo nos quais esta proposta foi baseada.

Nome	**Descrição**	**Tipo**	**Prós**	**Contras**	**Artigo**
Smart(Secure Multilayer Credit-Based Incentive	Utilização de criptografia como prova de envio, e a utilização de um banco virtual para gerenciar o pagamento e a recompensa. A moeda é composta de camadas com assinaturas por um dos nós que participarem na retransmissão para definir quem irá compartilhar a recompensa.	Baseado em crédito	Oferece um sistema seguro para a troca de créditos sem sacrificar muitos recursos da rede	Necessidade de um banco virtual; Vulnerável aos ataques *Edge Insertion* e *Edge Hiding*	Zhu, H., Lin, X., Lu, R., Fan, Y., & Shen, X. (2009). Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks.*Vehicular Technology, IEEE Transactions on*, *58*(8), 4628-4639.
MobiCent	Possui a capacidade de garantir que todos os nós irão cooperar, sem a necessidade de ativamente buscar comportamentos egoístas e maliciosos dos nós. Tem o funcionamento baseado em um jogo do tipo Vickery-Clarke-Groves, utilizando a teoria dos jogos para garantir a cooperação e	Baseado em crédito	Funciona com qualquer protocolo de roteamento e oferece proteção contra ataques de *Edge*	Necessidade de um banco virtual.	Chen, B. B. and Chan, M. C. (2010). Mobicent: a credit-based incentive system for disruption tolerant network. In INFOCOM, 2010 Proceedings IEEE, pages 1–9. IEEE.

	um pagamento justo de recompensa.		*insertion* e *Edge hiding*		
RELICS	Recompensa os nós da rede baseado em seu gasto de energia através de um sistema de *ranking*.	Baseado em reputação	Garante uma economia do uso de energia.	Vulnerável ao ataque Sybil. É necessário manter o *Ranking* atualizado em todos os nós.	Uddin, M. Y. S., Godfrey, B., and Abdelzaher, T. (2010). Relics: In-network realization of incentives to combat selfishness in dtns. In Network Protocols (ICNP), 2010 18th IEEE International Conference on, pages 203–212. IEEE.
MuRIS	Mecanismo de incentivo muito parecido com o MobiCent, mas possui um roteamento específico para cenários *multicast Publish/Subscriber*. Utiliza um mecanismo de recompensa parecido com o do MobiCent, mas recompensando caminhos que alcancem um maior número de nós *subscribers*, através no menor caminho.	Baseado em crédito	Possui melhor performance em ambientes *multicast*, do tipo *Publish/Subscribe.*	Necessidade de um banco virtual. Não funciona tão bem em ambientes *unicast*.	Incentive Based Data Sharing in Delay Tolerant Mobile Networks(2014) Wang, Y., Chuah, M.-C., and Chen, Y. (2012). Incentive driven information sharing in delay tolerant mobile networks. In Global Communications Conference (GLOBECOM), 2012 IEEE, pages 5279–5284. IEEE.

MINEIRO	Proposta apresentada mais recentemente, no trabalho de [Mota et al, 2015], este mecanismo visa detectar e evitar os nos egoístas com base na origem das mensagens recebidas utilizando somente informações locais [Mota et al, 2015].	Baseado em reputação	Não é necessário o conheciment o prévio de outros nós ou que seja feita a troca de chaves públicas [Mota et al, 2015].	Mecanismo não foi testado com um modelo me mobilidade real. Vulnerável a ataques que falsifiquem a origem da mensagem	Mota, V., Macedo, D., Ghamri-Doudane, Y., Nogueira, J. (2015). MINEIRO: Um Mecanismo de Incentivo para Aplicações em Redes Oportunísticas. XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.

Tabela 1: Comparação entre os mecanismos de incentivo analisados neste trabalho

3 O mecanismo proposto

O objetivo deste capítulo é discorrer sobre o problema motivador da dissertação, sua importância no cenário das redes DTNs, bem como apresentar qual o impacto deste problema no desenvolvimento de mecanismos de incentivo.

3.1 O problema da dependência do banco virtual

Uma característica comum em todos os mecanismos de incentivo que utilizam o sistema de créditos como incentivo aos seus nós é a existência de uma entidade centralizadora que funcione fora da rede, para gerenciar o pagamento de créditos entre os nós. Esta entidade, chamada de banco virtual, também pode prover outros serviços para a rede, como criptografia e autenticação dos nós.

Como visto na Seção 2.6, todos os mecanismos de incentivo baseados em crédito utilizam um banco virtual para controlar a cobrança de créditos e a recompensa dos nós retransmissores. Como também foi visto anteriormente, o uso deste banco virtual é um problema, pois é uma grande restrição para uma rede DTN, que é dispersa e desconexa por natureza, e nem sempre poderia acessar este banco virtual. Por isso, será proposto um mecanismo de incentivo por crédito que não utiliza um banco virtual para resolver o problema da necessidade de uma entidade central nos mecanismos de incentivo que utilizam créditos virtuais para incentivar os nós da rede a colaborarem.

No entanto, embora a utilização de um banco virtual solucione muitos problemas, sua utilização também é uma grande limitação para uma rede DTN, por causa de sua natureza distributiva, uma vez que não há como se garantir a conectividade permanente entre a rede e o banco virtual. Sendo assim, se esse banco virtual de alguma forma estiver indisponível, ou se for comprometido, nenhum nó na rede poderá trocar créditos, e então não haverá mais um incentivo para a colaboração entre esses nós independente do mecanismo utilizado.

Porém, ao eliminar o uso do banco virtual para controlar as transações entre os nós dá rede, será preciso garantir que os próprios nós possam controlar a troca de créditos.

Esta proposta busca criar um mecanismo no qual não seja utilizado um banco virtual como unidade centralizadora em uma rede distribuída.

3.2 Descrição do mecanismo Proposto

O trabalho aqui apresentado propõe uma abordagem semelhante à utilizada nas redes de BitCoins[1], para solucionar o problema apresentado na seção anterior e também para descrever como estas moedas serão geradas e mantidas pelo sistema, ou seja, esta proposta visa criar um mecanismo de incentivo baseado em créditos, apresentando como diferencial a não utilização de um banco virtual. Essas moedas virtuais, geradas a partir da retransmissão de mensagens na rede, serão utilizadas para recompensar os nós que trabalharem de forma cooperativa.

O mecanismo proposto, o qual chamaremos de DiCent (*DIstributive inCENTive*), foi desenvolvido em Java, baseando-se nos mecanismos já existentes, o MobiCent [Chen and Chan 2010], e o RELICS [Uddin et al. 2010], e sua descrição de classes e funções encontra-se disponível na Seção 3.4, neste capítulo.

3.2.1 Sistema de pagamento

Para tratar o pagamento de créditos como incentivo aos nós retransmissores, o DiCent utilizou as equações apresentadas por [Chen and Chan 2010] no desenvolvimento do MobiCent para garantir a cooperação dos nós, e evitar ataques de *Edge Insertion* pelos nós retransmissores. Os detalhes matemáticos de como serão realizadas o pagamento e a cobrança de créditos estão descritas na seção da formalização matemática do mecanismo proposto.

Os créditos serão distribuídos conforme o algoritmo MDR apresentado em MobiCent [Chen and Chan 2010]. O DiCent não realiza uma distribuição de moedas prévia. Conforme exista uma lista de mensagens para ser enviada, o mecanismo verifica se o nó que receberá a mensagem não gastará nenhum crédito (ou seja, início da rede) e assim, enquanto não for atingido um valor máximo fixado em 4100 moedas, os nós destinatários continuarão gerando créditos. Quando o número de moedas criadas atingirem o valor máximo, o incentivo a colaboração será realizado através da troca de

[1] O BitCoin é uma moeda virtual que foi proposta por um hacker, ou um grupo de hackers sob o pseudônimo de Hatoshi Sakamoto. Por funcionar de maneira descentralizada, algumas de suas características serão utilizadas nesta proposta.

moedas entre os nós da rede, sendo que a mensagem apenas será enviada caso o nó destino tenha créditos suficientes para recompensar os nós retransmissores.

Para receber o seu pagamento, um nó deverá analisar todas as transações que estão em seu registro. Se o nó estiver na lista de nós retransmissores, o mesmo irá somar o número de créditos oferecidos como recompensa. Se o nó for o nó destino, deverá diminuir uma quantidade de créditos igual ao pago, ou seja, enquanto houver nós retransmissores, todas as transações terão seus créditos atualizados, como pode ser observado no Algoritmo 1.

```
Somatório de Créditos
{
    Enquanto houver nós retransmissores {

        Transação t = próxima transação da lista

        créditos = (CréditosRecebidos x NúmeroRetransmissores -CréditosPagos);
    }
retorna créditos;
}
```

Algoritmo 1: Algoritmo do somatório de créditos

Este procedimento também ocorrerá com os outros nós que participarem da transação, para que o nó que está atualizando os créditos saiba quantos créditos cada nó possui, ou seja, independente de ser o nó emissor, o nó destino, ou mesmo um nó retransmissor, deverá ocorrer atualização no registro do mesmo com estas informações, e caso seja um novo nó na retransmissão, o mesmo deverá ser incluído na lista de nós, como pode ser observado no Algoritmo 2.

```
Atualização dos Créditos
{
        Enquanto houver nós retransmissores{
                Transação t=próxima transação da lista;
                        Se nó==destino
                                Atualizar o número de créditos do nó destino;
                        Se nó==emissor
                                Atualizar o número de créditos do nó emissor;

                        Se nó==retransmissor
                                Atualizar o número de créditos do nó retransmissor;

                        Senão
                                Adicionar o nó na lista com o crédito;
        }

}
```

Algoritmo 2: Algoritmo da atualização dos créditos

3.2.2 Funcionamento do mecanismo

O funcionamento do DiCent começa após receber do protocolo de roteamento a lista de mensagens a serem enviadas reorganizada conforme a politica pré-determinada pelo protocolo de roteamento (no caso deste trabalho o ProPHET, ou o MaxProp [Burgess et al 2006]). Para poder funcionar corretamente, o DiCent necessita que o protocolo de roteamento forneça as informações sobre o número de saltos percorrido pela mensagem até a entrega e a lista de nós retransmissores. Essas informações são necessárias para a distribuição da recompensa e a cobrança do nó destino, assim como é realizado nos mecanismos de incentivo MuRIS e MobiCent.

O DiCent realiza uma comparação do Registro dos nós envolvidos na transmissão da mensagem. Estes registros (implementados como uma lista de transações) contêm a gravação do histórico de transações para que os nós possam ter controle de todos os créditos existentes na rede, de forma distribuída e autônoma, a qual é semelhante às tabelas de *rank* utilizadas por [Uddin et al. 2010], no RELICS. Essa comparação inicial é realizada para que, ao trocar as informações sobre os registros, os nós tenham o conhecimento do número de créditos que um nó vizinho tem antes de enviar uma mensagem. Isto tem por objetivo impedir que um nó recebesse uma mensagem mesmo sem ter créditos para pagar pelo recebimento.

Para isso, foi implementada a classe Transações contendo as informações acerca de quais nós participaram do encaminhamento da mensagem, o momento da entrega da

mensagem e também da quantidade de créditos que foram pagos e recebidos como recompensa pela entrega da mensagem. Para saber quantos créditos um nó possui, deve-se olhar em todas as transações, em quais delas o nó participou como retransmissor e em qual o mesmo foi o destino da mensagem. Ao somar os créditos ganhos, e subtrair os créditos perdidos, é possível determinar quantas moedas o nó possui, e assim tomas a decisão de encaminhar ou não a mensagem.

O DiCent realiza as seguintes verificações antes de entregar a mensagem ao nó destinatário: a possibilidade de serem criados novos créditos, a existência de créditos para pagar pela transmissão da mensagem e o número de saltos já percorridos pela mensagem.

Por sua vez o nó destinatário envia uma lista com as informações referentes às mensagens já entregues, antes da entrega da mensagem, para que possa calcular a quantidade de créditos a ser paga aos nós que participaram da operação. Antes de enviar uma mensagem ao nó destinatário, caso todas as verificações sejam bem sucedidas, o mecanismo solicita ao protocolo de roteamento que a mensagem seja entregue, e essa operação é incluída no registro.

A mensagem deverá conter informações necessárias para inserir a transação no registro, como informação do nó emissor, a lista de nós retransmissores e o momento em que a mensagem foi entregue, para que o mecanismo possa funcionar de forma distribuída. Caso a mensagem seja enviada com sucesso, os dois nós deverão inserir esta operação em seus registros. Se a conexão entre os nós cair antes do fim da transmissão de uma mensagem, a operação não será incluída no registro. Se a transmissão for efetuada com sucesso, ambos os nós poderão adicionar a operação mesmo que a conexão caia logo em seguida, já que as suas cópias da mensagem possuem todas as informações necessárias para isto.

O DiCent possui algumas restrições: controle da quantidade de saltos que uma mensagem poderá tentar durante a transmissão da mensagem em 6 saltos, limite de criação de 64 créditos por hora créditos. O objetivo destas restrições é prevenir que um nó ataque a rede gerando uma quantidade muito grande de créditos, e impedindo que os outros nós recebam as suas mensagens.

Um grande problema do mecanismo proposto está no fato de que para impedir que os nós destino sejam beneficiados com um ataque de *Edge Insertion* seria necessário cobrar dos nós destino uma quantidade de moedas virtuais superior do que a quantidade de moedas que será distribuída como incentivo para os nós retransmissores. Isto

causaria uma redução da quantidade de créditos existentes na rede. Esta perda de créditos obrigaria o mecanismo de incentivo a estar sempre criando novas moedas para suprir as que foram perdidas.

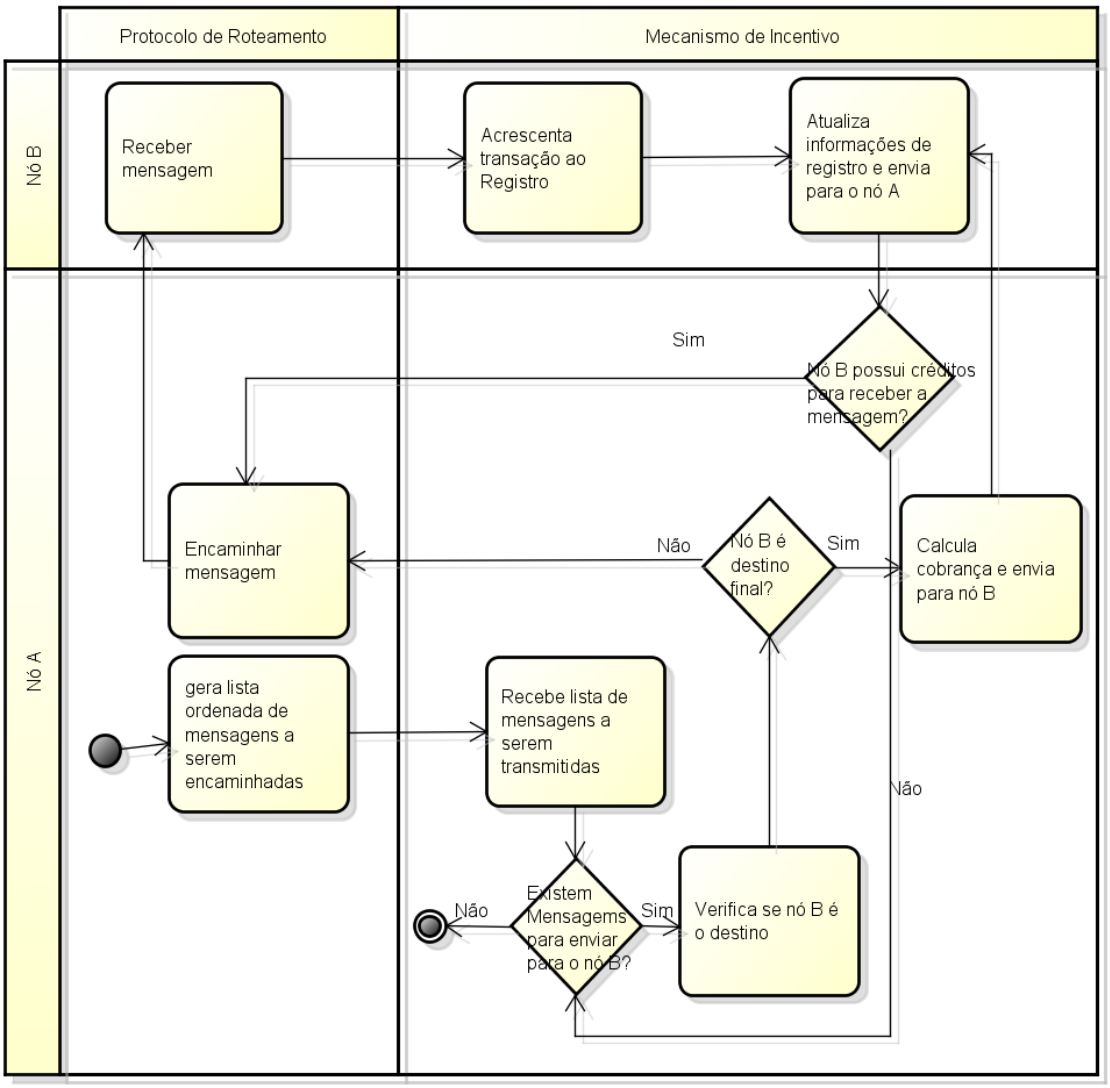


Figura 3: Funcionamento do DiCent

3.3 Formalização matemática do mecanismo proposto

Para demonstrar a validade do mecanismo proposto, assim como provar que o DiCent incentiva a colaboração dos nós, e não permite que os nós retransmissores se beneficiem com um ataque do tipo *Edge Insertion*, foi feita uma formalização matemática nesta seção.

O mecanismo proposto utilizará uma fórmula baseada nos lemas e teoremas desenvolvidos por [Chen and Chan 2010] e [Wang et al 2014].

Para validar matematicamente o mecanismo proposto, neste trabalho, utilizaremos as seguintes definições:

N = Número máximo de nós que podem participar da entrega da mensagem, onde: $\forall N \in \mathbb{N}$ e $N > 1$.

n = Número de nós que efetivamente participaram da entrega da mensagem, onde $\forall N \in \mathbb{N}$ e $N \geq n \geq 1$.

$R_{(n)}$: Recompensa que cada nó receberá se *n* nós participarem da retransmissão.

$C_{(n)}$: Quantidade de créditos cedidos pelo nó destino aos nós retransmissores, se *n* nós participarem da retransmissão.

3.3.1 Recompensa para os nós retransmissores:

Para cada nó que participar da entrega da mensagem a recompensa é igual a:

$$\mathbb{R}_{(n)} = 2^{N-n}$$

Equação 1: Recompensa para nós retransmissores

Cobrança do nó destino: O nó destinatário deverá pagar uma quantidade de créditos exatamente igual à soma dos valores recebidos como recompensa oferecida a todos os nós retransmissores.

$$\mathbb{C}_{(n)} = \sum_{k=1}^{n} \mathbb{R}_{(n)}$$

Equação 2: Cobrança realizada ao nó destino

Exemplo:

Partindo do pressuposto que o total de nós N = 10 e o total dos nós participantes são *n* = 4, a quantidade de créditos recebidos como recompensa por cada nó retransmissor é $2^{10-4} = 2^6 = 64$ créditos.

3.3.2 Teoremas

Teorema 1: Para impedir que os créditos sejam criados de forma descontrolada, causando uma perda do valor dos créditos, ou que os créditos sejam perdidos durante o pagamento do incentivo, o valor cobrado ao nó destino será sempre igual ao valor

oferecido como recompensa a todos os nós que efetivamente participaram da retransmissão da mensagem conforme a equação abaixo:

$$\mathbb{C}_{(n)} = n \times 2^{N-n}$$

Equação 3: Cobrança do nó destino sem perda de moedas

Prova:

Base da Indução:

Como base para a indução matemática, utilizaremos um valor de n=1. Substituindo estes valores na equação anterior temos como base para a indução 2^{N-1}=1 x $^{2N-1}$. Como esta equação é verdadeira para qualquer valor de N, a Base da Indução está validade.

Hipótese Indutiva:

$\sum_{k=1}^{k} 2^{N-k} = k \times 2^{N-k}$

= k x 2^{N-k}= k x 2^{N-k}

Prova de k+1:

Ao substituir o k na Hipótese Indutiva por k+1, temos que $\sum_{k=1}^{k+1} 2^{N-(k+1)} = (k+1) \times 2^{N-(k+1)}$. Resolvendo o somatório, temos que (k+1) x $2^{N-(k+1)}$ = (k+1) x $2^{N-(k+1)}$. Como esta equação é verdadeira para qualquer valor de k, o Teorema 1 está provado como gostaríamos de demonstrar.

Para provar este teorema por indução matemática, será utilizada a base de (*n*=1). Substituindo os valores de acordo com as equações acima, temos: 1 x $2^{N-1} \geq$ 1 x 2^{N-1}, validando a base da indução. Como Hipótese Indutiva, será utilizado *k* x $2^{N-k} \geq k$ x 2^{N-k}. Ao substituir por (k+1) teremos (*k*+1) x $2^{N-(k+1)} \geq$ (*k*+1) x $2^{N-(k+1)}$. Como ambos os lados da inequação são iguais, o Teorema 1 está provado como queríamos demonstrar.

Teorema 2: Os nós retransmissores não devem ter incentivo ao utilizar um ataque de *Edge Insertion*.

Lema 1: Para que um nó retransmissor não seja incentivado a utilizar um ataque de *Edge Insertion*, $R_{(n)} \geq R_{(n+1)}$.

Prova:

Como visto nas definições, a recompensa recebida por cada nó retransmissor é de $R_{(n)}$. Ao utilizar um ataque de *Edge Insertion*, o nó atacante irá inserir um nó *sybil* na lista de nós retransmissores para ganhar uma quantidade de créditos como recompensa igual à soma obtida pelo nó atacante e o nó sybil. Com a inserção do nó sybil, o número

de nós retransmissores aumentou para (n + 1). Com isso a recompensa obtida por cada nó mudou para $R_{(n+1)}$. Para que o nó atacante não se beneficie, a quantidade de créditos recebida pelo nó atacante mais a quantidade de créditos recebida pelo nó sybil deve ser igual ou menos do que a quantidade que o nó receberia se não utilizasse o ataque, ou seja:

$$\mathbb{R}_{(n)} \geq \mathbb{R}_{(n+1)}$$

Equação 4: Recompensa para evitar ataques de Edge Insertion

Prova:

Prova por indução. Para a base da prova por indução matemática será utilizado n = 1. Substituindo os valores nas equações anteriores temos $2^{N-1} \leq 2 \times 2^{N-(1+1)}$, Diminuindo em 1 o expoente da primeira parte da inequação, colocando um 2 em evidencia, temos $2 \times 2^{N-2} \leq 2 \times 2^{N-2}$, o que invalida a base da indução.

Para a hipótese indutiva será utilizada a inequação $2^{(N-k)} \leq 2x2^{N-(k+1)}$.

Para a prova de K + 1, na inequação e $2^{N-(k+1)} \leq 2 \times 2^{N-(k+2)}$ basta novamente modificar a primeira parte da inequação para $2 \times 2^{N-(k+2)} \leq 2 \times 2^{N-(k+2)}$ provando o teorema como queríamos demostrar.

Teorema 3: O nó destino não deve ter incentivo ao utilizar um ataque de *Edge Insertion*.

Prova:

Como o mecanismo proposto não é capaz de impedir que o nó destino se beneficie com um ataque de *Edge Insertion* será fornecido um contraexemplo para demonstrar isto.

Contraexemplo: Supondo que n = 2 e N = 5. A recompensa recebida por cada nó que participar da retransmissão será de 2^{5-2} = 8. A cobrança do nó destino será de 16 créditos. Ao realizar o ataque, a recompensa cedida a cada nó será de 2^{5-3} = 4, e a cobrança será de 12 créditos.

Como a cobrança do nó destino ao realizar o ataque é menor do que a cobrança ao não realizar o ataque, o mecanismo não é capaz de impedir que o nó destino se beneficie com um ataque de *Edge Insertion*, como queríamos demonstrar.

Teorema 4: Nenhum nó deve ser incentivado a realizar ciclos no encaminhamento da mensagem.

Prova:

Para demonstrar que nenhum nó poderia se beneficiar com este tipo de ataque, deve-se considerar que cada nó receberá 2^{N-n} como recompensa.

Caso a mensagem seja repassada duas vezes por todos os nós do ciclo, ao invés de apenas uma vez, o número de nós retransmissores será aumentado em k. Com isto, cada nó retransmissor receberia $2^{N-(n+k)}$ de recompensa, e os nós do ciclo receberiam $2 \times 2^{N-(n+k)}$, que e igual à $2^{N-(n+k-1)}$.

Ciclo de um nó (k=1):

Se k = 1, a recompensa recebida será de 2^{N-n} que é a mesma recompensa que o nó receberia sem o ciclo.

Ciclo de mais de um nó (k>1):

Se k > 1, a recompensa recebida pelo no será de $2^{N-(n+k-1)}$. Como $2^{N-n} \geq k \times 2^{N-(n+k+1)}$, o nó não poderá se beneficiar com a criação de ciclos.

Como não é possível que um nó receba mais créditos com a criação de um ciclo, nenhum nó irá ser adicionado mais de uma vez na lista de nós retransmissores, como queríamos demonstrar.

3.4 Detalhamento da implementação do mecanismo DiCent no simulador The ONE

Para que o mecanismo DiCent funcionasse com o protocolo de roteamento ProPHET, foram implementadas as seguinte funções em Java no simulador de redes oportunistas The ONE:

- *routing.DicentProphet:* Classe principal do mecanismo de incentivo. É responsável pela distribuição e cobrança de créditos e pela manutenção do registro de transações.
    - *UpdateCredits()*: Esta função atualiza os créditos de todos os nós que constem no registro de operações. Isto é feito somando os créditos ganhos pelo nó em todas as mensagens que o mesmo retransmitiu e diminuindo o número de créditos relativo às mensagens que o nó recebeu.
    - *HasNode(DTNHost host)*: Busca um determinado nó em uma lista que contenha todos os nós conhecidos.
    - *AddNode(DTNHost host, double credit)*: Adiciona um nó em uma lista de nós conhecidos, com uma quantidade de créditos igual à variável credits.
    - *DeleteNode(DTNHost host)*: Apaga um nó da lista de nós conhecidos.
    - *UpdateNode(DTNHost host, double credits)*: Atualiza as informações sobre os créditos de um nó na lista de nós conhecidos. Para isto, esta função chama

os métodos *DeleteNode(DTNHost host)* e *AddNode(DTNHost host, double credit)*..

- *AddTransaction(Transacao t)*: Adiciona um objeto do tipo transação na variável register (List<Transacao>). Esta função também pode ser chamada com os parâmetros (Mensagem m, int recompensa, int pagamento).
- *calculateCredits(DTNHost host)*: Calcula os créditos de um determinado nó *host* com base nas transações contidas no registro. Isto é feito somando toda a recompensa que o mesmo obteve como nó retransmissor, e subtraindo os créditos pagos pelo mesmo ao receber uma mensagem.
- *GetReward(int hopCount)*: Retorna a recompensa de um nó, com base na quantidade de saltos que uma determinada mensagem teve antes de ser entregue.
- *GetPayment(int hopCount)*: Retorna a quantidade de créditos que um nó deve oferecer como pagamento para receber uma mensagem.
- *sumCoins()*: Essa função retorna a quantidade de moedas existentes na rede, com base nas moedas recebidas e pagas nas transações existentes no registro de transações.
- *PredCredist(int retransmissores, double probabilidade)*: Retorna a quantidade esperada de créditos a serem recebidas. Esta métrica é baseada na quantidade de créditos a ser recebida com a entrega e uma métrica fornecida pelo protocolo de roteamento. Neste caso, foi utilizada a fração de entrega fornecida pelo protocolo ProPHET.
- *GetRegister()*: Retorna a variável register (List<Transacao>).
- *HasTransaction(Transacao t)*: Procura por todo o registro de transações (register) por uma transação t específica. Esta função retorna verdadeiro, caso a transação seja encontrada, e falso caso a transação ainda não tenha sido adicionada.
- *UpdateRegister(List<Transacao> reg)*: Atualiza o registro de operações (*List<Transacao>* register) com base em um registro de operações de um nó vizinho *(List<Transacao> reg)*.
- *OrderMessages(List<Tuple<Message, Connection>> messages)*: Ordena as mensagens contidas na variável *messages* a serem enviadas através de uma conexão com base na expectativa de recebimento de créditos fornecida pela função *PredCredist(int retransmissores, double probabilidade).*

- *GetMessages()*: Retorna uma lista de mensagens e conexões List<Tuple<Message, Connection>>messages a serem enviadas de acordo com o protocolo de roteamento. Neste caso, somente foram adicionadas as mensagens cujo nó vizinho possuía uma maior fração de entrega.
- *SendMessages()*: Começa a enviar mensagens através de uma conexão para um nó vizinho.
- *checkCoins(double coins, DTNHost host)*: Verifica se um nó vizinho host possui crédito suficiente para receber uma mensagem.
- *UpdateRestrictions()*: Atualiza as restrições para a criação de créditos na fase de criação de créditos.

- *Routing*.dicent.Transacao: Classe que representa o objeto Transação, que constitui nos elementos que formam o registro. Esta classe possui as seguintes funções:
  - *getTime()*: Retorna a hora em que a transação ocorreu, armazenado na variável *time.*
  - *setTime(double time)*: Modifica o valor da variável time para o fornecido pelo parâmetro da função.
  - *getEmissor()*: Retorna o nó emissor.
  - *setEmissor(DTNHost emissor)*: Modifica o valor do nó emissor para o fornecido pelo parâmetro da função.
  - *getDestino()*: Retorna o nó que recebeu a mensagem.
  - *setDestino(DTNHost destino)*: Modifica o valor do nó destino para o fornecido pelo parâmetro da função.
  - *getRetransmissores()*: retorna uma lista (List<DTNHost>) com os nós que efetivamente participaram da retransmissão da primeira entrega da mensagem.
  - *setRetransmissores(*List<DTNHost> retransmissores*)*: Atribui uma lista de nós para a variável retransmissores.
  - *getCreditosPagos():* Retorna a quantidade de créditos oferecida como recompensa pelo nó destino.
  - *setCreditosPagos(int creditosPagos):* Atribui um novo valor à quantidade de créditos gasta pelo nó destino.
  - *getCreditosRecebidos():* Retorna a quantidade de créditos paga a cada um dos nós retransmissores.

- *setCreditosRecebidos(int creditosPagos):* Modifica a quantidade de créditos recebida pelos nós retransmissores.

- *Routing.TupleComparatorDiCent*: Classe responsável por comparar as tuplas *Tuple<Message, Connection>* de acordo com a função *OrderMessages(List<Tuple<Message, Connection>> messages)* para que o nó possa transmitir as mensagens pela função *SendMessages()* de forma que receba o maior número de créditos por mensagem enviada.

4 Avaliação de desempenho do mecanismo proposto

Este capítulo apresenta uma avaliação do mecanismo proposto, realizada através do método simulação, para que seja possível apresentar os resultados obtidos ao se utilizar o DiCent, com o auxílio de um simulador especialmente desenvolvido para redes oportunistas, utilizando *traces* de contato reais. Serão avaliadas as métricas da fração de entrega, da sobrecarga média (*overhead)* e do atraso médio de transmissão. O mecanismo proposto será avaliado através de diversas simulações que serão descritas ao final desta seção. Para isto, foi escolhido o simulador de redes oportunistas The ONE.

Serão realizadas comparações entre o DiCent, mecanismo apresentado nesta proposta, e o mecanismo de incentivo desenvolvido por [Uddin et al. 2010], o RELICS, mecanismo este escolhido por funcionar com qualquer protocolo de roteamento. Para a avaliação serão utilizados os protocolos de roteamento ProPHET e MaxProp. Serão utilizados traces de contato reais, extraídos do projeto CRAWDAD, já descrito anteriormente, para simular a movimentação e o contato entre os nós.

4.1 O projeto CRAWDAD

Conforme [Scott et al 2006], CRAWDAD (*Community Resource for Archiving Wireless Data At Dartmouth*,) é um repositório de dados de rede sem fio de dados para a comunidade científica. Este arquivo tem a capacidade de armazenar dados de rastreamento sem fio de muitos locais contribuintes, além de pessoal para desenvolvimento dos melhores instrumentos de coleta, anonimamente, e análise dos dados. Este trabalho é desenvolvido com os líderes comunitários para garantir que o arquivo atende às necessidades da comunidade de pesquisa.

4.2 O simulador utilizado

As simulações foram realizadas através do simulador The ONE (*Opportinistic Network Environment* [Keranen et al. 2009]. Assim como no protocolo proposto, a linguagem utilizada para desenvolvê-lo foi Java.

Tal simulador foi escolhido por suas características referentes à simulação de redes oportunistas, bem como a possibilidade de utilizar traces de mobilidades reais. Outro motivo para a sua utilização é o fato deste simulador ser largamente utilizado para validar redes DTN, tendo sido utilizado na maior parte dos artigos presentes na literatura.

Em [Villares et al, 2013] são citadas outras características deste simulador, tais como o suporte a diferentes protocolos de roteamento, como, por exemplo, o *First Contact, Direct Delivery, Spray and Wait,* Epidêmico, ProPHET e MaxProp, além da possibilidade de que, caso seja necessário, o The ONE também permite a simulação com protocolos desenvolvidos externamente, os quais podem ser importados, ou até mesmo a implementação de novos protocolos no próprio simulador. Para esta proposta, foram utilizados os protocolos de roteamento ProPHET e MaxProp.

4.3 Descrição do cenário utilizado nas simulações

Para a realização das simulações, foi usado o The ONE, versão 1.4.1, com os protocolos de roteamento ProPHET e MaxProp:

Para avaliar o mecanismo proposto, foi escolhido um cenário do repositório do CRAWDAD, o *dataset* INFOCOM 05[Scott et al 2006], extraído durante a INFOCOM de 2005, cuja coleta de dados representa o período entre os dias 7 e 10 de Março de 2005. Foram criadas 1000 mensagens durante as simulações, cada uma com 2,5Kb de tamanho. O buffer dos nós que compõem a rede foi definido como 100Mb, para evitar que houvesse descarte por sobrecarga do mesmo.

Configuração	Cenário INFOCOM 5
Quantidade de nós do trace	41 nós
Número total de mensagens trocadas	1000 mensagens
Tamanho do Buffer	100M

Tamanho de cada mensagem	2,5K
TTL	15h, 24h, 36h, 48h.
Tempo de simulação	275000s

Tabela 2: Configuração do cenário INFOCOM utilizado para realizar as simulações

4.4 Métricas de desempenho

As métricas escolhidas para comparação entre os mecanismos de incentivo foram a fração de entrega, o atraso médio e o *overhead*, onde:

Fração de entrega da mensagem: razão entre as mensagens que foram entregues aos nós destinos e as mensagens que foram efetivamente criadas pelos nós emissores. Portanto, as cópias das mensagens que foram criadas durante o processo de encaminhamento pelos nós retransmissores não são contadas por esta métrica.

Overhead (sobrecarga) médio para entrega da mensagem: mede quantas mensagens foram criadas pelos nós retransmissores para que as mensagens sejam entregues. Essa métrica é calculada ao diminuir a quantidade de mensagens retransmitidas (*messages relayed*) pelo número de mensagens entregues, e então dividir o resultado pelo número de mensagens entregues.

Atraso médio na entrega da mensagem: média entre o intervalo de tempo entre o momento em que uma mensagem é criada até o momento em que a mesma é entregue, ou seja: Primeiro é calculado o atraso de cada pacote, subtraindo o tempo em que a mesma foi entregue do momento em que a mesma foi criada. Este valor de atraso é somado para cada mensagem entregue, e depois, a soma dos atrasos individuais é dividida pelo número de mensagens entregues.

4.5 Resultados obtidos

Após realizar diferentes simulações, foram feitas comparações entre os resultados obtidos ao se utilizar o mecanismo proposto com outro mecanismo já existente, o RELICS para que fossem comparados aos protocolos de roteamento escolhidos, quando estes apresentam comportamento egoísta. O RELICS foi escolhido para ser comparado com o mecanismo proposto pelo fato do RELICS possuir uma compatibilidade com diversos protocolos de roteamento propostos pela literatura.

4.5.1 Resultados obtidos em simulações utilizando um trace de mobilidade real em um cenário de conferência (INFOCOM)

O primeiro passo foi comparar o comportamento dos mecanismos de incentivo em relação ao cenário no qual é considerado o comportamento de nós egoístas.

Para representar a ação de nós egoístas durante as simulações, foram utilizados nós frugais (*Frugal nodes*), assim como no trabalho de [Shevade et al. 2008] e [Uddin et al. 2010]. Estes nós apresentam uma quantidade extrema de egoísmo, e apenas retransmite suas próprias mensagens, se recusando a enviar mensagens que tenham sido criadas por outros nós. No trabalho de [Shevade et al. 2008], é dito que, caso todos os nós sejam frugais, o comportamento da rede se degenera, e a entrega de mensagens ocorre apenas de forma direta, ou seja, as mensagens apenas seriam entregues caso o nó destino entre em contato com o nó emissor.

Como este cenário seria compatível com o uso do protocolo de roteamento por entrega direta, e não representaria um cenário em que apenas alguns nós apresentariam um comportamento egoísta, para ilustrar o efeito do egoísmo nos protocolos de roteamento foi utilizado o mesmo padrão de egoísmo apresentado por [Uddin et al. 2010]. Neste trabalho, os nós múltiplos de 5, começando pelo nó 0, foram escolhidos para serem nós frugais. Este número de nós frugais foi escolhido, pois, caso o número de nós egoístas fosse muito reduzido, os efeitos dos nós egoístas na rede não seriam tão evidentes. Da mesma forma, caso o número de nós egoístas fosse muito grande, a rede se comportaria de forma idêntica ao uso do protocolo de roteamento por entrega direta, não permitindo observar a diferença entre os efeitos do egoísmo nos diferentes protocolos de roteamento estudados.

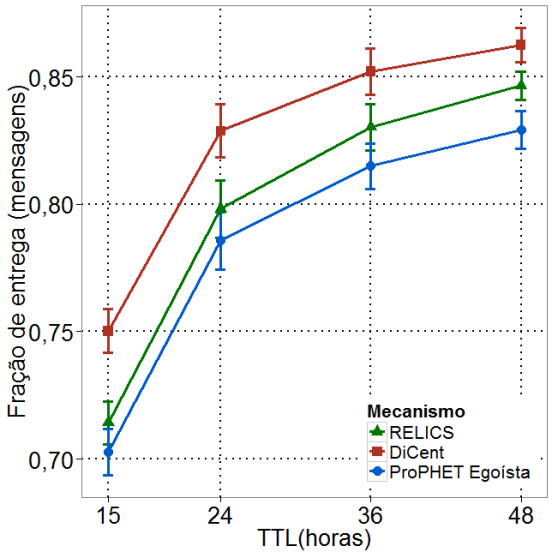


Figura 4: Média da fração de entrega utilizando o protocolo de roteamento ProPHET junto com os mecanismos de incentivo DiCent e RELICS

Na Figura 4 é possível observar que, devido à existência de nós egoístas no cenário utilizando o ProPHET Egoísta, o fato destes nós não encaminharem as mensagens que foram criadas pelos outros nós da rede causaram uma redução na fração de entrega, sendo esta a mais baixa dos três.

Mesmo com a restrição ao envio de mensagens causado pelo *rank* dos nós utilizado pelo RELICS, esse mecanismo de incentivo foi capaz de incentivar a colaboração dos nós, apresentando uma fração de entrega superior ao ProPHET egoísta. Isso significa que, mesmo com os nós utilizando o RELICS darem prioridade aos nós com um *rank* mais alto, e se recusando a enviar mensagens com nós de *rank* 1 ou menor, a rede ainda assim conseguiu uma fração de entrega maior do que no cenário egoísta.

A limitação de saltos imposta ao mecanismo proposto neste cenário foi suficiente para que fosse possível apresentar uma maior fração de entrega do que a apresentada pelo mecanismo de incentivo RELICS, A diferença entre os dois mecanismos de incentivo foi sendo reduzida à medida que o TTL foi aumentado, pois com um tempo de vida maior no RELICS, foi possível que os nós aumentassem o seu *rank*, podendo assim enviar mais mensagens. Caso todos os nós possuíssem um *rank* grande o suficiente para enviar todas as suas mensagens, a sua fração de entrega seria igual à do DiCent sem limite de créditos e saltos.

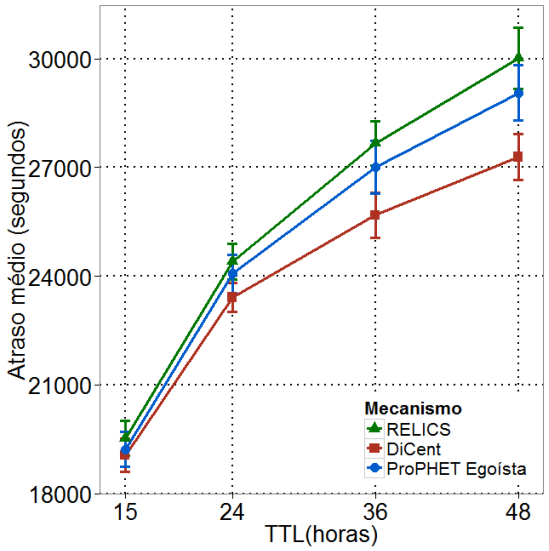


Figura 5: Atraso médio na entrega de mensagens utilizando o protocolo ProPHET junto com os mecanismos de incentivo DiCent e RELICS

Como as oportunidades para a entrega de mensagens aumentam conforme o tempo de vida dos pacotes, nas simulações realizadas com um TTL baixo apresentou um atraso médio semelhante, já que apenas as mensagens que podiam ser entregues mais rapidamente tiveram o seu atraso contabilizado, conforme pode ser observado na Figura 5.

Com a limitação de saltos imposta pelo mecanismo DiCent, as mensagens que necessitariam de um maior número de saltos para serem entregues, e por isto também possuiriam um atraso médio maior, foram descartadas. Por isto, este mecanismo apresentou o menos atraso médio nos cenários analisados, principalmente ao se utilizara um tempo de vida dos pacotes mais alto. No caso dos cenários utilizando o ProPHET egoísta, a baixa fração de entrega apresentada devido ao comportamento dos nós também impediu que as mensagens que teriam um atraso de entrega maior, mas que necessitariam utilizar os nós egoístas em sua rota fossem entregues. Como a restrição de entrega do RELICS é baseada no *rank* dos nós, e não impede que mensagens com um atraso maior sejam entregues, este mecanismo apresentou um maior atraso médio que os demais.

Conforme foi aumentado o tempo de vida dos pacotes, novas oportunidades de entrega foram surgindo, independente do mecanismo. No entanto, devido à característica do mecanismo RELICS de priorizar as entregas onde há um ganho de reputação maior, ao invés do DiCent que sempre prioriza os caminhos mais curtos, a

diferença entre o atraso dos dois mecanismos tende a aumentar junto com o tempo de vida dos pacotes.

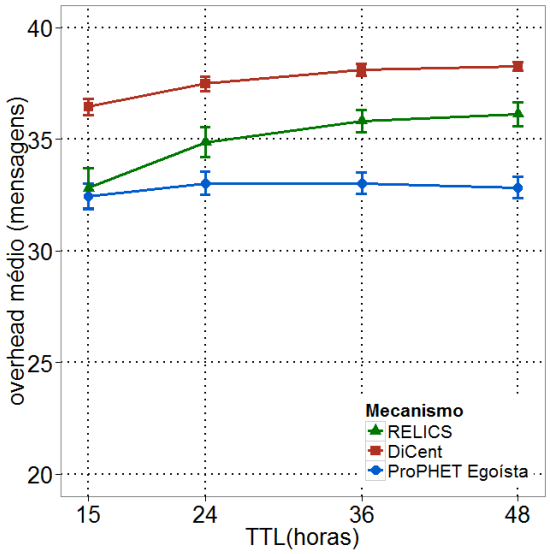


Figura 6: *Overhead* médio na entrega de mensagens utilizando o protocolo de roteamento ProPHET junto com os mecanismos de incentivo DiCent e RELICS

Nas simulações que utilizaram o ProPHET egoísta, o descarte de mensagens causado pelo comportamento egoísta dos nós causou uma grande redução no *overhead*, como observado na Figura 6. Já a restrição de saltos utilizada pelo mecanismo DiCent não foi capaz de reduzir o *overhead*, já que estas simulações apresentaram um *overhead* superior às demais. Caso fosse utilizado um valor menor para o limite máximo de saltos, poderia haver um impacto negativo na fração de entrega e no atraso médio, de forma que um *overhead* mais alto no mecanismo DiCent pode ser um custo necessário para a maior taxa de entrega e menor atraso médio.

Se o limite máximo de saltos fosse o maior possível, ou seja, igual ao número de nós da rede, o overhead encontrado tenderia ao *overhead* que seria encontrado ao se utilizar o protocolo epidêmico, e o gasto de créditos seria muito maior, devido às equações utilizadas para a cobrança e pagamento dos nós. A diferença entre o *overhead* encontrado entre os mecanismos de incentivo RELICS e DiCent se deve ao fato dos nós que utilizam o RELICS não encaminharem as mensagens cujos nós de origem tenham um *rank* menor ou igual a 1. Como nestes cenários houve um maior número de mensagens descartadas devido a um *rank* baixo no RELICS do que ao limite de saltos do mecanismo proposto, o RELICS obteve um *overhead* menor do que o DiCent, o que é um ponto negativo para o mecanismo proposto.

Para avaliar o comportamento dos nós egoístas e dos mecanismos de incentivo RELICS e DiCent caso seja utilizado um protocolo de roteamento diferente, foram feitas implementações de ambos os mecanismos de incentivo utilizando o protocolo de roteamento MaxProp. O número de nós frugais que utilizaram apenas o protocolo de roteamento MaxProp foi mantido o mesmo que o número utilizado nas simulações com o protocolo de roteamento ProPHET.

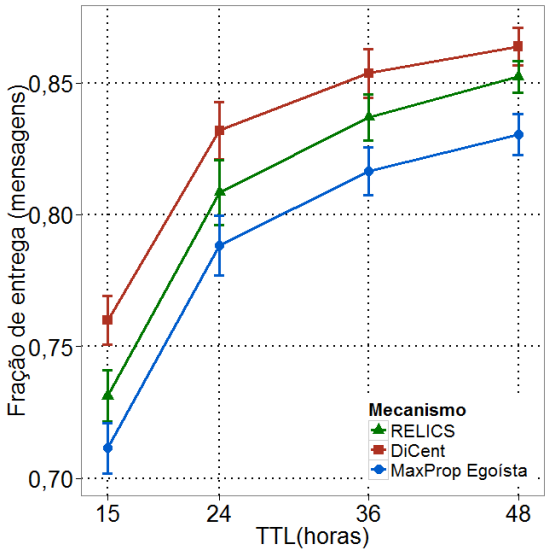


Figura 7: Fração de entrega de mensagens utilizando o protocolo de roteamento MaxProp junto com os mecanismos de incentivo DiCent e MaxProp

De maneira semelhante ao que ocorreu nos cenário que utilizaram o protocolo de roteamento ProPHET, o descarte de mensagens provocado pelo egoísmo dos nós nas simulações com o MaxProp egoísta provocou uma menor fração de entrega do que a encontrada ao se utilizar os mecanismos de incentivo RELICS e DiCent, o que justifica a utilização de um mecanismo de incentivo nos casos de existirem nós egoístas na rede.

O limite de número máximo de saltos utilizado pelo mecanismo proposto foi o suficiente para manter a fração de entrega superior à encontrada ao utilizar o mecanismo de incentivo RELICS. Esta vantagem é reduzida com o uso de tempos de vida mais altos, já que com o tempo existe uma tendência ao utilizar o mecanismo de incentivo RELICS de que os nós obtenham um valor de *rank* cada vez mais alto, o que os permite enviar um número crescente de mensagens.

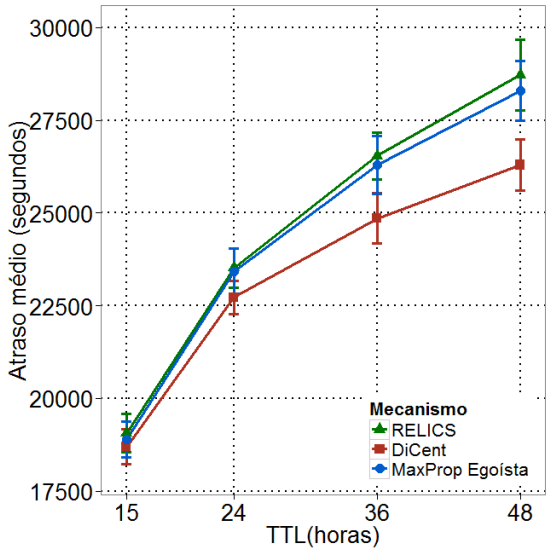


Figura 8: Atraso médio na entrega de mensagens utilizando o protocolo de roteamento MaxProp junto com os mecanismos de incentivo DiCent e RELICS

Em relação ao atraso médio, O limite de saltos do DiCent ainda foi o suficiente para incentivar os nós a colaborarem e mitigarem os efeitos do egoísmo, em relação ao protocolo de roteamento MaxProp com os nós egoístas. No entanto, as limitações de entrega em relação ao *rank* do mecanismo de incentivo RELICS provocaram um atraso médio superior aos outros cenários analisados. Isso é causado pela entrega de mensagens com um valor de atraso alto, emas que os nós de origem possuíam um valor de *rank* alto, e o descarte de mensagens com um valor de atraso baixo, mas com um nó de origem com um valor de *rank* igual ou inferior a 1.

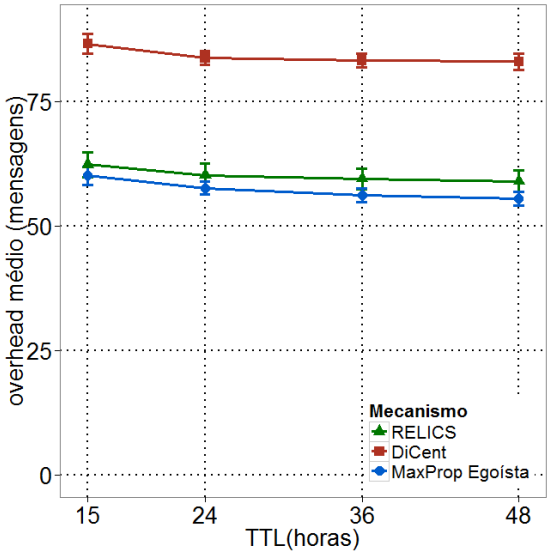


Figura 9: *Overhead* médio de mensagens utilizando o protocolo de roteamento MaxProp junto com os mecanismos de incentivo DiCent e RELICS

Devido aos cálculos de probabilidade de entrega e ao limite máximo de saltos para a entrega do DiCent, o mecanismo proposto apresentou um overhead muito acima do mecanismo RELICS. O MaxProp com nós egoístas apresentou o menor overhead, devido às mensagens descartadas pelos nós egoístas.

4.5.2 Resultados obtidos ao variar o número máximo de saltos permitidos no mecanismo de incentivo DiCent

Após as simulações realizadas na Seção 4.5.1, foram realizadas outras simulações para descobrir se ao modificar o número máximo de saltos permitidos pelo mecanismo DiCent, haveria algum impacto significativo nas métricas apresentadas nas simulações anteriores.

A primeira métrica avaliada nestas simulações foi a fração de entrega. Como é importante que o número de mensagens entregues seja o maior possível, a mesma será a base para avaliar se é possível utilizar um menor número de saltos permitido.

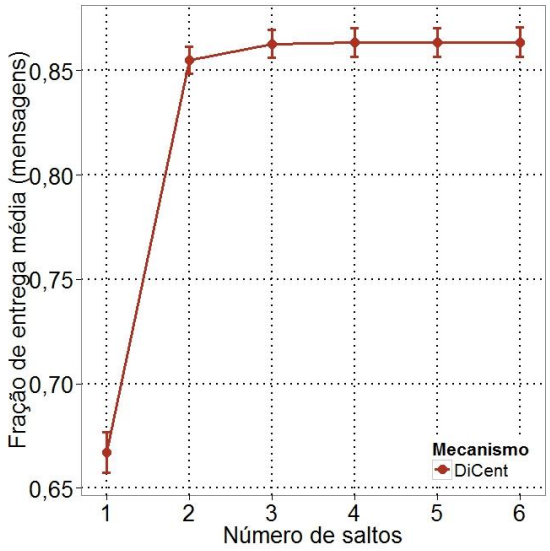


Figura 10: Fração de entrega do mecanismo de incentivo DiCent ao variar o número máximo de saltos

Como é possível observar na Figura 10, a fração de entrega é menor quando o número de saltos é igual a 1, o que significa que as mensagens serão obrigatoriamente entregues diretamente ao seu destino final. Ao aumentar o número máximo de saltos para 2, houve um grande aumento na fração de entrega, aproximadamente 0,85. Este valor obteve um aumento menor quando o número máximo de saltos foi modificado para 3, e se manteve constante para todos os valores iguais e maiores de 3.

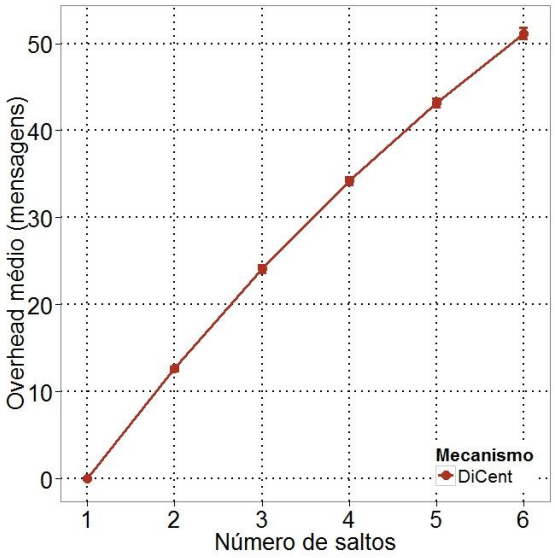


Figura 11: *Overhead* de mensagens do mecanismo de incentivo DiCent ao variar o número máximo de saltos

A Figura 11 ilustra que, ao aumentar o número máximo de saltos permitido pelo mecanismo de incentivo DiCent, o overhead de mensagens também aumenta. A métrica overhead continuou aumentando de forma quase linear, mesmo ao utilizar 9 saltos como o máximo permitido. Estes dados ilustram que, quanto menor for o número de saltos, melhor para a rede, já que irá reduzir a sobrecarga gerada pelas mensagens extras. No entanto, a redução do número de saltos também pode causar uma redução na taxa de entrega, sendo necessário avaliar, dado as características do cenário, até aonde é vantajoso restringir a sobrecarga de mensagens, ao custo da fração de entrega.

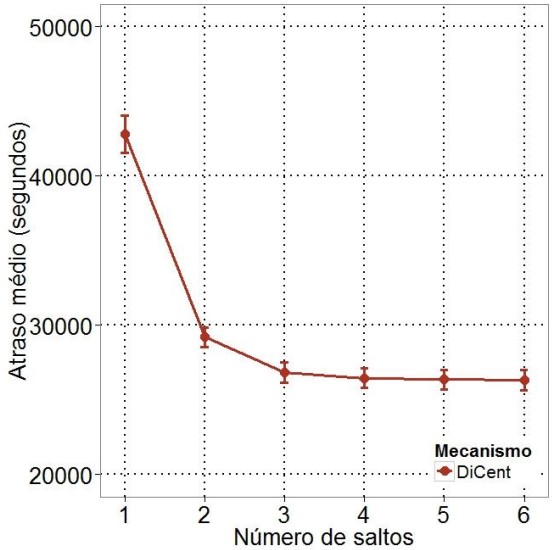


Figura 12: Atraso médio do mecanismo de incentivo DiCent ao variar o número máximo de saltos

Também foi avaliado o atraso médio. Como pode ser observado na Figura 12, quando se utiliza número máximo de saltos permitido igual a 1, o atraso médio atinge o seu valor máximo, ultrapassando os 40.000 segundos. Ao aumentar o número máximo de saltos permitido para 2, o valor do atraso é reduzido para menos de 30.000. O valor do atraso médio atinge o seu mínimo ao utilizar valores maiores ou iguais a 4 como número máximo de saltos permitido.

4.5.3 Resultados obtidos em simulações utilizando um trace de mobilidade real em um cenário de lazer (RollerNet)

Após terem sido realizadas simulações utilizando o *trace* de contatos extraídos de uma conferência foi escolhido outro cenário real para avaliar o mecanismo de incentivo proposto. O cenário RollerNet foi escolhido por ser altamente conexo, e por representar outro tipo de contato, no caso uma situação de lazer. Primeiro foram realizadas simulações utilizando o protocolo de roteamento ProPHET e os mecanismos de incentivo DiCent e RELICS. Com base nos resultados encontrados na Seção 4.5.2, foi escolhido 3 como número máximo de saltos permitido pelo DiCent. Todos os outros parâmetros das simulações permaneceram inalterados. Os dados sobre a configuração deste cenário podem ser observados na Tabela 3 a seguir

Configuração	Cenário Rollernet
Quantidade de nós do trace	62 nós
Número total de mensagens trocadas	1000 mensagens
Tamanho do Buffer	100M
Tamanho de cada mensagem	2,5K
TTL	36, 72, 108, 144. (min)
Tempo de simulação	9999 (segundos)

Tabela 3: Configuração do cenário RollerNet utilizado para realizar as simulações

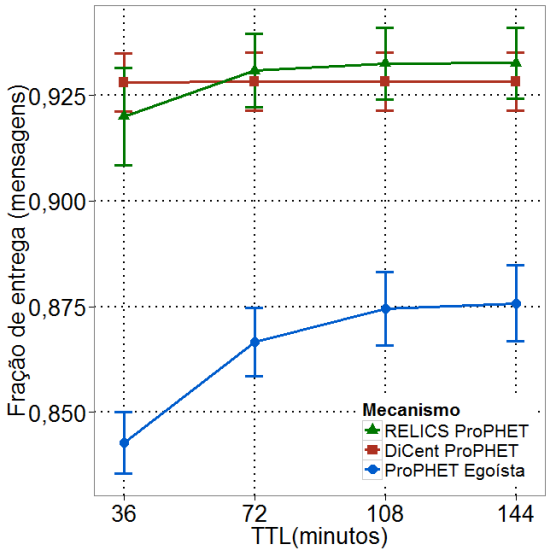


Figura 13: Fração de entrega de mensagens utilizando o protocolo de roteamento ProPHET junto com os mecanismso de incentivo DiCent e RELICS

Como é possível observar na Figura 13, ambos os mecanismos de incentivo, RELICS e DiCent, conseguiram incentivar a colaboração dos nós, aumentando a fração de entrega em relação ao obtido utilizado apenas o protocolo de roteamento ProPHET com nós frugais. O mecanismo de incentivo DiCent obteve um resultado mais consistente, o que pode ser notado pelos resultados da fração de entrega se manterem constantes, apesar da variação do tempo de vida dos pacotes. Neste cenário, apesar do mecanismo de incentivo RELICS aparentar um desempenho superior ao mecanismo proposto, com uma maior fração de entrega nas simulações que utilizaram um tempo de vida igual ou maior do que 72 minutos, ambos os mecanismos de incentivo obtiveram resultados muito semelhantes, pois a maior parte dos resultados se encontra dentro do intervalo de confiança.

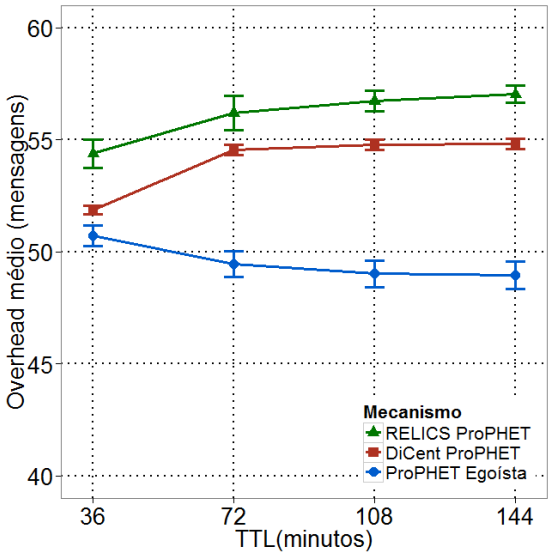


Figura 14: *Overhead* médio de mensagens utilizando o protocolo de roteamento ProPHET junto com os mecanismso de incentivo DiCent e RELICS

A Figura 14 ilustra os resultados obtidos da métrica *overhead* médio nas simulações. Nestas simulações, o DiCent foi capaz de obter um valor inferior ao obtido utilizando o mecanismo de incentivo RELICS. Isto ocorreu devido à restrição de número máximo de saltos, o que impediu que a mensagem fosse replicada para um número maior de nós.

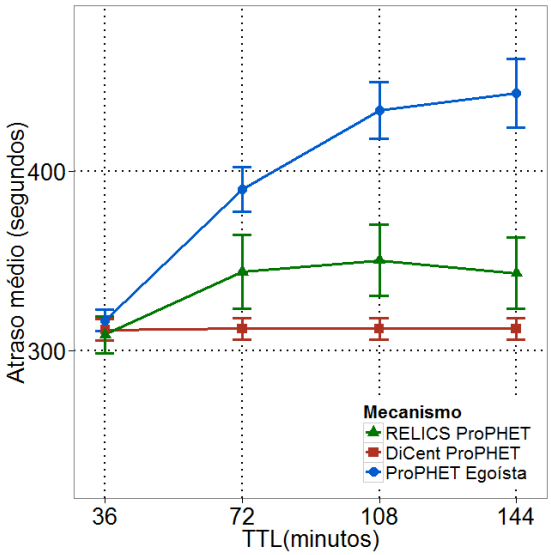


Figura 15: Atraso médio na entrega de mensagens utilizando o protocolo de roteamento ProPHET junto com os mecanismso de incentivo DiCent e RELICS

Como é possível observar na Figura 15, o atraso na entrega dos pacotes foi maior ao utilizar apenas o protocolo de roteamento com nós egoístas. O mecanismo proposto conseguiu não apenas incentivar mais a colaboração dos nós mantendo o atraso menor, como o manteve constante independente do TTL utilizado.

Após estas simulações, foram realizadas novas simulações com o protocolo de roteamento MaxProp.

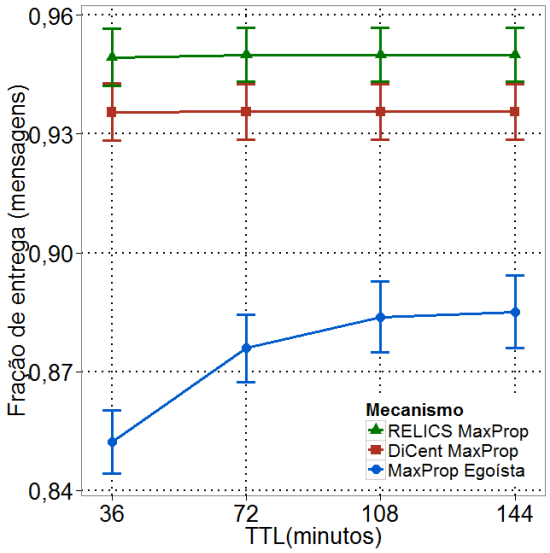


Figura 16: Fração de entrega de mensagens utilizando o protocolo de roteamento MaxProp junto com os mecanismso de incentivo DiCent e RELICS

A Figura 16 ilustra o comportamento dos mecanismos de incentivo RELICS e DiCent utilizando o protocolo de roteamento MaxPROP, e o mesmo com nós egoístas. Embora ambos tenham apresentado um comportamento semelhante em relação à evolução da fração de entrega ao aumentar o TTL, o mecanismo proposto ainda apresentou um valor abaixo do RELICS, o que é um ponto negativo para o mecanismo proposto.

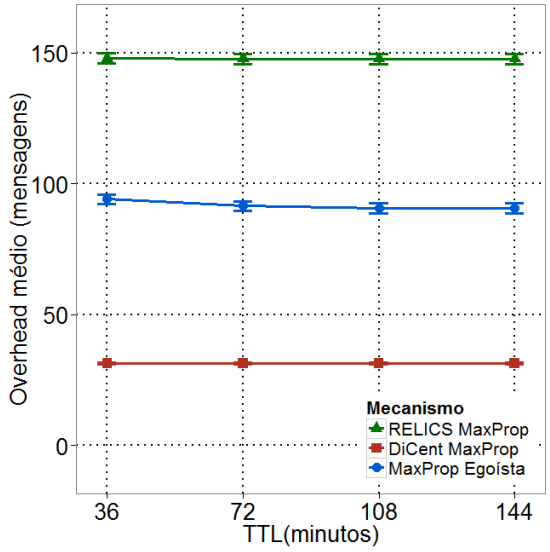


Figura 17: *Overhead* médio de mensagens utilizando o protocolo de roteamento MaxProp junto com os mecanismso de incentivo DiCent e RELICS

Após isto, foi analisada a métrica *overhead*, como ilustrado pela Figura 17. Esta métrica não apresentou uma diferença em relação à mudança do tempo de vida dos pacotes. No entanto, o mecanismo DiCent conseguiu manter um *overhead* mais baixo que o apresentado pelo protocolo de roteamento MaxProp com nós frugais. Isto indica que a restrição ao número de saltos máximo permitido foi capaz de reduzir o *overhead* gerado ao impedir que fossem criadas mais cópias. No entanto, tal redução não impactou negativamente no incentivo à colaboração oferecido aos nós, uma vez que o mecanismo DiCent apresentou uma maior fração de entrega.

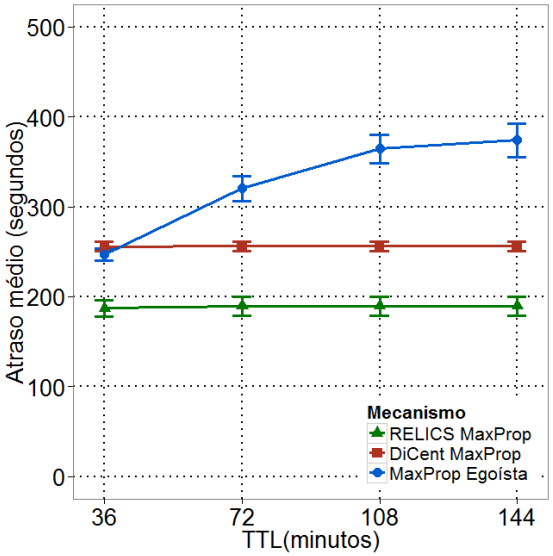


Figura 18: Atraso médio na entrega de mensagens utilizando o protocolo de roteamento MaxProp junto com os mecanismos de incentivo DiCent e RELICS

A Figura 18 ilustra que ambos os mecanismos de incentivo obtiveram um atraso constante, independente do tempo de vida dos pacotes. O RELICS obteve um atraso menor em relação ao DiCent, ao custo de um *overhead* maior, como foi observado na Figura 21. Embora o protocolo de roteamento MaxProp tenha obtido um valor próximo ao do mecanismo de incentivo DiCent com um TTL de 36 minutos, o egoísmo dos nós provocou o aumento no atraso da entrega ao aumentar o TTL.

5 Conclusão e sugestões para trabalhos futuros

Neste trabalho foi observada a importância de se utilizar mecanismos de incentivo para incentivar nós egoístas a colaborar em redes DTNs. Os efeitos do egoísmo em uma rede DTN foram avaliados ao se utilizar de nós frugais e comparar o desempenho do protocolo de roteamento com estes nós egoístas e o desempenho que o protocolo de roteamento apresentou junto com os mecanismos de incentivo. Foram utilizados dois protocolos de roteamento durante estas simulações, o ProPHET e o MaxProp. Também foram vistos diversos ataques possíveis a estes mecanismos, e outras propostas para incentivar nós egoístas presentes na literatura.

A contribuição deste trabalho foi demostrar a possibilidade de se criar um mecanismo que incentive os nós a colaborarem através da distribuição de créditos sem que seja necessária a utilização de uma entidade central externa, algo difícil de ser feito em redes com frequentes desconexões e um grande atraso durante as transmissões.

No cenário do INFOCOM, o mecanismo proposto, DiCent, se mostrou capaz, conforme observado no Capítulo 4, de incentivar a colaboração dos nós para garantir a transmissão de mensagens. O DiCent foi comparado com o mecanismo RELICS [Uddin et al. 2010], e obteve uma melhor fração de entrega e um atraso médio menor, ao custo de um *overhead* maior. Ao utilizar o protocolo de roteamento ProPHET, o DiCent apresentou uma fração de entrega de 0,75 com o TTL de 15 horas, e o RELICS apresentou uma fração de entrega de 0,71. O mecanismo proposto manteve uma maior

fração de entrega ao aumentar o TTL, obtendo uma fração de entrega de mais de 0,85. Como estes valores se mantiveram muito próximos mesmo ao utilizar o protocolo de roteamento MaxProp, esta diferença foi causada pelo fato de que, mediante o comportamento egoísta apresentado pelos nós que utilizam o mecanismo de incentivo RELICS, mensagens de nós com *ranks* muito baixos não foram retransmitidas, ao contrário do mecanismo proposto, que apenas não transmite mensagens com uma contagem de saltos acima de um limite permitido.

Este comportamento também foi observado ao se analisar o *overhead* médio. Como o DiCent retransmite uma maior quantidade de mensagens, a sobrecarga média de mensagens medida nas simulações utilizando ambos os protocolos de roteamento foram menores quando utilizado o mecanismo de incentivo RELICS. Ao utilizar o protocolo de roteamento ProPHET, o *overhead* encontrado nas simulações com o mecanismo proposto variou de 36,5 até 38, dependendo do TTL utilizado, enquanto o *overhead* medido ao se utilizar o mecanismo de incentivo RELICS variou de 33 até 36 mensagens. Esta diferença se mostrou mais acentuada ao utilizar o protocolo de roteamento MaxProp, onde todas as medidas de *overhead* do DiCent foram maiores do que 80 mensagens e o mecanismo de incentivo RELICS não apresentou um *overhead* superior a 65 mensagens em nenhuma simulação. O mecanismo proposto também apresentou uma variância inferior a encontrada ao se utilizar o mecanismo RELICS. Isso pode ser explicado pelo fato de que o *rank* do nó de origem da mensagem teve um impacto maior na decisão de encaminhar a mensagem do que o número de saltos, métrica utilizada pelo DiCent para a escolha da mensagem a ser encaminhada.

A última métrica analisada foi o atraso médio. Embora esta tenha sido a métrica em que os dois mecanismos de incentivo apresentaram resultados mais próximos, os gráficos ilustraram uma tendência que o mecanismo proposta apresentaria um atraso

médio menor em relação ao apresentado pelo mecanismo de incentivo RELICS ao se aumentar o TTL. Enquanto a diferença entre os atrasos apresentada ao utilizar um tempo de vida em cada pacote de 15 horas manteve os resultados dos dois mecanismos de incentivo dentro do intervalo de confiança, o DiCent apresentou um atraso médio menor em aproximadamente 3.000 segundos em relação ao atraso médio apresentado pelo mecanismo RELICS quando o TTL foi aumentado para 48 horas. Isso caracteriza o fato de que o DiCent seria mais apropriado para redes DTN em que seja necessário manter uma fração de entrega mais alta, e que o *overhead* não tenha um impacto significativo para justificar a escolha de outro mecanismo de incentivo.

Em simulações posteriores, foi comprovado que o *overhead* de mensagens causado pelo DiCent pode ser mitigado ao se restringir o número máximo de saltos para a entrega de uma mensagem. Embora isto possa causar uma redução na fração de entrega das mensagens, foi observado que é possível determinar um valor para o número máximo de saltos permitido pelo mecanismo de incentivo DiCent para que não haja uma redução na fração de entrega e nem no atraso médio. Esta característica do mecanismo de incentivo o torna mais flexível e permite que ele se adapte melhor as condições e exigências específicas de cada cenário, e ainda assim reduzindo o custo causado pela sobrecarga de mensagens na rede.

Nas simulações no cenário de lazer RollerNet, foi observado que o custo de *overhead* de mensagens do DiCent conseguiu ser reduzido ao se utilizar um número máximo de saltos de 3. Embora a fração de entrega ao utilizar o protocolo de roteamento ProPHET tenho sido bem similar em relação ao mecanismo de incentivo RELICS, o DiCent conseguiu um resultado superior ao RELICS ao utilizar o protocolo de roteamento MaxPROP, mantendo uma taxa de *overhead* inferior.

O atraso médio encontrado pelo DiCent no cenário RollerNet foi constante independente do protocolo de roteamento utilizado, dentro do intervalo de tempo de vida dos pacotes analisado. Isto indica que, devido à preferência de caminhos para a entrega onde exista um menor número de saltos do DiCent, para maximizar o ganho de recompensa, o TTL de cada pacote não teve uma influência significativa. Isto se mostrou um ponto fraco do DiCent no cenário RollerNet ao utilizar o protocolo de roteamento ProPHET, onde o mecanismo de incentivo RELICS obteve um atraso menor.

Foi também mostrado neste trabalho que o DiCent é capaz de utilizar uma abordagem de incentivo por créditos de maneira distribuída, sem a necessidade de um banco virtual para gerenciar as transações realizadas entre os nós. No lugar deste banco virtual foi proposto um registro contendo todas as transações realizadas entre os nós, onde é possível descobrir quantas moedas cada nó possui antes de enviar uma mensagem.

Através do modelo matemático apresentado no Capítulo 3 para o pagamento de créditos baseado nas fórmulas utilizadas por [Chen and Chan 2010] foi possível demonstrar que ao utilizar o mecanismo proposto nenhum nó retransmissor poderia se beneficiar ao utilizar um ataque de *Edge Insertion*. No entanto, evitar que o nó destino se beneficie por este ataque sem que haja uma perda de créditos ainda é um desafio.

Como um trabalho futuro, pretendemos analisar a criação e perda de moedas em uma rede em que os nós possam entrar e sair a qualquer momento. Este cenário também seria um grande desafio quanto à segurança e autenticação dos nós, já que isso também teria de ser feito pelos nós da rede, de forma descentralizada. Também pretendemos realizar simulações envolvendo outros traces de mobilidade, para avaliar os efeitos de um diferente tipo de mobilidade entre os nós.

Referências bibliográficas

VIEIRA, A., 2012 VDTN-TD: *Protocolo de roteamento vanet/dtn baseado em tendência de entrega*. Dissertação de Mestrado. Universidade Estadual do Ceará, Fortaleza, Ceará, Brasil.

Brun, O., El-Azouzi, R., Prabhu, B., and Seregina, T. (2014). Modeling rewards and incentive mechanisms for delay tolerant networks. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*, pages 233–240. IEEE.

Borah, J., Devi, D., & Singh, Y. J. (2010). Analysis and evaluation of probabilistic routing protocol for intermittently connected network. *Journal of Web & Semantic Technology (IJWesT)*, *1*(1).

Burgess, J., Gallagher, B., Jensen, D., & Levine, B. N. (2006, April). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In INFOCOM (Vol. 6, pp. 1-11).

Cao, Y. and Sun, Z. (2013). Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys & Tutorials*, IEEE, 15(2):654–677.

Chen, B. B. and Chan, M. C. (2010). Mobicent: a credit-based incentive system for disruption tolerant network. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

Chen, H., & Chen, G. (2007). A resource-based reputation rating mechanism for peer-to-peer networks. In *Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on* (pp. 535-541). IEEE.

de Melo, M. C. L. (2011). *IMPLEMENTAÇÃO E AVALIAÇÃO DE UM MODELO PARA TROCA DE MENSAGENS EM DTNs UTILIZANDO TRÁFEGO AÉREO* (Doctoral dissertation, Universidade Federal do Rio de Janeiro

de Oliveira, C. T., Moreira, M. D., Rubinstein, M. G., Costa, L. H. M., and Duarte, O. C. M. (2007). Redes tolerantes a atrasos e desconexões. *SBRC Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.*

de Oliveira, C. T. (2008). Uma Proposta de Roteamento Probabilístico para Redes Tolerantes a Atrasos e Desconexões (Doctoral dissertation, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO).

Douceur, J. R. (2002). pages 251–260. Springer.

Durst, R. (2002). A infrastructure security model for delay tolerant networks. Fall, K. (2003). A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pages 27–34. ACM.

Fall, K. (2003). A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM.

Fernandes, N. C., Moreira, M. D., Velloso, P. B., Costa, L. H. M. K., & Duarte, O. C. (2006). Ataques e mecanismos de segurança em redes ad hoc. Minicursos do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'2006), 49-102.Haris, A. (2010). A DTN Study: Analysis of Implementations

and Tools. PhD thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark.

Keränen, A., Ott, J., and Kärkkäinen, T. (2009). The one simulator for dtn protocol evaluation. In *Proceedings of the 2nd international conference on simulation tools and* techniques, page 55. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Li, F., Wu, J., and Srinivasan, A. (2009). Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets. In *INFOCOM 2009, IEEE*, pages 2428–2436. IEEE.

Lindgren, A., Doria, A., and Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3):19–20

Ly, M. K.-M. (2014). Coining bitcoin's "legal-bits": Examining the regulatory framework for bitcoin and virtual currencies. *Harv. J. Law & Tec*, 27:587–587.

Martins, André Francisco (2014) Uma aplicação proposta para troca de arquivos em redes oportunistas sem infraestrutura. Dissertação (Mestrado em Informática) - Universidade Federal do Estado do Rio de Janeiro

Mota, V., Macedo, D., Ghamri-Doudane, Y., Nogueira, J. (2015). MINEIRO: Um Mecanismo de Incentivo para Aplicações em Redes Oportunísticas. *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.*Nadeem, A., & Howarth, M. P. (2013). A survey of MANET intrusion detection & prevention approaches for network layer attacks.*Communications Surveys & Tutorials, IEEE*, *15*(4), 2027-2045.

Ning, T., Yang, Z., Wu, H., and Han, Z. (2013). Self-interest-driven incentives for ad dis- semination in autonomous mobile social networks. In INFOCOM, 2013 Proceedings IEEE, pages 2310–2318. IEEE.

Pirretti, M., Zhu, S., Vijaykrishnan, N., McDaniel, P., Kandemir, M., & Brooks, R. (2006). The sleep deprivation attack in sensor networks: Analysis and methods of defense. *International Journal of Distributed Sensor Networks*, *2*(3), 267-287.

Rashmi, A. S. (2014). Detection and Prevention of Black-Hole Attack in MANETS. *International Journal of Computer Science Trends and Technology (IJCST)–Volume*, *2*, 204-209.

Resnick, P. et al. (2001). The social cost of cheap pseudonyms. Journal of Economics & Management Strategy, 10(2):173–199.

Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A. (2006). CRAWDAD data set Cambridge/haggle (v. 2006-01-31). Downloaded from http://crawdad.org/cambridge/haggle/.

Sermpezis, P. and Spyropoulos, T. (2014). Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks. *Computer Communications*, 48:71–83.

Shevade, U., Song, H. H., Qiu, L., and Zhang, Y. (2008). Incentive-aware routing in dtns. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 238–247. IEEE.

Spyropoulos, T., Psounis, K., and Raghavendra, C. S. (2008). Efficient routing in intermittently connected mobile networks: the multiple-copy case. Networking, IEEE/ACM Transactions on, 16(1):77–90.

Sugiyama, K., Kubo, T., Tagami, A., and Parekh, A. (2013). Incentive mechanism for dtn-based message delivery services. In *Global Communications Conference (GLOBECOM),* 2013 IEEE, pages 3108–3113. IEEE.

Trifunovic, S. and Hossmann, A. (2014). Stalk me if you can: the anatomy of Sybil attacks in opportunistic networks. In *Proceedings of the 9th ACM MobiCom workshop on Challenged networks*, pages 37–42. ACM.

Uddin, M. Y. S., Godfrey, B., and Abdelzaher, T. (2010). Relics: In-network realization of incentives to combat selfishness in dtns. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 203–212. IEEE.

Vahdat, A., Becker, D., et al. (2000). Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University.

Villares, T. L., Campos, C. A. V., & Viana, A. C. (2013). A Influência de Nós Especiais na Entrega de Mensagens nas Redes Tolerantes a Atrasos e Interrupções. *WRA*, *13*, 44-57.

Xiao, L., Greenstein, L. J., Mandayam, N. B., & Trappe, W. (2009). Channel-based detection of Sybil attacks in wireless networks. *Information Forensics and Security, IEEE Transactions on*, *4*(3), 492-503.

Wang, Y., Chuah, M. C., & Chen, Y. (2014). Incentive based data sharing in delay tolerant mobile networks. *Wireless Communications, IEEE Transactions on*, *13*(1), 370-381.

Warthman, F. et al. (2003). Delay-tolerant networks (dtns): A tutorial.

Wei, K., Liang, X., and Xu, K. (2014). A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues. *Communications Surveys & Tutorials*, IEEE, 16(1):556–578.

Wei, L., Cao, Z., and Zhu, H. (2011). Mobigame: A user-centric reputation based incentive protocol for delay/disruption tolerant networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5. IEEE.

Yunchuan, G., Lihua, Y., Licai, L., and Binxing, F. (2014). Utility-based cooperative decision in cooperative authentication. In INFOCOM, 2014 Proceedings IEEE, pages 1006–1014. IEEE.

Zhu, Y., Xu, B., Shi, X., and Wang, Y. (2013). A survey of social-*based routing in delay tolerant networks: positive and negative social effects. Communications Surveys & Tutorials, IEEE*, 15(1):387–401.

Zhu, H., Lin, X., Lu, R., Fan, Y., & Shen, X. (2009). Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks.*Vehicular Technology, IEEE Transactions on*, *58*(8), 4628-4639.