CLASSIFICAÇÃO DE FALHAS DE REDES MÓVEIS EM AMBIENTE DE NUVEM

Relatório Científico

Número do Processo: 2013/17823-0

Período: Fevereiro de 2015 a Janeiro de 2016

Responsável: Profa. Dra. Regina Lúcia de Oliveira Moraes (regina@ft.unicamp.br)

Pesquisador Adjunto: Prof. Dr. Varese Salvador Timóteo (varese@ft.unicamp.br)

Instituição Sede do Projeto: Faculdade de Tecnologia - FT/UNICAMP

UNICAMP / Limeira - SP

Fevereiro 2016

Resumo

Este relatório tem como objetivo apresentar as atividades realizadas no âmbito do Projeto com processo n. 2013/17823-0 no segundo período, que corresponde a doze meses, com vigência de 01 de Fevereiro de 2015 a 31 de Janeiro de 2016.

No período anterior foi feita a revisão bibliográfica relacionada a Sistemas Distribuídos, com o objetivo de identificar os tipos de falhas que foram consideradas características desse ambiente. Esse conjunto de falhas foi utilizado como ponto de partida para a definição de um *faultload* do ambiente computacional que é foco dessa pesquisa, ou seja, redes móveis em nuvem (*mobile cloud computing*).

Também foram identificados trabalhos que abordaram o tema redes móveis em nuvem, particularmente aqueles relacionados à garantia de algum atributo de dependabilidade (*dependability* em inglês), isto é, disponibilidade, confiabilidade ou assuntos relacionados, tais como tolerância a falhas ou segurança. Essa revisão bibliográfica teve continuidade nesse período com o objetivo de identificar novos trabalhos que foram publicados durante o ano de 2015.

O desenvolvimento da pesquisa, neste segundo período, teve como base o protocolo epidêmico, o trabalho com simuladores e comunicação móvel com o protocolo 4G. Os tipos de redes utilizadas para os experimentos foram: sem fio 802.11 no modo *ad-hoc* e redes móvel 3G/4G. Para redes sem fio ad-hoc foi utilizada a topologia full-connect (onde todos os nodos se comunicam, ou seja, todos são vizinhos). Também duas aplicações foram utilizadas para os experimentos em redes ad-hoc, uma sob um protocolo epidêmico, NEEM e outra operando sob o TCP com arquitetura cliente-servidor. Para obter o comportamento padrão foram feitas execuções com trocas de mensagens entre os nós dos participantes da rede (*workload*), sem que nenhuma interferência artificial fosse utilizada, obtendo dessa forma, o comportamento normal do ambiente de teste que foi utilizado como oráculo (*Golden Run*). Outras execuções foram feitas utilizando as falhas características de ambientes distribuídos (conforme a literatura existente: falhas de omissão, falhas de atraso, falhas bizantinas, i.e., corrupção de mensagens), que foram injetadas durante a execução do *workload*.

Para o ambiente móvel foi desenvolvido um simulador, que atua especialmente nos testes com redes puramente móveis 3G / 4G. Basicamente, um *host* transmissor coordena a troca de mensagens entre os nós que compõem a rede no simulador. Esse ambiente foi instalado e configurado para dar suporte aos experimentos. Utilizaram-se mensagens ICMP por meio do utilitário "*ping*" e de acordo com cada cenário de teste em específico (injeção de falhas de perda de pacotes ou de atrasos) e considerando o acesso móvel 3G/4G, o acesso banda-larga cabeado ou a rede local. A injeção de falhas de atrasos se dá com apoio da ferramenta Netem[1] do Linux.

1. Introdução

Este relatório tem como objetivo apresentar as atividades realizadas no âmbito do Projeto com processo n. 2013/17823-0 no segundo ano, com vigência de 01 de Fevereiro de 2015 a 31 de Janeiro de 2016. O projeto tem foco no estudo de falhas que ocorrem no ambiente móvel que fazem comunicação com a nuvem (*cloud*).

Dispositivos móveis, tais como celulares e *tablets*, fazem parte do dia-a-dia dos cidadãos e das organizações. A limitação de recursos desses dispositivos faz do ambiente de nuvem uma opção natural. Dessa

[1] Netem em http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

forma, a sociedade está chegando a um novo patamar de mobilidade na utilização de sistemas, tirando vantagem tanto da mobilidade quanto da disponibilidade de aplicações e recursos na nuvem (*mobile-cloud computing*).

O modelo de computação em nuvem (*cloud computing*, em inglês) utilizando redes móveis *ad hoc*, ou seja, que não têm a necessidade de criação de uma infraestrutura própria, é uma opção bastante utilizada atualmente. Como todo ambiente computacional, esta nova proposta de ambiente está sujeita a falhas[2]. Entender o comportamento de aplicações e comunicação em presença de falhas é essencial para aumentar a dependabilidade (*dependability* em inglês) dos componentes envolvidos. Para isso, caracterizar as falhas que realmente ocorrem nesse ambiente é essencial para poder emula-las, observar o comportamento do sistema na presença destas e criar mecanismos que minimizem o impacto de possíveis falhas que permanecem no software em fase operacional.

Falhas de software (falhas de programação) são, reconhecidamente, a causa principal de defeitos nos sistemas (Gray 1990, Lee e Iyer 1995) e isso envolve os protocolos de comunicação que são programados para gerenciar a troca de mensagens. A avaliação de dependabilidade (*dependability* em inglês) é naturalmente baseada em técnicas que incluem observações no ambiente operacional (Gray 1990), injeção de falhas (Arlat et al 1993) e testes de robustez (Koopman e De Vale 1999).

A ativação das falhas, acontecimentos raros e difíceis de serem identificados (se assim não fosse, as falhas seriam descobertas e eliminadas na fase de teste), pode levar muito tempo para ocorrer naturalmente. Assim, é necessário um mecanismo que acelere a ativação das falhas no sistema. Nesse sentido, a injeção de falhas é uma técnica útil para se obter esse fator de aceleração na ativação das falhas. Através dessa técnica torna-se possível a obtenção de leituras significativas em intervalo de tempo viável (Chillarege e Bown 1989). Assim, experimentos para a avaliação da dependabilidade podem usar a técnica, da seguinte forma: (i) o sistema em observação é acionado através de uma carga (*workload*), ao mesmo tempo em que um conjunto de falhas (*faultload*) é injetado. O comportamento do sistema é, então, observado de forma a avaliar suas propriedades de dependabilidade (Kanoun e Spainhower 2008).

O objetivo dessa proposta foi definir um *faultload* baseado em falhas de comunicação para que os testes baseados em falhas no ambiente de mobilidade na nuvem possam utilizá-lo para testes (padronizados ou não) de avaliação de dependabilidade. Ou seja, como proporcionar uma resposta à questão "como definir um *faultload* baseado em falhas de comunicação para que os testes baseados em falhas no ambiente de mobilidade na nuvem possam utiliza-lo para testes de avaliação de dependabilidade?". A resposta a esta questão passa, obrigatoriamente, pela resolução das questões acerca da representatividade, classificação de falhas, técnicas de injeção. Analisar e determinar quais ameaças devem ser consideradas para avaliar cada ambiente computacional (nesse caso, rede móvel *ad hoc* que utilizam aplicações que fazem uso de recursos da nuvem) é uma tarefa essencial para a definição de *faultloads* representativos (Kanoun e Spainhower 2008).

O desenvolvimento do trabalho teve sucesso em definir e configurar um ambiente para a pesquisa experimental. Foram feitos experimentos com protocolos epidêmicos e TCP, com a emulação de falhas de omissão, atraso de mensagens, considerando redes *ad hoc* (sem mobilidade) e redes sem fio (*MANET* e 4G). A ferramenta FIRMAMENT (Drebes 2005) foi adaptada para apoiar os experimentos por injeção de falhas e mostrou-se adequada para todos os testes com redes *ad hoc*. Os experimentos para as redes 3G e 4G não puderam ser feitos da mesma forma devido a especificidades que não permitiram as adaptações com as mesmas tecnologias. As redes 4G foram

[2] A terminologia em português utilizada neste projeto segue o proposto por Leite e Loques (1987).

observadas com o uso de simuladores, uma vez que o ambiente real não se mostrou viável. O documento está organizado da seguinte forma: a seção 2 apresenta o Plano de Trabalho de acordo com o que foi proposto no projeto, acrescido dos comentários sobre o desenvolvimento de cada uma das etapas. A seção 3 descreve e avalia o apoio institucional recebido e a seção 4 comenta sobre o uso da reserva técnica.

2. Realizações no Período

Esta seção traz o plano de trabalho adaptado do inicial, que constou no relatório parcial enviado no final do primeiro ano do projeto e é apresentado na Tabela 1.

Considerando o cronograma apresentado no relatório do final do primeiro ano, uma grande parte dos objetivos foi cumprida. Foi feita a atualização bibliográfica, de forma que possa ser repetida durante as etapas necessárias. Aspectos de Caracterização de Falhas na rede 802.11 foram estudados em campo, em laboratório e em simulador considerando-se a rede sem fio sem e com mobilidade. O estudo do *faultload* considerando falhas de omissão, atrasos e Bizantinas foi evoluído. Nesse período foi gerado um modelo estatístico preditivo para falhas na rede 802.11. Além disso, foi implementado no simulador NS-3 uma aplicação *Multicast-Gossip* que permitiu escalar os experimentos da rede 802.11 no modo *ad hoc* e avaliar o protocolo epidêmico em face dos diversos parâmetros que podem trazer impacto aos resultados.

Com relação a protocolos de redes móveis não foi possível efetuar experimentos com os dispositivos reais, tanto pela dificuldade de se implementar uma aplicação que fosse adequada para os testes quanto pela instabilidade do serviço provido pelas operadoras. Isso fez com que os testes demandassem tempo não razoável e falhassem diversas vezes antes de se conseguir o término de um único experimento. Além disso, a escalabilidade em termos de número de dispositivos seria novamente um problema, motivo pelo qual foi feita a opção de execução dos experimentos em simulador de redes móveis (4G). Esses experimentos permitiram a emulação de perdas e atrasos de pacotes na troca de mensagens entre dispositivos na rede 4G.

Tabela 1: Cronograma relativo ao segundo ano do projeto

Descrição da atividade	Sem. 3 (Fev a Jul 2015)	Sem. 4 (Ago 2015 a Jan 2016)
1- Atualização Bibliográfica		
2-Aspectos de Caracterização Falhas (802.110)		
3- Comparação de resultados em novo ambiente (3G)		
4- Comparação de resultados em ambiente final (3G, 4G)		
5- Definição do *faultload*		
6- Projeto do Protótipo 3G / 4G		
7- Preparo de Ambiente Experimental 3G / 4G		
8- Experimentos		
9- Preparação de Publicações		

No primeiro ano houve uma publicação feita em revista que usou conhecimentos mais teóricos. Nesse segundo ano, foram publicados dois trabalhos completos e dois resumos estendidos em conferências. Há uma extensão do trabalho teórico a ser submetido para um número especial (*special issue)* até março de 2016. Além desses, outros trabalhos devem ser submetidos ainda no primeiro semestre de 2016: um trabalho que estende um dos trabalhos publicados em conferência, um trabalho sobre o desenvolvimento de um simulador especificamente desenvolvido para o projeto e um trabalho relacionado às redes móveis 4G.

2.1 Comentários sobre as etapas do cronograma

Esta seção apresenta as etapas do cronograma e o status de desenvolvimento no momento do fechamento desse relatório.

2.1.1. Atualização bibliográfica.

A atividade de atualização bibliográfica se deu de forma continuada.

Status: Cumprida.

2.1.2. Preparação de publicações.

Essa atividade tem a intenção de disseminar o conhecimento criado pelo projeto por meio de publicações em eventos da área, assim como em periódicos. Relatórios do projeto do primeiro ano encontram-se no website do grupo de pesquisa e uma versão do presente relatório estará disponível no mesmo ambiente após aprovado.

Uma das publicações foi apresentada, pela coordenadora do projeto, no "*International Conference and Workshop on Computing and Communication on Computer and Information Technology - IEMCON 2015*", ocorrido em Vancouver, Canadá, em Outubro de 2015. Esse trabalho, intitulado "*Analyzing the Behavior of Communication Faults in ad-hoc Networks*", divulgou os resultados dos experimentos em laboratório com as redes ad-hoc (protocolo 802.11) e utilizou esses resultados como base para propor um modelo estatístico de previsão de falhas futuras (modelo de regressão beta inflacionado em um). Esse modelo permitiu escalar os resultados para uma rede com um maior numero de nós (acima de 8 nós), uma vez que a escalabilidade foi bastante criticada nas primeiras submissões que fizemos sobre o tema, motivo pelo qual decidimos faze-la com base em modelo estatístico.

Outro trabalho completo em conferência foi apresentado, pela coordenadora do projeto, no "*The 15th IEEE International Conference on Computer and Information Technology - CIT 2015*", ocorrido em Liverpool, UK, em Outubro de 2015. O trabalho versou sobre um modelo UML que agrega as questões de segurança e privacidade em decorrência de falhas. O trabalho, intitulado "*Towards a UML Profile for Privacy-Aware Applications*", propôs e validou com usuários especialistas um perfil UML que guia o desenvolvimento de aplicações que precisam cuidar para que falhas e ataques não comprometam a qualidade dos serviços fornecidos.

Um terceiro trabalho foi publicado na forma de resumo estendido no "XIV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC2015)", tendo sido apresentado pela primeira autora. O título do trabalho, "Adaptação de métodos de avaliação de acessibilidade e usabilidade com foco no uso de dispositivos móveis por idosos", retrata a preocupação com falhas no uso de dispositivos móveis que podem trazer problemas na comunicação pelo público idoso. Outro resumo estendido será apresentado em Pisa, Itália no "*31st ACM Symposium on Applied Computing - SAC 2016*", tendo sido intitulado "*Usability Heuristics and Accessibility Guidelines: a Comparison of Heuristic Evaluation and WGAG*". Essa publicação trata de avaliação de interfaces quanto ao

atendimento de normas de certificação. A cópia dessas publicações se encontram em anexo no SAGE / FAPESP e os respectivos resultados não serão comentados por questão de espaço no documento.

Outras publicações encontram-se em andamento. A primeira publicação que está sendo finalizada é uma extensão do artigo publicado em periódico em 2015, agregando nessa nova versão, modelos Fuzzy. Esta publicação, intitulada "***n-Steps-Ahead Failure Prediction using the Kalman Filter, the Laplace Trend Test and Fuzzy Control***" será encaminhada para um numero especial (*special issue*) da revista "*Software Reliability Engineering*", da Elsevier (submissão em Março de 2016). Esse estudo com base mais teórica é importante para dirigir a busca por novas falhas em software e seu abstract se encontra em anexo no SAGE / FAPESP.

A segunda publicação está baseada nos experimentos com o protocolo epidêmico NEeM, cujos resultados são parcialmente apresentados nesse relatório. Nesse trabalho, foram observado os experimentos com 5 a 8 nós e foi proposto um novo modelo estatístico que permite escalar os resultados obtidos no laboratório. A escrita está sendo refinada para submissão no próximo mês. O título escolhido foi "***Communication Fault prediction in ad-hoc networks with full-connect topology***". Pretende-se submetê-lo a um congresso.

A terceira publicação, também em andamento, está baseada nos experimentos com o protocolo epidêmico que foram feitos em um simulador, especialmente desenvolvido para o projeto. O simulador permite parametrizar o protocolo NEeM de forma a otimizar suas entregas quando o número de nós da rede é escalado a centenas de nós. Os resultados também constam no presente relatório e a escrita está sendo refinada para submissão a partir de abril de 2016. O título provável será "***Performance Evaluation of Multicast-Gossip Protocol in ad-hoc Networks***". Pretende-se submete-lo a uma revista.

Uma quarta publicação, também em andamento, está baseada nos experimentos com redes puramente móveis (4G). Os experimentos foram desenvolvidos em simulador, utilizando os mesmos cenários utilizados para as redes *ad-hoc*, de forma que será possível obter comparações dos resultados observados. O título provável desse artigo será "***Injeção de falhas em ambientes 3G/4G: Um estudo comparativo envolvendo simulações e emulações de tráfego de rede***" e pretende-se submete-lo a um congresso.

Se tivermos sucesso em todas essas submissões, o projeto terá gerado 3 publicações em revistas, 4 trabalhos completos em congressos internacionais, 1 resumo estendido em congresso nacional e 1 resumo estendido em congresso internacional.

Status: Cumprida.

2.1.3 Aspectos de caracterização de falhas e Definição do *Faultload*.

Nesse período, o estudo de falhas foi feito de várias formas. Primeiramente foi observado o impacto de falhas artificiais em um ambiente experimental composto por uma rede *ad hoc* (*rede wireless*) com até oito nós, resultando em um aumento na escala da rede em laboratório em relação ao ano anterior (quando foram utilizados 5 nós). Os novos conjuntos de experimentos, que também tinham como objetivo trocar mensagens entre os nós da rede, utilizou o protocolo epidêmico NEeM. O estudo permitiu comparar resultados com aqueles obtidos anteriormente, o que também foi validado utilizando-se um modelo estatístico de previsão de falhas. Também foram exercitadas falhas de omissão e atrasos nas redes móveis 3G. Os experimentos foram feitos em simuladores.

Apesar dos diversos experimentos efetuados, não foram revelados novos tipos de falhas e chegou-se a conclusão de que os tipos de falhas característicos de sistemas distribuídos apontados por Cristian (1991), Schneider (1993) e Drobe *et al.* (2008) continuam válidos para as redes *ad-hoc* e puramente móveis (3G). Embora existam

algumas diferenças na maneira como as falhas se manifestam e impactam o ambiente, falhas de *crash*, atrasos e omissão na entrega e recebimento de pacotes e falhas Bizantinas continuam presentes nesses novos ambientes.

Status: Cumprida.

2.1.4. Preparo do ambiente experimental 3G / 4G.

O preparo do ambiente experimental para redes puramente móvel e a realização de experimentos são atividades que consistem em, respectivamente, integrar ferramentas necessárias para o funcionamento do simulador criado e realização de diversos testes para avaliar o impacto das falhas (perda de pacotes e atrasos) no sistema alvo. O impacto deve ser avaliado segundo o comprometimento da efetividade da entrega das mensagens e disponibilidade dos serviços. As limitações e outros conceitos relevantes devem ser observados.

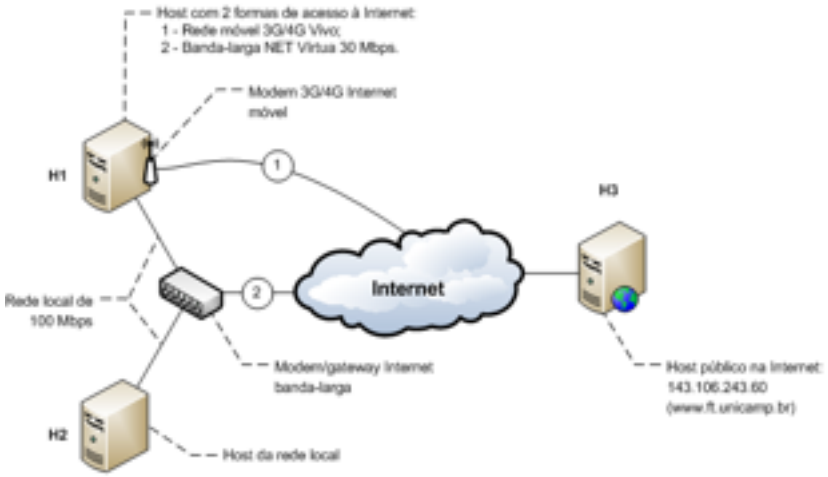

Figura 1: Ambiente Experimental 3G / 4G

Ambiente Experimental 3G/4G. O ambiente experimental utilizado pela rede puramente móvel, foi composto por *laptops*, um modem externo 4G (Electroson 4G LU11), conexões de banda larga e rede local. Embora o hardware tenha suporte para 4G (tanto o modem quanto o chip), não há suporte de tecnologia 4G provida por parte do provedor de serviço. Dessa forma, os resultados registrados usaram a tecnologia 3G. Além desse ambiente 4G foi utilizado um ambiente com conexão de banda larga de 30 Mb e um ambiente conectado em rede local de 100 Mbps (Figura 1).

No ambiente da Figura 1, três *hosts* com acesso à Internet, tidos como *hosts* H1, H2 e H3, foram utilizados para fins de experimentação. Desses, o *host* H1 atua como *host* transmissor do fluxo de tráfego de rede e, por sua vez, os *hosts* H2 e H3 atuam como *hosts* receptores de tal fluxo de tráfego. No âmbito do acesso à rede, H1 é capaz de se conectar a Internet por meio de uma rede móvel 3G/4G fornecida pela Operadora Vivo, e, também, por meio de uma conexão banda-larga (NET Vírtua 30Mbps) fornecida pela Operadora NET. Nesse ambiente, H1 e H2 também fazem parte de uma mesma rede local, estando conectados entre si por meio de um enlace cabeado de 100 Mbps. Por sua vez, H3 é um *host* público na Internet, sendo representado pelo *host* de endereço IPv4 143.106.243.60 (www.ft.unicamp.br).

Em termos de dispositivos de conexão utilizados por H1, seu acesso à Internet móvel se dá por meio do modem Electroson 4G LU11[3] e à Internet banda-larga por meio do modem/*gateway* Sagemcom F@st 3284[4]. Quanto às transmissões utilizando a rede móvel da Operadora Vivo, de acordo com as estatísticas de conexão, a tecnologia empregada em seu processo de comunicação foi a UMTS/HSPA (3G). De modo complementar, as coberturas disponíveis pela Operadora Vivo[5] no local onde os testes foram realizados (Faculdade de Tecnologia da Universidade Estadual de Campinas, Limeira/SP, CEP 13484-332) são 2G e 3G.

Ambiente Experimental ad hoc sob Protocolo Epidêmico. Considerando os experimentos com o protocolo epidêmico continuou-se a utilização do protocolo NEeM (Pereira *et al.*, 2003) para efetuar a disseminação de mensagens na rede. Como geradora do *workload* (nesse caso, geração de mensagens), também foi utilizada a mesma aplicação - NetEpidemic. Ela dissemina mensagens de um nó gerador para os demais nós que se encontram conectados sob a mesma *cloud*. Ela também administra o recebimento de notificações de cada nó, quando esses recebem as mensagens enviadas. A aplicação adaptou o protocolo NEeM como uma biblioteca, para lidar com a disseminação de mensagens. O Castadiva (Hortelano et al. 2007) foi utilizado para emular a mobilidade nesse ambiente. O ambiente experimental é, em sua essência, o mesmo, tendo sido variado apenas o numero de nós que saltou de 5 para 8 nós. Por questão de restrição de espaço não constam nessa versão do relatório os detalhes que já estavam inclusos no relatório do primeiro período.

Status: Cumprida parcialmente, uma vez que não tivemos êxito com os experimentos utilizando os dispositivos móveis reais, embora tenha sido desenvolvido um simulador para emular o mesmo ambiente.

2.1.5. Experimentos

Para todos os experimentos foram utilizados os mesmos cenários. As aplicações específicas de cada ambiente foram executadas por 15 minutos, simultaneamente, por todos os nós da rede e a cada 30 segundos a aplicação enviava uma mensagem de cada um dos nós para os demais nós que compunham o ambiente. Cada experimento foi repetido por 10 vezes e os resultados foram obtidos como a média das 10 execuções.

Experimentos com redes *ad hoc* em laboratório. Os experimentos com redes *ad hoc* se iniciaram no primeiro ano do projeto e foram relatados no relatório do primeiro período. Os experimentos foram feitos tanto sob o protocolo epidêmico NEeM quanto sob o protocolo TCP. No fechamento do relatório tínhamos utilizado uma rede com 5 nós e falhas de perda de pacotes e atrasos. Logo nas primeiras submissões, percebeu-se a importância da escalabilidade dos experimentos, pois uma crítica recorrente era o fato da rede ser composta por um pequeno numero de nós. Optou-se então para, com base nos primeiros resultados, elaborar um modelo matemático de predição para emular ambientes similares àqueles utilizados em laboratório, quando foram utilizados dispositivos reais. Restringiu-se esse modelo ao protocolo epidêmico pois foi o protocolo que melhor traduziu o comportamento que se buscava. Essa fase de desenvolvimento gerou uma publicação em congresso internacional e se encontra em anexo no SAGE /

[3] Modem 4G LU11 em http://produtos.celistics.com/Produtos/4G/LU11/Produto_LU11.htm

[4] Modem Sagemcom F@st 3284 em http://www.sagemcom.com/broadband/gateways/docsis-gateways/fst-3284/

[5] Vivo (Cobertura e Roaming) em http://www.vivo.com.br/portalweb/appmanager/env/web?_nfls=false&_nfpb=true&_pageLabel=vivoVcCobNacCoberturaVivoPage&WT.ac=portal.internet.modem.coberturaeroaming#

FAPESP, motivo pelo qual não será apresentada no relatório (por favor, consulte a publicação intitulada "Analyzing the Behavior of Communication Faults in ad-hoc Networks" - IEMCON'15).

Experimentos com 6 a 8 nós na rede *ad hoc*. Para uma melhor avaliação dos experimentos realizados em laboratório com equipamentos reais e conectados através de uma topologia full-connect em rede sem fio no modo *ad-hoc*, bem como a validação do modelo preditivo, decidiu-se escalar os experimentos para 6, 7 e 8 nós, visando avaliar o comportamento da rede. Para não repetir todos os experimentos, uma vez que o modelo estatístico preditivo comprova o comportamento semelhante da rede, optou-se por selecionar um único cenário para execução dos experimentos. O cenário escolhido considerou 60% dos nós da rede sob falhas. Logo foram executados os experimentos com, respectivamente, 3, 4 e 5 nós sob falha. O cenário 1 foram os experimentos já executados anteriormente com 5 nodos tanto utilizando a aplicação NetEpidemic quanto a NetTCP. Um novo modelo estatístico foi gerado com base nos resultados obtidos em laboratório, similarmente ao que tinha sido feito para a topologia com 5 nós.

Experimentos com base na aplicação no simulador NS-3 para redes *ad hoc*. Com o objetivo de efetuar as simulações de MANET, operando sobre um ambiente *multicast*, com abordagem de *gossip*, foi desenvolvida a aplicação *Multicast-Gossip* no simulador de redes NS-3. Esta aplicação se baseia no protocolo ICMP, no nível de transporte da camada de rede. O algoritmo para *gossip* foi implementado no nível de aplicação, ou seja, todas as regras e parâmetros utilizados estão na camada de aplicação. O protocolo ICMP foi escolhido devido a sua vasta utilização para comunicação e descoberta de nodos ativos na rede. A implementação da aplicação *Multicast-Gossip* no NS-3 foi baseada nos trabalhos de três outras implementações referentes a *multicast* e *gossip (*Alenazi et. al., 2015; Falke et. al., 2015; Pereira et al., 2003). Baseado nestes trabalhos definiu-se os parâmetros a serem utilizados na aplicação implementada para este projeto, que estão descritos na Tabela 2. Estes parâmetros definem a maneira como a comunicação irá ocorrer no ambiente e como interferem diretamente nos resultados.

Tabela 2. Parâmetros *Gossip* para *Multicast*

Número de Nós da Rede	Define a quantidade de nós que fazem parte da rede. Não está sendo contemplada neste trabalho a partição ou acréscimo de nós na rede. Logo, as quantidades de nós são fixas nos experimentos.
Fanout	Identifica a quantidade de vizinhos que um nó irá replicar uma mensagem recebida.
Hop	Identifica a quantidade de vezes ("saltos") que uma mensagem irá ser replicada no ambiente. Cada nó recebe no corpo da mensagem o dado e o numero de *hop* da mesma. Então incrementa 1 no *hop* da mensagem e envia para outros nós. A mensagem não é mais replicada quando o número de *hop* chega a este limite definido.
Mobilidade	Indica o tipo de Mobiliade para o Ambiente. Nesse caso, foi utilizado o modelo RWP, que distribui os nós na área definida e efetua a movimentação dos mesmos randomicamente de acordo com a velocidade também estabelecida. Sem Mobilidade, utiliza o modelo GRID, que distribui os nós uniformemente com distancias fixas um do outro, na área definida para o ambiente, e não possui velocidade.
Alcance de Sinal do Nodo	Indica o alance de sinal, em metros, atingido por um nó, já que o ambiente é sem fio. Este alcance pode variar entre os equipamentos de 30 a 50 metros. Neste trabalho prefiriu-se utilizar a cobertura de 30 metros, conforme indicado pela maior parte dos fabricantes.
Velocidade do Nodo	Indica a velocidade em m/s que o nó irá possuir quando o ambiente for com mobilidade

Quantidade de Vizinhos	Indica a quantidade máxima de vizinhos que um nó poderá ter no período que estiver ativo na rede.
Intervalo de *Broadcast*	Indica o intervalo, em segundos, que será enviada uma mensagem de *broadcast* para sondagem e reconhecimento da rede, para que cada nó encontre seus vizinhos.
Intervalo de Data	Indica o intervalo, em segundos, que será enviada uma mensagem contendo dados para os vizinhos de um nó, o *gossip* efetivamente.
Área total do Experimento	Indica a área, em m², em que os nós estarão distribuídos. Esta área é definida através dos valores para os eixos X e Y de um plano e serão proporcionais ao número de nós.
Estratégia de *Gossip*	Indica a estratégia utilizada para a replicação de mensagem, visto que existem várias maneiras de propagar uma mensagem na comunicação *multicast com gossip*.

O modelo da implementação da aplicação *Multicast-Gossip*, pode ser visualizado na Figura 2.

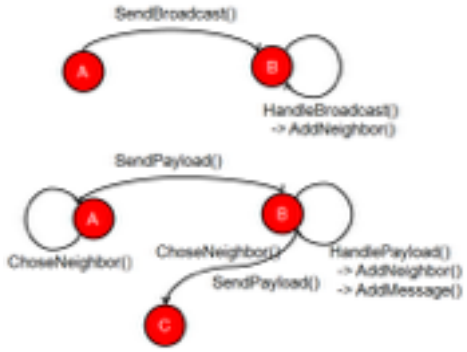

Figura 2. Modelo da Aplicação *Multicast-Gossip*-NS-3.

Na aplicação foram implementados sete métodos principais que efetuam e auxiliam a comunicação *multicast*. O método ***SendBroadcast()*** é responsável por enviar, a cada intervalo de tempo definido para broadcast, uma solicitação para todos os nós da rede. Este método utiliza o protocolo ICMP para gerar um ACK (acknowledgment) entre todos os nós e descobrir quais estão próximos e são vizinhos. O método ***HandleBroadcast()*** é responsável por receber o ACK enviado por um nó e adicionar este nó à lista de vizinhos. Para isso chama o método ***AddNeighbor(),*** que possui uma lista de vizinhos com o IPAddress e a hora que cada nó foi adicionado como vizinho. Na atualização desta lista, é verificado se o IPAddress já existe e, se existir, é atualizado com a hora atual. Também nesta atualização da lista de vizinhos, é verificado se a lista está cheia, ou seja, com o numero máximo de vizinhos definido. Caso esteja cheia, o nó que possuir um tempo mais antigo é eliminado da lista de vizinhos.

O método ***SendPayload()*** é responsável por enviar as mensagens contendo dados, para que seja disseminada em *gossip* para os demais nós. É definida uma quantidade limite de mensagens a serem enviadas pelo Source. Este método invoca o método ***ChooseNeighbor()***, que efetua uma seleção aleatória da lista de vizinhos para os quais será enviada a mensagem. A quantidade de vizinhos que serão selecionados é definida através do parâmetro de *Fanout,* que atribui a quantidade máxima de vizinhos para replicar a mensagem. Também é adicionada na mensagem de dado o número de *hop*, iniciado com 1 pelo nó Source. O método ***HandlePayload()*** é ativado assim que uma mensagem de dados é recebida pelo nó. Este recebe a mensagem que contêm os dados, identificando o número do *Hop* atual da mensagem. Esta mensagem é adicionada à lista de mensagem recebidas do nó e reencaminhada para os vizinhos do nó, invocando o método ***SendPayload()*** novamente, porém incrementando um

ao número do *hop*. Caso o *hop* já tenha chegado ao limite, a mensagem não é mais encaminhada pelo nó. Para minimizar a quantidade de mensagens redundantes foi implementada uma estratégia de *gossip* que não reencaminha a mesma mensagem, mesmo que o número de *hop* seja inferior ao máximo.

Para avaliação do comportamento da rede no simulador definiu-se as métricas de avaliação que foram utilizadas na apuração dos resultados. Estas métricas estão vinculadas à análise de confiabilidade (*Reliability*) e desempenho (*Performance*). São elas:

(i) **Taxa Entrega (*Delivery Rate*)**: avalia a quantidade de mensagens não recebidas pelos demais nós da rede (*packet loss*), após envio de uma mensagem pelo nó *Source*.

(ii)**Média de Mensagens Redundantes:** Avalia a quantidade de mensagens redundantes recebidas por cada nó, uma vez que foi utilizada a estratégia de *gossip* para mensagens redundantes.

(iii) **Tempo Máximo de Recebimento de Mensagens (Latência):** Avalia tempo entre o envio da mensagem pelo *Source* até o recebimento do nó com maior tempo de recebimento.

(iv) **Taxa de Nós que recebem todas as Mensagens:** Avalia o quanto uma mensagem foi disseminada na rede através da quantidade de nós atingidos (*spread network*).

Os cenários de execução de experimentos estão descritos na Tabela 3. Cada cenário possui um identificador, a quantidade de nós da rede, *fanout*, *hop*, tipo de mobilidade e velocidade. Para a quantidade de nós da rede definiu-se uma sequencia de progressão geométrica iniciada em 8 com quociente 2, utilizando-se de um mínimo de 8 e um máximo de 128 nós.

A mobilidade RWP indica o tipo de mobilidade implementado no simulador através do algoritmo de *RandomWayPoint*. A velocidade definida inicialmente foi a de 2,22 m/s para os experimentos com mobilidade, que representa uma pessoa andando rapidamente em um espaço geográfico.

O *fanout* indica a quantidade máxima de vizinhos que serão selecionados para enviar a mensagem. Este parâmetro foi alterado em cada conjunto de experimento para avaliar sua influencia na entrega de mensagens, uma vez que é por ele que ocorre a disseminação da mensagem. O *hop* indica a quantidade de saltos que a mensagem poderá percorrer. A quantidade máxima de vizinhos que um nó pode ter foi definida como a quantidade total de nós da rede menos um. Para a área em que os nós estarão distribuídos foi selecionada inicialmente uma área de 100 m² (10 m x 10 m), para 8 nós, sendo proporcionalmente aumentada conforme o aumento de número de nós (12.800 m² para 128 nós).

Tabela 3. Parâmetros dos Cenários

ID	Qtd. Nós	*Fanout*	*Hop*	Mobilidade	Velocidade	Área (m²)
1 a 8	8 a 1024	5	5	RWP	2,220	100-12800
9 a 16	8 a 1024	6	5	RWP	2,220	100-12800
17 a 24	8 a 1024	7	5	RWP	2,220	100-12800
25 a 32	8 a 1024	8	5	RWP	2,220	100-12800
33 a 40	8 a 1024	9	5	RWP	2,220	100-12800
41 a 48	8 a 1024	10	5	RWP	2,220	100-12800

Em todos os nós foi definido um alcance de sinal de 30 metros, conforme definição de fabricantes dos equipamentos com rede sem fio de uso comercial. Tanto para enviar *broadcast*, quanto para enviar dados, foi definido um intervalo de 10 segundos. Porém o *broadcast* de mensagens se inicia no tempo 5 segundos e os envio de dados é iniciado no tempo 10 segundos, para que houvesse tempo hábil de encontrar vizinhos antes de enviar as mensagens de dados. Os resultados obtidos com o simulador NS-3 também são apresentados na seção 2.2.

Experimentos com base na aplicação no simulador para redes 3G / 4G. Um simulador, desenvolvido especialmente para os testes com redes puramente móveis 3G / 4G, fica em execução no host transmissor H1 (ver Figura 1), fazendo com que mensagens sejam trocadas entre os nós que compõem a rede no simulador. Esse ambiente foi instalado e configurado para dar suporte aos experimentos. Não surtiu efeito os experimentos utilizando dispositivos móveis reais, uma vez que não havia recursos de software que pudessem emular falhas e monitorar os experimentos e o desenvolvimento desses produtos se mostraram muito complexos para o tempo do projeto. Os dispositivos móveis foram utilizados para testes, mas estes não evoluíram adequadamente, motivo pelo qual foram substituídos pelo simulador.

No ambiente do simulador, fluxos de tráfego são transmitidos do *host* transmissor H1 para os *hosts* receptores H2 e H3 (ver Figura1), utilizando mensagens ICMP por meio do utilitário “ping” e de acordo com cada cenário de teste em específico (injeção de falhas ou injeção de atrasos e considerando o acesso móvel 3G/4G, o acesso banda-larga cabeado ou a rede local). Por sua vez, a injeção de falhas se dá por meio da ferramenta Netem[6] do Linux, configurada no *host* transmissor H1. A injeção de falhas teve como base pacotes ICMP enviados do *host* transmissor H1 para os *hosts* receptores H2 e H3. Os percentuais de falhas de perdas de pacotes se iniciaram em 50% de perda, com acréscimo de 5% a cada novo cenário, até o limite final de 95% de falhas, que foram injetadas em cada fluxo de tráfego enviado do host transmissor para o receptor. Tais definições resultaram em 10 cenários distintos quanto à injeção de falhas de perda de pacotes na rede. Para cada cenário, cinco testes distintos (de T1 à T5) foram realizados com duração total de 15 minutos, enviando uma mensagem ICMP do *host* transmissor para o *host* receptor a cada 30 segundos, resultando num total de 30 mensagens por teste. Tais cenários/testes foram realizados de modo idêntico em termos do acesso móvel 3G/4G, de banda-larga cabeada e de rede local. Em termos de configuração, apenas as variáveis relacionadas a interface de rede do *host* transmissor (wan0, wlan0 ou eth0, por exemplo) e ao percentual de falhas que será injetada em seus fluxos de transmissão (nesse caso, de 50% à 95% de falhas) são modificadas .

Assim como nos cenários de injeção de falhas de perda de pacotes, a ferramenta Netem também foi utilizada para a injeção de falhas de atrasos, i.e., *delays* na transmissão. Para tal, a injeção de atrasos teve como base pacotes ICMP enviados do *host* transmissor H1 para os *hosts* receptores H2 e H3. Os percentuais de atrasos iniciaram em 10ms com acréscimo de 10ms a cada novo cenário, até o limite final de 100ms de atraso. Tais definições resultaram em 10 cenários distintos quanto à injeção de atrasos na transmissão.

Os experimentos acerca da injeção de falhas de atrasos foram realizados de modo similar ao de injeção de falhas de perda de pacotes. Para cada cenário, 5 testes distintos (de T1 à T5) foram realizados. Cada teste teve duração total de 15 minutos, com 1 mensagem ICMP enviada do *host* transmissor para o *host* receptor a cada 30

[6] Netem em http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

segundos, resultando num total de 30 mensagens por teste. Tais cenários/testes foram realizados de modo idêntico em termos do acesso móvel 3G/4G, de banda-larga cabeada e de rede local.

Em termos de configuração, apenas as variáveis relacionadas a interface de rede do *host* transmissor (wan0, wlan0 ou eth0, por exemplo) e ao tempo (em ms) que será injetada em termos de atrasos em seus fluxos de transmissão (nesse caso, de 10ms à 100 ms de atrasos) são modificadas. Os resultados obtidos com esse simulador também são apresentados na seção 2.2.

2.2. Resultados dos Experimentos.

Esta seção apresenta os resultados dos experimentos efetuados nessa etapa que compreende o uso do protocolo epidêmico, uso de simuladores para escalar redes sob protocolo epidêmico e o uso de 3G. Os primeiros resultados surgiram do modelo estatístico que foi gerado com base nos experimentos em laboratório com uma topologia de 5 nós. Esses resultados se encontram em um trabalho publicado no IEMCON'15 e que foi colocado em anexo no SAGE / FAPESP. Pelos resultados observados, em tese, o modelo pode ser aplicado para qualquer topologia de rede em que os equipamentos sejam similares, a rede seja *ad-hoc* com topologia *full-connect* e o número de vizinhos seja igual a variáveis *fanout* do protocolo epidêmico. As próximas subseções apresentam os experimentos que escalaram a rede para até 8 nós, experimentos que se apoiaram no simulador NS-3 e experimentos com redes puramente móveis (3G).

2.2.1. Experimentos com 6 a 8 nós na rede ad hoc (escalabilidade física da rede)

Os resultados obtidos nos experimentos com até 8 nós, visando escalar a topologia em número de nós em laboratório, são apresentados nessa seção. O cenário escolhido considerou 60% dos nós da rede sob falhas. Utilizando-se a aplicação *NetEpidemic* foram executados os experimentos nos cenários com 6, 7 e 8 nós. O cenário com 5 nós já havia sido executado no período anterior. A Figura 3 apresenta os resultados com a quantidade de mensagens recebidas sobre o percentual de falhas injetadas no nó 1, sem a presença de injeção de falhas e a Figura 4, para o nó 2, com a presença de falhas.

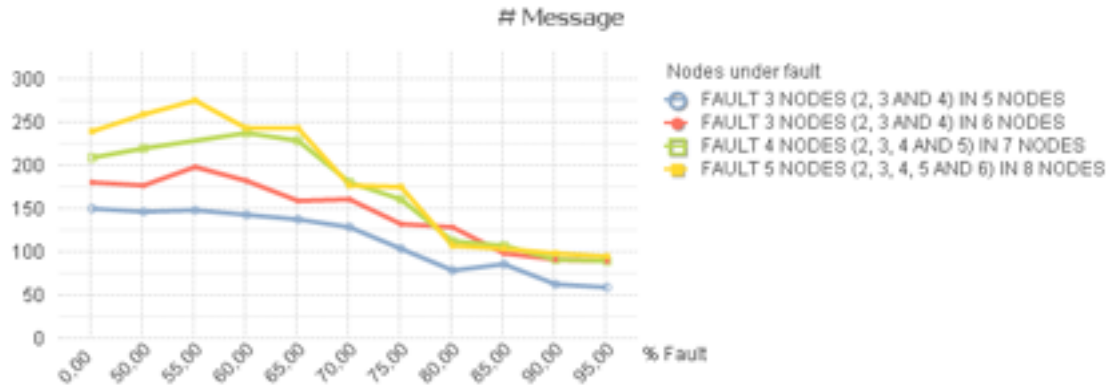

Figura 3. Experimentos com 60% de nós sob falhas, perspectiva do nó 1

Figura 4. Experimentos com 60% de nós sob falhas, perspectiva do nó 2

Observa-se que para este ambiente com topologia full-connect (onde todos os nós se conhecem) a partir de 60% de injeção de falhas, já provoca uma omissão na entrega das mensagens e a queda é gradual, similarmente ao que foi observado com a topologia com 5 nós. Existe uma provável proporcionalidade e o modelo de previsão utilizado revelou isso nas simulações executadas (a observação valida o modelo obtido).

Algo interessante aconteceu com experimentos com 7 e 8 nós, que não haviam sido observados antes em nenhum experimento. As mensagens redundantes foram registradas (ou seja, o mesmo ID de mensagem foi recebido mais de uma vez). Isso realmente ocorre internamente no protocolo, pois como o protocolo epidêmico NEEM se baseia em *Multicast-Gossip*, os vizinhos mandam a mesma mensagem mais de uma vez e os parâmetros de *fanout* e *hop* utilizados proporcionam a disseminação da mensagem através de vários caminhos, gerando redundância e uma possível sobrecarga na rede. Mesmo que o protocolo epidêmico NEEM tenha como meta garantir um bom gerenciamento de *buffer,* através de seus mecanismos internos, que não foram avaliados neste projeto, ainda existe uma redundância de mensagens.

2.2.3. Experimentos com simulador NS_3.

Com os dados coletados através do simulador NS-3 durante a execução sob controle da aplicação *Multicast-Gossip*, foi possível avaliar as métricas definidas para o protocolo epidêmico. A Tabela 4 apresenta o resumo dos resultados para o cenário executado com 8 nós. Para as demais configurações (de 16 a 128 nós) os resultados foram similares, agravando-se conforme houve um aumento no número de nós da rede. São apresentados as métricas de latência, taxa de entrega (*Delivery Rate*), percentual de nós atingidos na rede (*spread network*) e quantidade média de mensagens duplicadas recebidas por nó.

Tabela 4. Fragmento dos Resultados do Simulador (para 8 nós)

Nós da Rede	*Fanout*	Latência (s)	% Menssagens Perdidas	% de nós Atingidos	Média de Mensagens duplicadas recebidas por nó
8	5	0,012630	0,96	98,90	4
	6	0,012500	0,21	99,76	5
	7	0,014567	0,00	100,00	5
	8	0,014567	0,00	100,00	5
	9	0,014567	0,00	100,00	5
	10	0,014567	0,00	100,00	5

Percentual de Nós atingidos na rede (*Spread Network*). A primeira métrica avaliada foi o espalhamento da mensagem na rede, ou seja, quantos nós foram atingidos na disseminação de uma mensagem. Para esta métrica foi calculada a média da quantidade de nós que recebeu cada mensagem em cada experimento e em seguida calculada a média dos 10 experimentos. A Figura 5 apresenta o gráfico desta métrica.

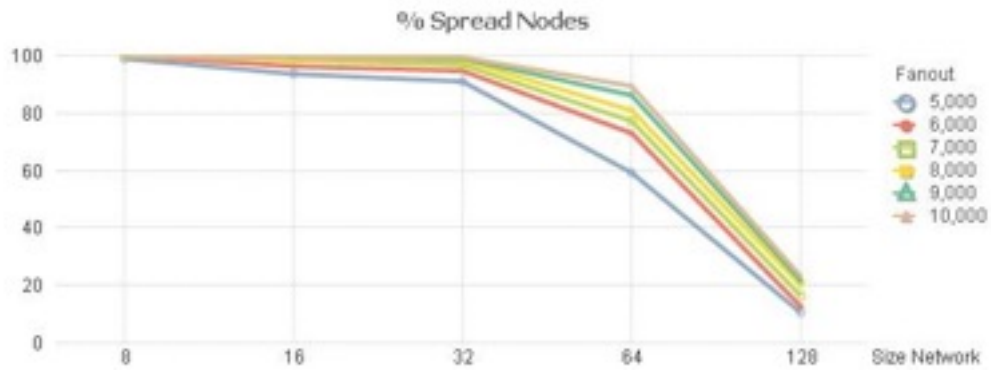

Figura 5. Percentual de nós atingidos na rede (*spread network*)

Observa-se que quanto menor o número de nós, melhor foi o espalhamento da rede e que a disseminação é afetada com o aumento de nós na rede, i.e, um menor numero de nós são atingidos. Mesmo com alta taxa de entrega, nem todos os nós presentes receberam a informação. Observa-se que com maior número de *fanout,* melhor é o espalhamento, porém, como o ambiente possui a variável área e velocidade, alguns nós podem não ser atingidos por estarem fora do alcance. Outra variável que explica esta métrica é a variável *hop* (saltos). Se há poucos saltos da mensagem, mesmo com *fanout* alto, a disseminação pode não ser tão eficaz. Por fim, a estratégia utilizada de não reencaminhar mensagens já recebidas pelo nó pode interferir nesta disseminação. Por um lado diminui-se a redundância de mensagem, mas por outro pode-se afetar a disseminação.

Latência. A Figura 6 apresenta os resultados obtidos para a métrica de latência. Esta métrica foi avaliada considerando a média do tempo inicial do envio da mensagem pelo nó *Source* até o maior tempo de recebimento da mensagem por outro nó que recebe a mensagem pela primeira vez. Em seguida , foi calculada a média novamente dos 10 experimentos executados.

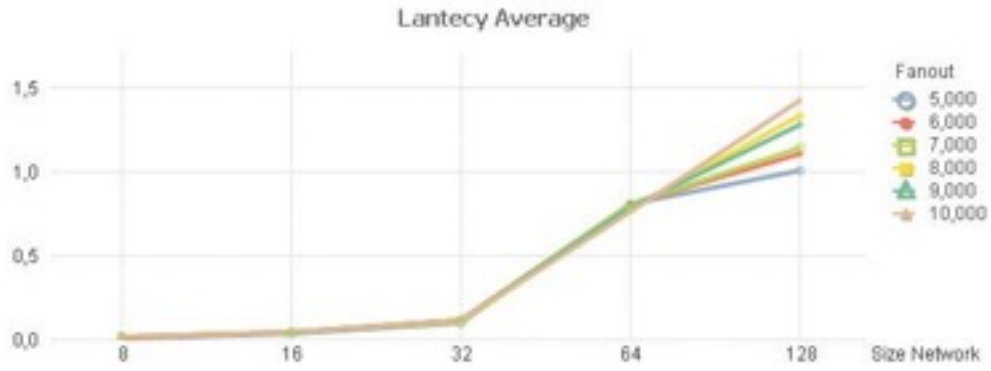

Figura 6. Métrica de Latência.

Observa-se que com o aumento de nós na rede o tempo de recebimento de uma mensagem também aumenta, porém a variável *fanout* não tem tanta influência nesta métrica. Em todos os cenários com a alteração do *fanout,* a latência é pouco alterada. Com exceção do experimento com 128 nós, onde devido ao pouco espalhamento (*spread network*) da mensagem na rede a latência foi alterada.

Taxa de Entrega (Delivery Rate). Esta métrica avalia a taxa de mensagens perdidas na rede. Foi calculada considerando a média da soma de mensagens perdidas de cada nó, em cada experimento, e em seguida aplicada a média dos 10 experimentos realizados. A Figura 7 apresenta os resultados.

Figura 7. Métrica de Taxa de Entrega.

Observa-se novamente que quanto maior for a variável *fanout*, melhor é a taxa de entrega (percentual menor de perda de mensagens), mesmo aumentando a quantidade de nós na rede. Pode-se apurar que a taxa de entrega é afetada devido ao pouco espalhamento da rede (*spread network*), pelos mesmos motivos já descritos. Quanto maior o número de nós, maior é a perda, devido ao baixo *spread* da rede e, consequentemente, a variável *fanout* que limita a disseminação da mensagem e a estratégia de *gossip* utilizada (minimizar a redundância).

Mensagens Duplicadas. A última métrica analisada foi a de recebimento de mensagens duplicadas em cada nó. Para isto, foi calculada a média da soma de mensagens recebidas em duplicidade em cada nó de cada experimento, em seguida aplicado a média dos 10 experimentos. As mensagens recebidas pela primeira vez eram

armazenadas em cada nó da aplicação e a cada novo recebimento da mesma mensagem era gerado um registro de mensagem duplicada no arquivo de log.. A Figura 8 mostra o gráfico da métrica das mensagens duplicadas nos experimentos.

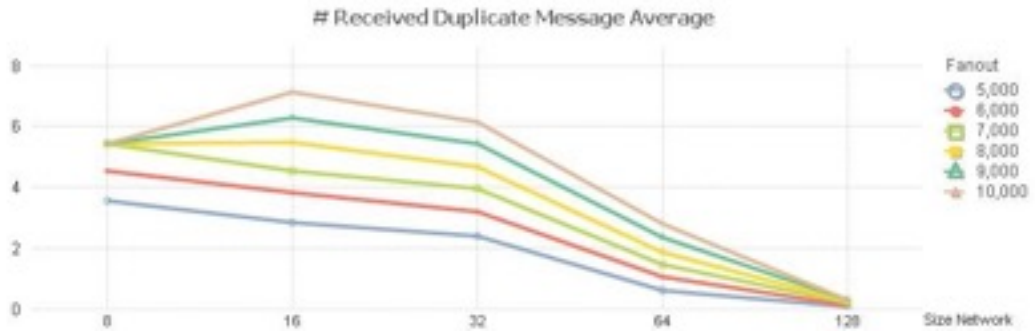

Figura 8. Métrica de Mensagens Duplicadas.

Nota-se novamente que a variável *fanout* age diretamente na métrica de mensagens recebidas em duplicidade por cada nó. Quanto maior o *fanout*, maior a redundância. Isto é explicado uma vez que a variável *fanout* coordena o reencaminhamento de mensagens na rede. A variável *hop*, também interfere nesta métrica, porém, como a estratégia de *gossip* utilizada foi a de não reencaminhar mensagens já recebidas, a redundância foi minimizada. Observa-se que ao aumentar a quantidade de nós na rede, as mensagens duplicadas diminuíram. Isto se deve ao baixo espalhamento da rede (*spread network*) que afeta as entregas de mensagens em todos os nós.

2.2.4. Experimentos com simulador para perda de pacotes

Para o ambiente com acesso móvel à Internet, fluxos de transmissão foram gerados do *host* H1, conectado à rede móvel 3G/4G da Operadora Vivo, com destino ao *host* receptor H3, estando esse conectado à Internet por meio da infraestrutura de rede disponibilizada pela Faculdade de Tecnologia (FT) da Universidade Estadual de Campinas (UNICAMP).

Nesse contexto, a Tabela 5 apresenta os resultados obtidos, respectivamente, acerca do percentual (primeira linha) e da quantidade de pacotes perdidos (segunda linha) para os 10 cenários de injeção de falhas, considerando fluxos de transmissão entre os hosts H1 e H3 por meio de acesso móvel 3G / 4G, acesso por meio de banda larga e acesso por meio da rede local.

Tabela 5. Pacotes perdidos - acesso móvel

Cenário de injeção de falhas	Média 3G/4G	Média Banda Larga	Média Rede Local
Cenário 1, injeção de 50% de falhas	**45% 13,6**	**48,2% 14,6**	**53,6% 16,2**
Cenário 2, injeção de 55% de falhas	**54,2% 16,4**	**59% 17,8**	**49,8% 15**
Cenário 3, injeção de 60% de falhas	**59,6% 18**	**58,2% 17,6**	**65% 19,6**
Cenário 4, injeção de 65% de falhas	**63,2% 19**	**66,4% 20**	**61% 18,4**
Cenário 5, injeção de 70% de falhas	**64,4% 19,4**	**69,8% 21**	**69,6% 21**
Cenário 6, injeção de 75% de falhas	**75,8% 22,8**	**77,6% 23,4**	**78,2% 23,6**
Cenário 7, injeção de 80% de falhas	**81,8% 24,6**	**85% 25,6**	**82,2% 24,8**
Cenário 8, injeção de 85% de falhas	**81,6% 24,6**	**87% 26,2**	**85,2% 25,6**
Cenário 9, injeção de 90% de falhas	**88,4% 26,6**	**88,4% 26,6**	**87,2% 26,2**
Cenário 10, injeção de 95% de falhas	**95,8% 28,8**	**94,2% 28,4**	**93,4% 28,2**

De modo geral, pode-se observar que os fluxos de transmissão entre os *hosts* H1 e H3 apresentaram resultados médios, em termos de pacotes perdidos, condizentes com os percentuais de injeção de falhas para cada cenário em questão em qualquer tipo de conexão; por exemplo, para a injeção de 50% de falhas, houve perda média de 45% dos pacotes transmitidos do *host* H1 para o *host* H3 utilizando a rede 3G/4G (perda média de 13,6 pacotes dos 30 transmitidos entre tais *hosts*), 48,2% utilizando a rede banda larga (perda média de 14,6 pacotes dos 30 transmitidos) e 53,6% utilizando a rede local (perda média de 16,2 pacotes dos 30 transmitidos).

2.2.5. Experimentos com simulador para atrasos de pacotes

Similarmente, em relação às falhas de atraso na transmissão de pacotes, fluxos de transmissão foram gerados do *host* H1, com destino ao *host* receptor H3, estando esse conectado à Internet por meio da infraestrutura de rede disponibilizada pela Faculdade de Tecnologia (FT) da Universidade Estadual de Campinas (UNICAMP), enquanto a ferramenta Netem provocava os atrasos na transmissão.

Nesse contexto, a Tabela 6 apresenta os resultados obtidos acerca da média dos tempos de resposta para os 5 testes de transmissão (de T1 à T5) e considerando os 10 cenários de injeção de atrasos descritos. De modo complementar, a Tabela 6 também apresenta os resultados obtidos em um cenário sem a injeção de atrasos (0 ms), possibilitando sua comparação com os demais cenários.

De modo geral, pode-se observar que em ambos os cenários e testes (com a injeção de atrasos e sem a injeção de atrasos), há uma grande distinção (em ms) entre os tempos de resposta obtidos por meio do utilitário

"ping". Como exemplo, os tempos médios obtidos no cenário com 100ms de atrasos para a rede 3G/4G (728,34 ms) foram inferiores aos tempos obtidos, por exemplo, nos cenários com 20ms de atrasos (779,94ms) e 80ms de atrasos (778,95ms) na mesma rede. De modo similar, o cenário sem a injeção de atrasos (0ms) obteve, em alguns de seus testes, tempos de resposta superiores aos cenários com a injeção de atrasos.

Tabela 6. Tempos de resposta - acesso móvel

	Transmissões com atrasos de										
	0 ms	**10 ms**	**20 ms**	**30 ms**	**40 ms**	**50 ms**	**60 ms**	**70 ms**	**80 ms**	**90 ms**	**100 ms**
	Média	**Média**	**Média**	**Média**	**Média**	**Média**	**Média**	**Média**	**Média**	**Média**	**Média**
Média 3G/4G	**494,14**	**386,50**	**779,94**	**486,44**	**524,82**	**525,10**	**461,22**	**515,01**	**778,95**	**712,10**	**728,34**
Média BLarg	**24,54**	**35,28**	**44,48**	**53,91**	**63,07**	**73,80**	**83,02**	**93,03**	**103,07**	**113,04**	**122,67**
Média Local	**0,55**	**10,57**	**20,64**	**30,70**	**40,67**	**50,66**	**60,64**	**70,68**	**81,23**	**90,62**	**100,66**

Já para o acesso banda-larga cabeado à Internet, pode-se observar que, de modo geral, os tempos obtidos nos testes e cenários envolvendo a injeção de atrasos se mostraram condizentes com os tempos definidos para cada atraso em questão. Como exemplo, tendo como base o cenário sem a injeção de atrasos (0ms), pode-se observar que os tempos obtidos para os demais cenários tiveram suas respostas com valores muito próximos aos definidos em suas respectivas injeções de atraso (acréscimo de 10ms a cada novo cenário, tendo como base o cenário sem atrasos – 0ms).

Para o ambiente de rede local, fluxos de transmissão foram gerados do *host* H1 para o *host* H2, ambos conectados em uma rede local cabeada com enlaces de 100 Mbps. De modo similar ao ambiente banda-larga cabeado, pode-se observar que, de modo geral, os tempos obtidos nos testes e cenários envolvendo a injeção de atrasos se mostraram condizentes com os tempos definidos para cada atraso em questão (aumento aproximado de 10 ms a cada novo cenário).

Outra particularidade que pode ser observada refere-se aos tempos de resposta como um todo. Ao observarmos os valores médio obtidos, nos ambientes envolvendo acesso móvel à Internet, acesso banda-larga à Internet e o ambiente de rede local, há uma grande distinção em relação ao tempo. Nesse contexto, os tempos obtidos na rede local são muito inferiores aos obtidos no acesso banda-larga que, por sua vez, são muito inferiores aos tempos obtidos no acesso móvel 3G/4G.

2.6 Contribuições no segundo ano do projeto

Após apresentado esses resultados, destacamos as contribuições do projeto nesse segundo ano:

- Escalabilidade em termos de numero de nós para teste de comunicação em rede ad hoc com protocolo epidêmico NEeM (5 a 8 nós físicos)
- Elaboração de modelo estatístico para emular o comportamento da rede ad hoc com protocolo epidêmico (*modelo beta inflacionado em um*)

- Desenvolvimento de uma aplicação no simulador NS-3 para o estudo de parâmetros do protocolo epidêmico (espalhamento de mensagem, latecia e taxa de entrega)
- Definição do ambiente experimental 3G / 4G
- Desenvolvimento de um simulador para os experimentos na rede puramente móvel 3G / 4G
- Dois trabalhos completos em conferências internacionais, dois resumos estendidos, sendo um nacional e um internacional
- Artigo a ser publicado em conferência em fase de refinamento da escrita a ser submetido em Março /2016
- Artigo a ser submetido para revista em fase de escrita.
- Dois artigos a ser submetido em congressos internacionais em fase de escrita.
- Dois trabalhos de Iniciação Científica finalizados.
- Trabalho de mestrado em fase de escrita da dissertação.

Referências

ALENAZI M.J.F., CHENG Y., ZHANG D., STERBENZ J.P.G., **Epidemic Routing Protocol Implementation in NS-3**. Workshop on NS-3 ACM WNS3 2015. Barcela, Spain, 2015.

ARLAT, J., COSTES, A., CROUZET, Y., LAPRIE J.C., POWEL D. **Fault Injection and Dependability Evaluation of Fault Tolerant Systems**. IEEE Trans on Sw Engineering, Vol. 16, pp. 166-174, 1993.

CHILLAREGE, R., BOWEN, N. **Understanding Large System Failures – A Fault Injection Experiment**. Proc. of the 19th IEEE Int Symp on Fault Tolerant Computing – FCTS´89, Chicago, Illinois, USA, pp. 356-363, 1989.

CRISTIAN F. **Understanding fault-tolerant distributed systems**. Communications of the ACM, 34(2):56–78, 1991.

DOBRE C., POP F., CRISTEA V. **A Simulation Framework for Dependable Distributed Systems**. In: Proc. of International Conference on Parallel Processing - Workshop, Portland, 2008.

DREBES R., **Firmament: Um Módulo de Injeção de Falhas de Comunicação para Linux**. Dissertação de Mestrado em Ciência da Computação, UFRGS, Porto Alegre, Brasil, 2005.

FALKE M., LI T., UM J., Performance Evaluation of *Gossip* Protocol in Peer-to-Peer Mesh Networks, NS-3, 2015.

GRAY, J. **A Census of Tandem Systems Availability Between 1985 and 1990**. In: IEEE Trans on Reliability. Vol. 39, 1990, pp.409-418.

HORTELANO J., NÁCHER M., CANO J. C., CALAFATE C., MANZONI P. **Castadiva: A Test-Bed Architecture for Mobile Ad Hoc Networks.** IEEE, 2007.

KANOUN K., SPAINHOWER L. **Dependability benchmarking for computer systems**. San Francisco: John Wiley & Sons, 2008. 362 pp.

KOOPMAN, P. & DeVALE, J., **Comparing the Robustness of POSIX Operating Systems** Fault Tolerant Computing Symposium, June 1999.

LEE, I. , IYER, R. **Software Dependability in the Tandem GUARDIAN System.** In: IEEE Trans on Sw Engineering. Vol. 21, 1995, pp. 455-467.

LEITE J., LOQUES O. **Introdução à Tolerância a Falhas.** II SCTF, cap. 4 do minicurso, Campinas, SP, Brasil, 1987.

PEREIRA, J., RODRIGUES, L., MONTEIRO M., OLIVEIRA R., KERMARREC A-M. **NEEM: network-friendly epidemic multicast**. In: Proc. of 22nd Int Symp on Reliable Distributed Systems, 2003. pp. 15-24, 2003.

SCHNEIDER F. B. **What good are models and what models are good?** In Mullender, S., editor, Distributed Systems, 1993, pp. 17–26. Addison-Wesley, Workingham, 2nd edition.

3. Descrição e Avaliação do Apoio Institucional Recebido no Período

O apoio institucional requerido no período foi relacionado à instalação e configuração do ambiente experimental o que foi plenamente atendido pelos técnicos do laboratório de informática da Faculdade de Tecnologia. Quando a complexidade e detalhamento do trabalho excedeu a capacidade dos alunos e do pesquisador para resolver, um técnico treinado para esse tipo de trabalho foi deslocado para atender o projeto.

Também, a Faculdade de Tecnologia disponibilizou um servidor administrativo para auxiliar na prestação de contas, no arquivamento e controle dos documentos.

Estas iniciativas foram importantes para o melhor andamento dos trabalhos efetuados.

4. Descrição da Aplicação dos Recursos de Reserva Técnica e Valores Complementares

Compras
R$ 1092,88 (toners), Nota Fiscal 14.260, E B PEREIRA - ATACADO E VAREJO DE PROD. DE PAP. - ME, em 01/06/2015.
R$ 1079,00 (software Dell, 02 unidades, Nota Fiscal 428145, Magazine Luiza S/A, em 05/10/2015.
R$ 299,00 (equipamento comprado na loja Vivo), Nota Fiscal4.544, em 15/01/2016.

Total - R$ 2.470,88

Serviços de Terceiros
R$ 1.401,23 - Inscrição IEMCON Conference
R$ 2.507,36 - Inscrição IEEE CIT.
Seguro Assistência Viagem, valor R$ 92,00. Sérgio's Turismo Ltda EPP.

Total: R$ 4.000,59

Despesa com Transportes
Sergio's Turismo, valor: R$ 3.376,38. Destino: São Paulo/Vancouver;
Air France, valor: 3.088, 67. Destino: São Paulo/Amsterdam/Birmingham (aeroporto internacional mais proximo de Liverpool)

Total: R$ 6465,05.

Diárias
15/10/2015 até 17/10/2015 - IEMCON'15 - Valor: R$ 3.059,44
26/10/2015 até 28/10/2015 - IEEE CIT - Valor: R$ 3.118,56

Total: R$ 6.178,00.

Total Geral: R$ 16.643, 64. (o valor de R$ 643,64 foi complementado com verba pessoal)