Analisando o desempenho de um sistema de quóruns probabilístico para MANETs diante de ataques maliciosos

Elisa Mannes[1], **Eduardo da Silva**[1], **Aldri L. dos Santos**[1*]

NR2 - Departamento de Informática - Universidade Federal do Paraná
Caixa Postal 19.081 - 81.531-980 – Curitiba – PR – Brasil

{elisamannes,eduardos,aldri}@inf.ufpr.br

Abstract. *Due to their characteristics, such as dynamic topology and lack of infrastructure, Mobile Ad Hoc Networks (MANETs) are highly vulnerable to attacks. These characteristics make hard the design of reliable and secure network management systems. Quorum systems are effective tools for data sharing on traditional networks, ensuring data availability and consistency. Recently, these systems have been applied in MANETs, being PAN, a probabilistic quorum system for ad hoc networks, the first one to consider the characteristics of MANETs. However, such systems do not take into account malicious nodes in the environment. This work analyzes the impact of lack of cooperation, timing and data manipulation attacks against PAN. Two metrics were used in the evaluation: reliability degree and percentage of malicious nodes on the reading operations. Results show that PAN is vulnerable to these attacks, specially to the data manipulation, in which the system correctly conclude only 2% of reading when submitted to 30% of attackers in the writing operations.*

Resumo. *Devido às suas características, como a topologia dinâmica e a ausência de infraestrutura, as Redes Ad hoc Móveis (MANETs) são altamente suscetíveis a ataques. Essas características dificultam o provimento de sistemas de gerenciamento da rede confiáveis e seguros. Os sistemas de quóruns são mecanismos eficazes no compartilhamento de dados em redes convencionais, garantindo a disponibilidade e a consistência dos dados. Esses sistemas têm sido recentemente aplicados nas MANETs, sendo que entre eles destaca-se o PAN. Contudo, tais sistemas não consideram a existência de nós maliciosos no ambiente. Este trabalho analisa o impacto dos ataques de falta de cooperação, de temporização e de manipulação de dados contra o PAN levando em conta o grau de confiabilidade e a quantidade de participação dos nós maliciosos nas operações de leitura. Os resultados mostram que o PAN é vulnerável a esses ataques, principalmente ao ataque de manipulação de dados, no qual o sistema conseguiu concluir corretamente apenas 2% das leituras quando submetido a 30% de nós atacantes nas escritas.*

1. Introdução

As redes *ad hoc* móveis (MANETs) são compostas por dispositivos (nós) que se movimentam livremente e que dependem uns dos outros para a concretização de suas operações

*Este trabalho é apoiado pelo CNPq (Processo: 303770/2008-2)

[Zhou and Haas 1999]. Elas exigem uma administração distribuída e a sua topologia muda constantemente [Chlamtac et al. 2003]. Cada nó comunica-se diretamente com os nós que estão no seu raio de alcance, e para se comunicar com os demais nós, eles utilizam um roteamento com múltiplos saltos [Liu and Kaiser 2003]. Assim, os serviços de gerenciamento de rede dependem da colaboração dos nós, sendo necessário que as operações, além de eficazes, sejam também tolerantes à ação de possíveis nós maliciosos.

Vários serviços de gerenciamento da rede, como os serviços de mobilidade [Pei and Gerla 2001] e os serviços de gerenciamento distribuído de chaves criptográficas [van der Merwe et al. 2007], necessitam da replicação dos dados. Nas redes cabeadas, uma das formas efetivas de replicar dados, garantindo tanto a consistência quanto a disponibilidade dos dados é o uso de sistemas de quóruns [Malkhi and Reiter 1997, Martin et al. 2002]. Um sistema de quóruns é um conjunto de subconjuntos (quóruns) que se intersectam, e cada operação de leitura e de escrita é realizada em apenas um dos quóruns [Alvisi et al. 2000]. Entre as vantagens de seu uso, comparado com a replicação passiva ou ativa convencionais, estão a economia de recursos computacionais e de comunicação, além do aumento da tolerância a falhas. Entretanto, os sistemas de quóruns clássicos assumem a existência de uma infraestrutura fixa, um canal confiável e aplicações assíncronas [Malkhi and Reiter 1997], atributos que não são encontrados em uma MANET.

Considerando as características das MANETs, foram criados sistemas de quóruns específicos para essas redes [Luo et al. 2003, Gramoli and Raynal 2007]. Dentre esses sistemas, destaca-se o PAN (*Probabilistic quorum systems for Ad hoc Networks*) [Luo et al. 2003], por ser um dos poucos que foram avaliados por meio de simulações, enquanto a maioria é apenas teórico. O PAN consiste de um conjunto de estratégias de construção e de acesso aos quóruns que utiliza um mecanismo epidêmico na replicação dos dados [Murray 1993]. Mas, ele não possui mecanismos que o proteja contra ataques maliciosos comumente encontrados nas MANETs [Wu et al. 2006], como os ataques de falta de cooperação, de temporização e de modificação dos dados. Esses ataques, se aplicados contra o sistema de quóruns, podem comprometer a sua eficácia e confiabilidade, e consequentemente afetar o desempenho dos serviços de gerência da rede.

Este trabalho quantifica o impacto desses ataques maliciosos na confiabilidade do sistema de quóruns PAN para MANETs, e destaca quais pontos vulneráveis precisam ser tratados. Para isso, os ataques de falta de cooperação, de temporização e de manipulação de dados foram implementados nas operações de leitura e de escrita do PAN. Os ataques foram simulados no Network Simulator (NS-2), e as métricas utilizadas na análise são o grau de confiabilidade, usado em todos os ataques, e a quantidade de nós intermediários que participaram das operações de leitura, aplicados nos ataques de falta de cooperação e de manipulação de dados, ambos nas leituras. Os resultados mostram que o PAN é vulnerável a esses ataques, que degradam o seu desempenho à medida que aumenta o número de atacantes. Logo, os ataques acarretaram uma baixa confiabilidade no sistema, tendo seu pior desempenho no ataque de manipulação de dados.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 explica o funcionamento do PAN, suas características e suas vulnerabilidades. A Seção 4 descreve os procedimentos dos ataques e as métricas utilizadas na análise. A Seção 5 descreve os parâmetros nas simulações e apresenta os

resultados obtidos, e por fim, a Seção 6 conclui o trabalho e apresenta os trabalhos futuros.

2. Trabalhos relacionados

Diversos sistemas de quóruns têm sido utilizados para a replicação de dados nas redes convencionais, como os sistemas de quóruns dinâmicos [Alvisi et al. 2000], os mínimos [Martin et al. 2002], os síncronos [Bazzi 2000] e os probabilísticos [Malkhi et al. 2001]. Alguns deles consideram ataques bizantinos, como os quóruns f-mascaramento e f-disseminação [Malkhi and Reiter 1997]. Contudo, esses sistemas assumem canais confiáveis e aplicações síncronas e, portanto, não são indicados para as MANETs.

Dentre os sistemas de quóruns criados para MANETs encontram-se o PAN [Luo et al. 2003], que emprega mecanismos de propagação de dados baseados em *gossip* (fofoca), os sistemas de quóruns temporizados [Gramoli and Raynal 2007], que garantem que durante um determinado período de tempo haverá dois quóruns que se intersectam, e os de disseminação móvel [Tulone 2007], que utilizam pontos focais para a criação dos quóruns. Embora considerem características como a topologia dinâmica e a colaboração entre os nós, esses quóruns não consideram a presença de atacantes nas operações e, por isso, podem ser suscetíveis a ação de nós maliciosos.

Os comportamentos de má conduta em MANETs têm sido tratados em diversos trabalhos, como [Hu and Burmester 2009, Holmer 2007, Marti et al. 2000]. Entretanto, a maioria deles considera esses ataques no nível de roteamento e de serviços. Os sistemas de quóruns em geral têm sido avaliados principalmente com foco na Internet [Amir and Wool 1996, Owen and Adda 2006], e não existem estudos específicos que analisam o comportamento desses sistemas para MANETs diante de ataques. Esse trabalho estuda ataques maliciosos aplicados no sistema de quóruns PAN. Tais sistemas devem ser confiáveis e robustos, pois os demais serviços da rede podem utilizá-los como base para o seu correto funcionamento.

3. PAN: Sistema de quóruns probabilístico para redes *ad hoc* móveis

O PAN [Luo et al. 2003] é um sistema de quóruns probabilístico em que a construção dos quóruns baseia-se na probabilidade de intersecção entre eles. Os quóruns Q devem ser construídos segundo a probabilidade $P(Q_1 \cap Q_2 \neq \emptyset) \geq 1 - \varepsilon$, sendo que ε representa um valor muito pequeno. Para isso, o PAN utiliza um protocolo de envio de dados baseado em *gossip* para criar o quórum de escrita e mensagens *unicast* para criar o quórum de leitura. O PAN elege alguns nós da rede para formar um conjunto de armazenamento (StS - *Storage Set*), sendo que essa eleição pode ser feita pelo uso de algoritmos distribuídos [Krishna et al. 1997] ou pela escolha dos nós que possuem mais de um determinado recurso, como energia [Xu et al. 2002]. O restante dos nós são os clientes do sistema, e os servidores, quando estão atendendo pedidos de leitura ou de escrita dos clientes, são chamados de agentes dessa operação.

Para explicar o funcionamento do PAN, assume-se uma rede composta de 40 nós, dos quais 20 deles fazem parte do StS, similar ao descrito em [Luo et al. 2003] (Figuras 1 e 2). Os clientes enviam a requisição de leitura ou de escrita para qualquer um dos servidores do StS. As escritas são propagadas para um determinado número de servidores. O parâmetro que indica esse número de servidores é chamado de *fanout* (*F*), e é configurado antes da inicialização da rede sendo igual para todos os servidores. Nesse

cenário considera-se o $fanout$ igual a dois servidores (F=2). O tamanho do quórum de leitura (ξ) é pré-definido e igual para todos os nós, sendo que os integrantes do quórum são escolhidos aleatoriamente por cada agente, em cada leitura. Utiliza-se o $\xi = 4$, e esse valor inclui o próprio agente da operação. Na Tabela 1 estão descritas as notações usadas na explicação dos procedimentos dos ataques, na Seção 4.

Tabela 1. Notações

Notação	Descrição
s_x	identidade de um nó membro do StS
c_x	identidade de um nó cliente do sistema
M	conjunto de nós maliciosos
ξ	quórum de leitura
msg	mensagem de leitura ou escrita
C	$\mid C \mid = F \wedge \mid \xi - 1 \mid$
F	*fanout*
T	intervalo das propagações
$\Re$	seleção aleatória
$dado_m$	dado malicioso
T_m	tempo malicioso

3.1. Funcionamento

O PAN oferece aos nós clientes as operações de escrita e leitura. Os nós servidores, na condição de agentes, são responsáveis por receber e responder os pedidos dos clientes. Em uma operação de escrita, os agentes difundem a atualização recebida para os demais servidores com a colaboração dos outros nós. Já na operação de leitura eles consultam um quórum de leitura antes de enviar a resposta ao cliente. Essas operações são ilustradas nas Figuras 1 e 2 e acontecem da seguinte maneira:

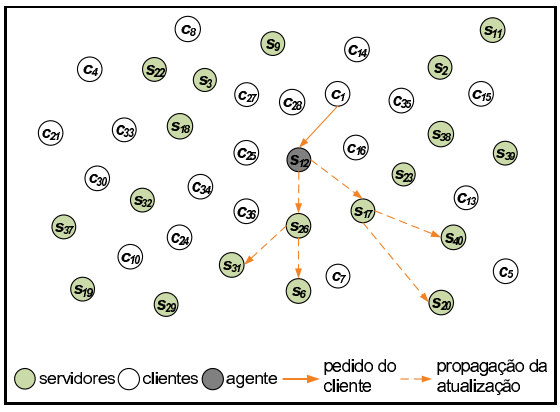

Figura 1. Escrita no PAN

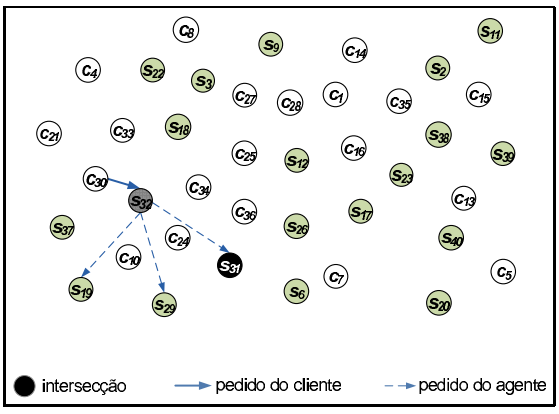

Figura 2. Leitura no PAN

ESCRITA - como parte do mecanismo de escrita, todos os servidores possuem um *buffer* que armazena as atualizações mais recentes de cada dado. A Figura 1 ilustra a execução de uma escrita no StS. Ao receber uma solicitação de escrita do nó cliente c_1, o nó servidor s_{12}, que agora é o agente dessa operação, adiciona a atualização no *buffer* e, em intervalos de tempos regulares, ele propaga essas atualizações. O nó agente, nesse exemplo, escolhe os nós servidores s_{26} e s_{17} para enviar essa atualização. O nó servidor s_{26}, por sua vez, escolhe os nós servidores s_{31} e s_6, enquanto o nó servidor s_{17} envia para os nós servidores

s_{20} e s_{40}. Depois de algum tempo, todos os nós servidores terão o valor atualizado. Ao final, o quórum de escrita é formado pelos nós servidores que receberam a atualização.

LEITURA - quando o nó servidor s_{32} recebe a requisição de leitura do nó cliente c_{30}, ele se torna o nó agente dessa operação, ilustrado na Figura 2. O nó agente escolhe os nós que vão compor o quórum de leitura, e então, encaminha esse pedido juntamente com a sua cópia local do dado que está sendo pesquisado para os nós do quórum, que nesse exemplo são os nós s_{19}, s_{29} e s_{31}. Com isso, os membros do quórum de leitura apenas respondem ao nó agente se possuírem um valor mais atualizado. Nesse caso, o nó agente espera pelas respostas dos nós servidores durante um tempo determinado, e na falta delas, ele responde ao nó cliente com seu próprio dado, por concluir que o seu dado é o mais atualizado. Além disso, caso os nós servidores do quórum recebam um valor mais atualizado do nó agente, eles atualizam a sua cópia local com esse valor e a adicionam no *buffer* para ser disseminado na próxima rodada de propagação. Considerando a escrita anterior, o nó s_{31} é a intersecção entre os quóruns de leitura e de escrita, e fornece o dado atualizado ao nó cliente s_{30}.

3.2. Vulnerabilidades

Os mecanismos utilizados pelo PAN para a construção dos quóruns e para a realização das operações que dão suporte ao serviço de gerenciamento de rede são ideais para o desempenho das MANETs. Porém, eles apresentam vulnerabilidades e podem ser suscetíveis a alguns ataques por parte de nós maliciosos.

Na escrita, o sucesso da operação está diretamente ligado a colaboração de todos os servidores do sistema envolvidos na divulgação da atualização. Deste modo, percebe-se que os nós, enquanto membros do StS, podem prejudicar o andamento dessa propagação de várias formas, entre elas, negando-se a propagá-las ou atrasando o intervalo de tempo entre as propagações. Com isso, as atualizações progridem de uma forma lenta. Isso pode ocorrer se os nós são egoístas e querem economizar recursos ou se os nós são maliciosos e desejam degradar o sistema. Além disso, os nós podem manipular os valores que estão sendo atualizados, causando uma inconsistência no sistema.

Já na leitura, os servidores consultados pelo agente podem deliberadamente não responder aos pedidos, forçando o agente a enviar o seu dado local para o cliente. O agente ou os servidores podem também modificar o valor do seu dado e enviar ao cliente esse valor manipulado. No primeiro caso, o sistema já prevê o tratamento de leituras em que os clientes não respondem (pelo uso de um *timeout*), minimizando o impacto do comportamento desses nós egoístas. Entretanto, o segundo caso é mais grave, pois além de fornecer ao cliente um dado errado, os servidores consultados também podem confiar na informação enviada pelo agente e atualizar as suas cópias. Consequentemente, o valor falso será propagado entre os servidores, comprometendo a confiabilidade do sistema.

4. Ataques e métricas de avaliação

Essa seção apresenta a forma como os ataques implementados exploram as vulnerabilidades do PAN e as métricas empregadas para a quantificação desses ataques.

4.1. Falta de cooperação

O ataque de falta de cooperação entre os nós acontece quando os nós comprometidos se recusam a executar a sua parte no andamento do sistema. O Procedimento 1 descreve

o comportamento de um nó comprometido e de um nó não-comprometido no PAN, em uma leitura. Na condição de agente, esses nós não consultam os membros do quórum de leitura para pesquisar o valor do dado que eles possuem. Sendo assim, o agente responde ao cliente diretamente com o dado que ele possui (linhas 1-4). Dependendo da velocidade de propagação de uma escrita e das condições gerais da rede, a probabilidade do agente ter o dado atualizado, exatamente quando um cliente realiza uma leitura, se torna pequena e, nesse caso, o cliente tem uma grande probabilidade de receber um dado desatualizado.

Procedimento 1 ATAQUE DE FALTA DE COOPERAÇÃO - LEITURA

nó servidor s_x ***ao receber uma requisição de leitura do cliente*** c_y

1: **se** $s_x \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
2: $msg \leftarrow dado_{local}$
3: $envia(msg, c_y)$ {*responde ao cliente com seu próprio dado sem consultar o quórum de leitura*}
4: **senão**
5: $msg \leftarrow dado_{local}$
6: $dest \leftarrow C \subset_{\Re} StS : |C| = |\xi| - 1$ {*escolhe os nós do quórum de leitura*}
7: **para todo** $s_w \in dest$ **faça**
8: $envia(msg, (req_{leitura}, s_w))$ {*envia a requisição de leitura e a sua cópia do dado*}
9: $timer_{s_w} \leftarrow 0$ {*inicia um timer para as requisições de leitura*}
10: **fim para**
11: **fim se**

nó s_w ***ao receber uma requisição de leitura do agente*** s_x

12: **se** $dado_{recebido} > dado_{local}$ **então**
13: **se** $s_w \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
14: $ignora$ {*não responde*}
15: **senão**
16: $msg \leftarrow dado_{local}$
17: $envia(msg, s_x)$ {*responde ao agente com o seu dado*}
18: **fim se**
19: **fim se**

Quando o nó é intermediário em uma operação de leitura, ele simplesmente deixa de responder ao pedido do agente (linhas 14-16). Nesse caso, uma leitura ainda pode ser corretamente concluída se pelo menos um dos servidores de um quórum de leitura esteja íntegro e responda ao agente, ou se o próprio agente tenha o dado atualizado.

Procedimento 2 ATAQUE DE FALTA DE COOPERAÇÃO - ESCRITA

nó s_x ***ao receber uma requisição de escrita***

1: **se** $dado_{recebido} > dado_{local}$ **então**
2: **se** $s_x \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
3: $ignora$ {*não grava o dado e não adiciona no buffer*}
4: **senão**
5: $dado_{local} \leftarrow dado_{recebido}$ {*atualiza o dado*}
6: *buffer* $\leftarrow$ *buffer* $+ dado_{local}$ {*adiciona no buffer*}
7: **fim se**
8: **fim se**

Já na operação de escrita, o nó comprometido, quando agente, não atualiza o seu dado e também não propaga o pedido do cliente para os outros servidores. Na condição de servidor intermediário que recebeu uma atualização por meio do *gossip*, o nó comprometido não escreve essa atualização e também não grava no *buffer* (linhas 1-3). Como

consequência, a propagação da escrita para os outros servidores não progride a partir desse nó. Ambas as situações são descritas no Procedimento 2.

Esse tipo de ataque de falta de cooperação, em que os nós não cooperam mas a conclusão da operação acontece de qualquer forma, se torna particularmente difícil de detectar, pois os clientes recebem uma resposta dos agentes mesmo que essa seja uma resposta desatualizada. Nesse caso, mesmo os mecanismos de reputação ou de diagnóstico têm dificuldade em identificar e isolar os nós maliciosos.

4.2. Temporização

O intervalo de tempo entre a propagação das escritas é um parâmetro importante no PAN. Ele determina a rapidez da divulgação de uma escrita, sendo que quanto mais rápido for essa propagação, mais rápido a escrita atinge todos os nós do StS, fazendo com que a formação do quórum de escrita seja mais rápida. O comportamento dos nós comprometidos é descrito no Procedimento 3.

Procedimento 3 ATAQUE DE TEMPORIZAÇÃO

ao expirar o timer do s_x

1: **enquanto** $buffer \neq \emptyset$ **faça**
2: $msg \leftarrow entrada \in buffer$
3: $buffer \leftarrow buffer \setminus entrada$ {*retira a entrada do buffer*}
4: $dest \leftarrow C \subset_{\Re} StS : |C| = F$ {*escolhe os servidores para propagar a atualização*}
5: **para todo** $s_w \in dest$ **faça**
6: $envia(entrada, s_w)$ {*envia a atualização*}
7: **fim para**
8: **fim enquanto**
9: **se** $s_x \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
10: $timer = T_m$ {*modifica o intervalo de propagação*}
11: **senão**
12: $timer = valor\_do\_sistema$ {*valor constante definido pelo sistema*}
13: **fim se**

Nesse tipo de ataque, os nós do StS que estão comprometidos atrasam a propagação da escrita, esperando um tempo maior (linha 10) que o definido inicialmente para enviar as atualizações para os demais nós.

4.3. Manipulação de dados

O ataque de manipulação de dados consiste em nós que recebem um dado e o modificam, seja na escrita ou na leitura. Essa situação é descrita no Procedimento 4. Em uma operação de leitura, o nó malicioso agente modifica o dado antes de consultar os outros quóruns (linhas 1-10). Nesse caso, o valor modificado sempre será maior que o dos demais servidores. Os servidores, portanto, não vão responder para o agente por acreditar que o seu dado está desatualizado. Isso é agravado pois os servidores que foram consultados, ao verificar que o agente tem um valor mais atual que o deles, atualizam os seus valores e propagam para outros nós, começando uma epidemia de um valor errado.

Se o servidor malicioso for um servidor do quórum de leitura consultado pelo agente, ele modifica o valor da sua cópia local e a repassa para o agente (linhas 22-29). O valor do dado recebido pelo agente será maior, e nesse caso, o agente além de atualizar a sua cópia, também propagará esse valor.

Procedimento 4 ATAQUE DE MANIPULAÇÃO - LEITURA

nó servidor s_x ***ao receber uma requisição de leitura do cliente*** c_y

1: **se** $s_x \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
2: $dado_{local} \leftarrow dado_m$ {*modifica o dado local*}
3: *buffer* $\leftarrow$ *buffer* $+ dado_{local}$ {*adiciona o dado local no buffer*}
4: $msg \leftarrow dado_{local}$
5: $dest \leftarrow C \subset_{\Re} StS : |C| = |\xi| - 1$ {*escolhe os nós do quórum de leitura*}
6: **para** $s_w \in dest$ **faça**
7: $envia(msg, (req_{leitura}, s_w))$ {*envia a requisição de leitura e a sua cópia do dado*}
8: $timer_{s_w} \leftarrow 0$ {*inicia um timer para as requisições de leitura*}
9: **fim para**
10: **senão**
11: $dado_{local} \leftarrow dado_{recebido}$
12: *buffer* $\leftarrow$ *buffer* $+ dado_{local}$ {*adiciona o dado recebido no buffer*}
13: $msg \leftarrow dado_{local}$
14: $dest \leftarrow C \subset_{\Re} StS : |C| = |\xi| - 1$ {*escolhe os nós do quórum de leitura*}
15: **para** $s_w \in dest$ **faça**
16: $envia(msg, (req_{leitura}, s_w))$ {*envia a requisição de leitura e a sua cópia do dado*}
17: $timer_{s_w} \leftarrow 0$ {*inicia um timer para as requisições de leitura*}
18: **fim para**
19: **fim se**

nó s_w ***ao receber uma requisição de leitura do agente*** s_x

20: **se** $dado_{recebido} > dado_{local}$ **então**
21: **se** $s_w \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
22: $dado_{local} \leftarrow dado_m$ {*modifica o dado local*}
23: $msg \leftarrow dado_{local}$
24: $envia(msg, s_x)$ {*responde ao agente com o dado malicioso*}
25: **senão**
26: $msg \leftarrow dado_{local}$
27: $envia(resp, s_x)$ {*responde ao agente com o seu dado*}
28: **fim se**
29: **fim se**

Na escrita, apresentada no Procedimento 5, o nó malicioso agente modifica o dado (linhas 1-4) em vez de propagar o dado originalmente recebido. Ele faz isso modificando o valor do dado e anexando um *timestamp* maior (linha 3). Depois, a propagação segue normalmente. Um nó intermediário comprometido também apresenta o mesmo comportamento, modificando o dado recebido do agente e propagando esse valor modificado.

Procedimento 5 ATAQUE DE MANIPULAÇÃO - ESCRITA

nó s_x ***ao receber uma requisição de escrita***

1: **se** $dado_{recebido} > dado_{local}$ **então**
2: **se** $s_x \in M$ {*verifica se o nó faz parte do conjunto de nós maliciosos*} **então**
3: $dado_{local} \leftarrow dado_m$ {*modifica o dado*}
4: *buffer* $\leftarrow$ *buffer* $+ dado_{local}$ {*adiciona o dado malicioso no buffer*}
5: **senão**
6: $dado_{local} \leftarrow dado_{recebido}$ {*atualiza a cópia*}
7: *buffer* $\leftarrow$ *buffer* $+ dado_{local}$ {*adiciona o dado recebido no buffer*}
8: **fim se**
9: **fim se**

4.4. Métricas de avaliação

A primeira das duas métricas utilizadas para quantificar o desempenho do PAN sob ataques é o grau de confiabilidade (G_c) [Luo et al. 2003]. O G_c quantifica a probabilidade dos quóruns de escrita e de leitura se intersectarem, além de reproduzir a quantidade de leituras corretamente obtidas pelo cliente. São consideradas leituras corretas aquelas que retornam ao cliente o último ou o penúltimo valor escrito no StS. Isso ocorre porque a última escrita anterior a leitura pode ainda estar em curso, e nesse caso, nem todos os nós têm o valor atualizado. Por isso, a escrita anterior a última também é contabilizada como correta. O G_c é definido conforme a Equação 1, em que L_c representa as leituras corretas concluídas e L o total de leituras emitidas.

$$G_c = \frac{\sum L_c}{|L|} \tag{1}$$

A segunda métrica é a quantidade de vezes que os nós maliciosos (Q_m) participaram das operações de leitura do PAN, na condição de intermediários. Essa métrica é válida para os ataques de falta de cooperação e de manipulação de dados, pois o ataque de temporização é realizado somente na operação de escrita. Essa métrica é expressa pela Equação 2, em que Lm representa uma leitura em que um membro do quórum de leitura é malicioso, e Q_l representa o quórum de leitura.

$$Q_m = \frac{\sum Lm_i}{|L|} \forall i \in L \quad \text{em que} \quad Lm_i = \begin{cases} 1 & \text{se} \quad \exists m \in Q_l(L_i) \\ 0 & \text{caso contrário} \end{cases} \tag{2}$$

5. Avaliação dos resultados

O PAN foi implementado no NS versão 2.33, usando um canal sem fio, em associação ao modelo de propagação *TwoRayGround*. A rede é composta por 50 nós, sendo que metade deles compõem o StS. Os nós movimentam-se de acordo com o padrão *Random Waypoint* em uma área de 1000x1000 metros, e possuem velocidades máximas de 2m/s, 5m/s, 10m/s e 20m/s com o tempo de pausa de 10s, 20s, 40s e 80s, respectivamente. Todos eles têm um raio de transmissão de 250 metros, e utilizam o AODV como protocolo de roteamento. As atualizações contidas no *buffer* são propagadas a cada 200ms, e o *fanout* é igual a dois servidores. O quórum de leitura é composto por quatro servidores, incluindo o agente. Os parâmetros anteriores são definidos no cenário do artigo do PAN. A quantidade de atacantes (f) é de 20%, 28% e 36% dos nós do StS, o que corresponde a 5, 7 e 9 nós atacantes, respectivamente. Um número maior de atacantes leva o sistema a um estado em que a aplicação é totalmente comprometida, o que torna inviável a execução da replicação.

Os pacotes com as requisições têm tamanho de 128 *bytes*, o suficiente para o tipo de dados que deseja-se replicar. Da mesma forma que em [Luo et al. 2003], as escritas e as leituras têm o seu intervalo determinado por uma distribuição de *Poisson*. A escrita tem um intervalo médio de 6 segundos e a leitura de 0,4 segundos. Os resultados apresentados são a média de 35 simulações com um intervalo de confiança de 95%, e o tempo de vida da rede é de 1500 segundos.

5.1. Falta de cooperação

A Figura 3 apresenta o resultado das simulações do PAN na presença de nós maliciosos que não cooperam com as operações. À medida que a quantidade de nós atacantes aumenta, o G_c diminui em ambas as operações. Na operação de leitura, um cenário com 20% de atacantes e velocidade de 10m/s tem um G_c de 95,4%. Quando a quantidade de atacantes aumenta para 28%, o G_c diminui para 93,5% nessa mesma velocidade. Com 36% de nós maliciosos e velocidade de 20m/s, o G_c é de 89,3%. Isso ocorre porque as leituras só serão totalmente comprometidas se todos os nós do quórum de leitura forem maliciosos. De outra forma, haverá servidores íntegros para responder à requisição dos agentes, compensando a falta de cooperação dos nós maliciosos.

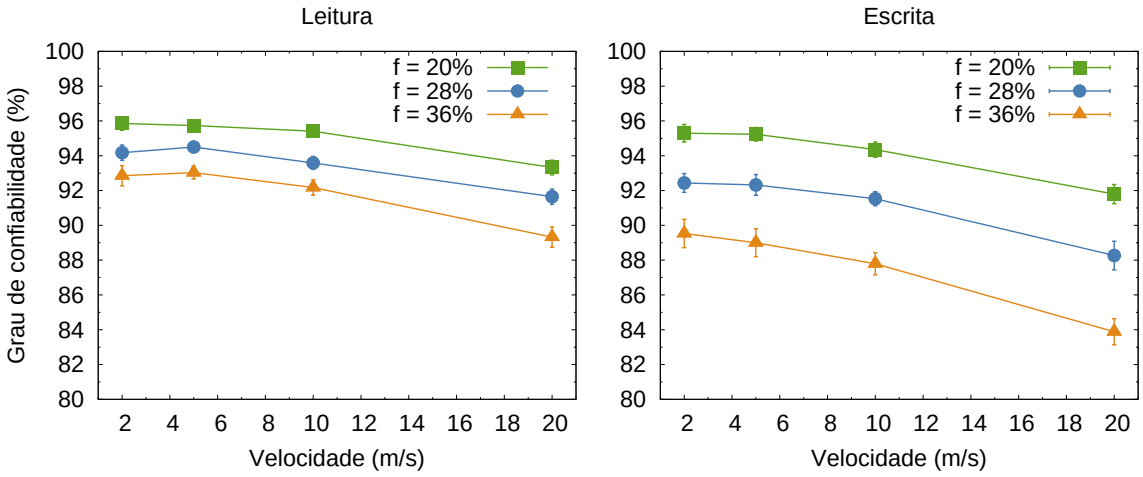

Figura 3. Ataque de falta de cooperação

Na escrita, os nós maliciosos conseguiram comprometer uma porcentagem maior de leituras. Com 20% de nós atacantes e velocidade de 10m/s, o G_c é de 94,3%. Ao aumentar a quantidade de atacantes para 28%, o G_c é de 92,4% nos cenários com velocidade de 2m/s e 88,2% com velocidade de 20m/s. Ao aumentar f para a quantidade máxima de atacantes considerada nas simulações, 36%, e velocidade de 20m/s, o G_c é de 83,8%. Nessa operação, a cooperação dos nós é muito importante, e quando os nós não cooperam na propagação, o grau de confiabilidade diminui gradativamente.

5.2. Temporização

Os nós maliciosos nesse tipo de ataque foram configurados para emitir as propagações a cada 400ms, 800ms e 3000ms. Os resultados são encontrados na Figura 4. Nas simulações em que há 20% dos nós do StS atrasando as propagações em T=400ms e com velocidade de 5m/s, o G_c é de 98,9%. Com o T=800ms e velocidade de 20m/s o G_c é de 96,9% e com o T=300ms, o G_c é de 96,2% considerando essa mesma velocidade.

Quando o número de atacantes aumenta para 28%, os cenários com velocidade de 2m/s são os mais afetados, sendo o seu G_c com T=400ms de 98,3%. Com o T=800ms e velocidade de 2m/s, o G_c tem uma pequena queda, e é de 97%, uma diferença de 1%. Já o cenário com velocidade de 10m/s consegue um G_c de 98,1%. Com o T=3000ms e velocidade de 2m/s, o G_c é de 95,8%, e com velocidade de 20m/s ele consegue concluir corretamente 95,7% das suas leituras.

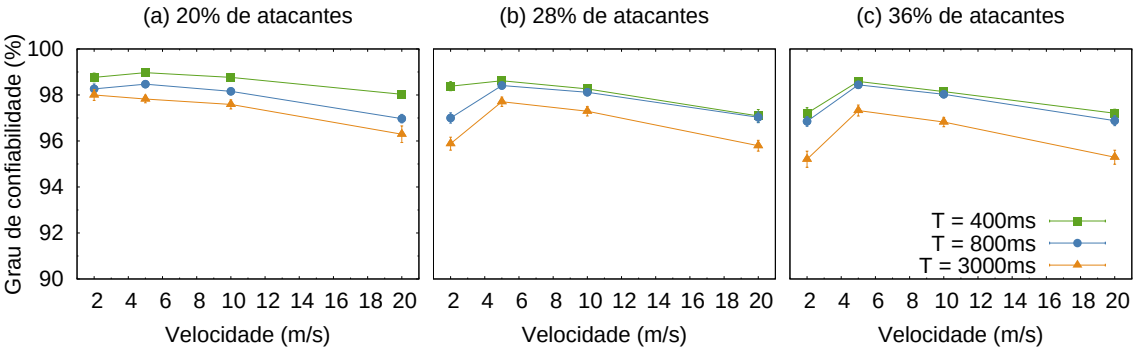

Figura 4. Ataque de temporização no protocolo *gossip*

Os cenários em que os atacantes são 36% da rede e com velocidade de 2m/s apresentam um G_c de 97,2%, enquanto que os cenários com o T=800ms o G_c com essa mesma velocidade é de 96,8%, uma diferença de apenas 0,4%. Com o T=3000ms e velocidade de 2m/s o G_c foi de 95,2%. O PAN não sofre um impacto relevante nesse ataque pela eficácia da propagação das escritas feitas pelos nós que não estão comprometidos, que conseguem manter boa parte da rede com os dados atualizados.

5.3. Manipulação

O ataque de manipulação de dados é o que mais impacta no desempenho do PAN. Isso ocorre pela forma com que as atualizações são propagadas pela rede, e pelo fato que uma leitura pode ser também uma escrita, no caso do valor ser mais atual do que o presente no nó que está fazendo a leitura. A Figura 5 apresenta os resultados para esse ataque. Na leitura, com a quantidade de 20% de atacantes e velocidade de 2m/s, o G_c é de 31,1% e com velocidade de 10m/s, é de 33%. Ao aumentar o número de atacantes para 28% e velocidade de 2m/s o G_c caiu para 24%, uma queda de 7,1%. Esse mesmo cenário com velocidade de 20m/s apresenta um G_c de 25,6%. Com 36% dos nós comprometidos e velocidade de 2m/s, o PAN consegue concluir 19,6% das leituras corretamente, enquanto com velocidade de 20m/s o G_c é de 21,1%.

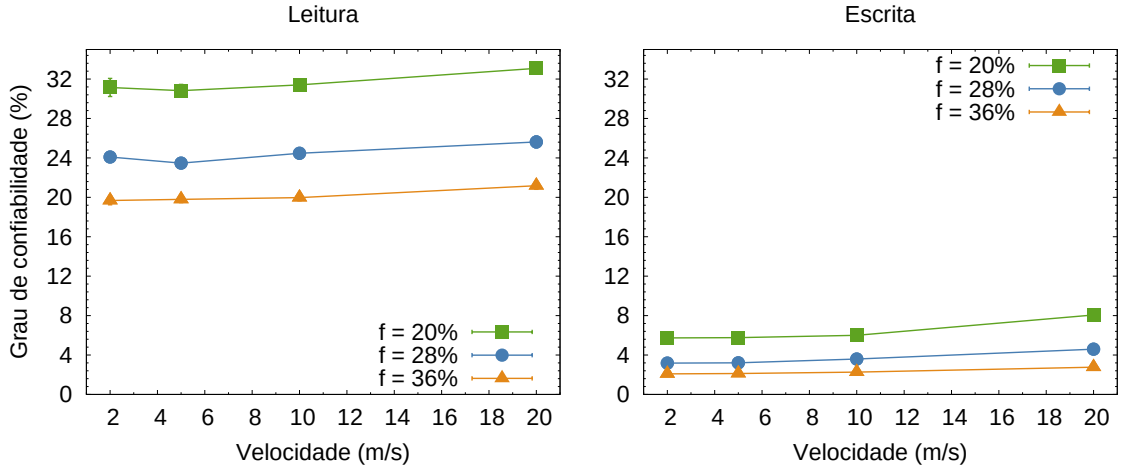

Figura 5. Ataque de manipulação

Na escrita o G_c tem um desempenho muito inferior ao alcançado com os outros ataques. Nesse caso, com velocidade de 2m/s e 20% de atacantes, o G_c é de apenas 5,7%.

Com 10m/s é de 6% e com 20m/s o G_c sobe para 8%. Com 28% de atacantes e com velocidade de 2m/s o G_c cai para 3,1%, e com 20m/s, para 4,5%. O cenário em que os atacantes são 36%, todas as velocidades apresentam um G_c semelhante. Para as velocidades de 2m/s, 5m/s, 10m/s e 20m/s, o G_c é de 2%, 2,1%, 2,2% e 2,7%, respectivamente.

Em uma MANET, em geral, quanto maior a velocidade, mais difícil é para os nós alcançarem uns aos outros, devido a dificuldade em manter as rotas entre eles. Isso faz com que o G_c no ataque de manipulação tenha um comportamento diferente dos demais ataques. Enquanto nos ataques de falta de cooperação e de temporização o G_c diminui à medida que a velocidade máxima de movimentação aumenta, no ataque de manipulação acontece o inverso. Isso ocorre pelo fato de que menos escritas são propagadas, e desse modo, a eficácia de um nó malicioso propagar o seu dado modificado diminui.

5.4. Participação de nós maliciosos nas leituras

Para os ataques que acontecem na leitura, foi contabilizada a quantidade de vezes que os nós maliciosos participaram de uma leitura, na condição de membros do quórum de leitura. No ataque de falta de cooperação, ilustrado na Figura 6, quando há 20% de nós maliciosos e velocidade de 2m/s, 41,5% do total de leituras é comprometido, e 38,3% do total de leituras é comprometido com em um cenário com velocidade de 20m/s. Ao aumentar a quantidade de atacantes para 28%, o número de leituras comprometidas sobe para 51,4% com velocidade de 5m/s, e para 48,9% com 20m/s. Com 36% dos nós comprometidos e velocidade de 5m/s, as leituras comprometidas são de 58,5% e de 55% com velocidade de 20m/s.

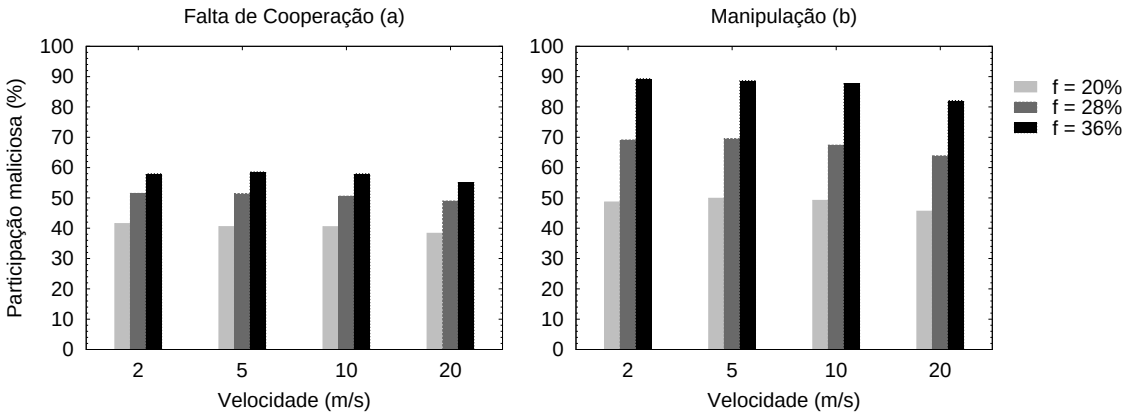

Figura 6. Participação dos nós intermediários nos ataques à leitura

Já no ataque de manipulação de dados, a quantidade de leituras comprometidas é de 49,8% com a velocidade de 2m/s, e de 45,6% com a velocidade de 20m/s. Ao aumentar o número de atacantes para 28%, o número de leituras comprometidas também aumenta, sendo que com uma velocidade de 2m/s a quantidade de leituras comprometidas é de 69,1%, um aumento de 19,3% na quantidade de leituras comprometidas. Quando o número de atacantes é de 36%, esse aumento é ainda maior. O cenário com velocidade de 2m/s tem 89,3% de suas leituras comprometidas, e com a velocidade de 20m/s, 82% de suas leituras são comprometidas.

Pode-se perceber que conforme a velocidade máxima de movimentação dos nós aumenta, a quantidade de leituras comprometidas por nós maliciosos diminui. Isso ocorre

porque ao aumentar a velocidade, menos pacotes são entregues. Logo, menos nós pertencentes ao quórum de leitura são consultados e entregam o dado malicioso ao agente.

6. Conclusões

Este trabalho quantificou o impacto de ataques maliciosos nas operações do PAN, um sistema de quóruns probabilístico para MANETs. Os ataques analisados foram os de falta de cooperação, de temporização e de manipulação de dados. As simulações consideraram as métricas de grau de confiabilidade e quantidade de nós maliciosos nas operações de leitura.

Os resultados das simulações mostraram que o PAN, apesar de considerar as características das MANETs na construção e operação dos quóruns, é suscetível a esses ataques. Uma rede com 20% de nós que não cooperam nas leituras, afeta cerca de 10% na confiabilidade do sistema com uma velocidade de 20m/s, enquanto que na escrita esse valor é de aproximadamente 16,5%. Isso ocorre porque os nós comprometidos se recusam a responder aos pedidos dos agentes e dos demais nós, forçando o agente a enviar o seu próprio dado para o cliente. O ataque de temporização apresentou seu pior desempenho quando 36% dos nós atrasaram a propagação da escrita em 3000ms, conseguindo concluir corretamente 95,5% das leituras com uma velocidade máxima de 20m/s. Esse impacto é devido aos nós maliciosos atrasarem a propagação da escrita pelo sistema, e desse modo, os nós levam mais tempo para serem atualizados.

O ataque de manipulação de dados teve um alto impacto no PAN, sendo que com 28% de nós maliciosos na escrita o sistema tem 96,6% das suas leituras comprometidas e com 36% de nós maliciosos, 97,8% das leituras são comprometidas, ambos com uma velocidade máxima de 2m/s. Essa baixa confiabilidade no sistema acontece porque os nós maliciosos, nesse tipo de ataque, modificam os dados, e o mecanismo com que o sistema propaga as atualizações faz com que esse dado modificado chegue facilmente aos outros nós da rede. Como trabalho futuro, pretendemos propor um modelo em que a execução do PAN não seja afetada com a presença desses ataques na rede, além de validar os resultados no contexto de uma aplicação, como no gerenciamento de chaves em MANETS.

Referências

Alvisi, L., Pierce, E. T., Malkhi, D., Reiter, M. K., and Wright, R. N. (2000). Dynamic byzantine quorum systems. In *Proceedings of the 2000 International Conference on Dependable Systems and Networks (DSN '00)*, pages 283–292, Washington, DC, USA. IEEE Computer Society.

Amir, Y. and Wool, A. (1996). Evaluating quorum systems over the internet. In *Proceedings of the 26th Annual International Symposium on Fault-Tolerant Computing (FTCS '96)*, pages 26–35, Washington, DC, USA. IEEE Computer Society.

Bazzi, R. A. (2000). Synchronous byzantine quorum systems. *Distributed Computing*, 13(1):45–52.

Chlamtac, I., Conti, M., and Liu, J. J. (2003). Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64.

Gramoli, V. and Raynal, M. (2007). *Principles of distributed systems*, chapter Timed Quorum Systems for Large-Scale and Dynamic Environments, pages 429 – 442. Springer Berlin.

Holmer, D. (2007). *Byzantine survivable routing for mobile ad hoc networks*. PhD thesis, Baltimore, MD, USA. Adviser-Awerbuch, Baruch.

Hu, J. and Burmester, M. (2009). *Cooperation in Mobile Ad Hoc Networks*, chapter 3, pages 1–15. Springer London.

Krishna, P., Vaidya, N. H., Chatterjee, M., and Pradhan, D. K. (1997). A cluster-based approach for routing in dynamic networks. *Computing Commununication Review (SIGCOMM)*, 27(2):49–64.

Liu, C. and Kaiser, J. (2003). A survey of mobile ad hoc network routing protocols. Technical Report TR 2003-08, Department of Computer Structures - University of Ulm, Germany.

Luo, J., Hubaux, J.-P., and Eugster, P. T. (2003). PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03)*, pages 1–12, New York, NY, USA. ACM.

Malkhi, D. and Reiter, M. (1997). Byzantine quorum systems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pages 569–578.

Malkhi, D., Reiter, M. K., Wool, A., and Wright, R. N. (2001). Probabilistic quorum systems. *Information Computing*, 170(2):184–206.

Marti, S., Giuli, T. J., Lai, K., and Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)*, pages 255–265, New York, NY, USA. ACM.

Martin, J.-P., Alvisi, L., and Dahlin, M. (2002). Minimal byzantine storage. Technical report, University of Texas at Austin, Department of Computer Sciences, Austin, Texas, USA.

Murray, J. (1993). *Mathematical biology*. Springer, Berlin, 2 edition.

Owen, G. and Adda, M. (2006). Simulation of quorum systems in ad hoc networks. In *3rd International Conference on Artificial Intelligence in Engineering and Technology*.

Pei, G. and Gerla, M. (2001). Mobility management for hierarchical wireless networks. *Mobile Network Applications*, 6(4):331–337.

Tulone, D. (2007). Ensuring strong data guarantees in highly mobile ad hoc networks via quorum systems. *Ad Hoc Networks*, 5(8):1251–1271.

van der Merwe, J., Dawoud, D., and McDonald, S. (2007). A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computing Survey*, 39(1):1–45.

Wu, B., Chen, J., Wu, J., and Cardei, M. (2006). *Wireless/Mobile network security*, chapter 12, pages 103–136. Springer-Verlag, New York, NY, USA.

Xu, K., Hong, X., and Gerla, M. (2002). An ad hoc network with mobile backbones. In *IEEE International Conference on Communications (ICC '02)*, volume 5, pages 3138–3143 vol.5.

Zhou, L. and Haas, Z. J. (1999). Securing ad hoc networks. *IEEE Network*, 13(6):24–30.