Sistema de quóruns probabilístico para MANETs tolerante a ataques de manipulação de dados

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

[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. *Although quorum systems with relaxed intersections are effective mechanisms for data replication in MANETs, in general they do not consider the existence of malicious nodes in their operations. Such systems, when facing data manipulation attacks, have the consistency and the availability of data affected, invalidating many of the read operations in the system. This paper presents a system for data replication, to be apllied in management and performance configuration of MANETs, based on probabilistic quorums, tolerant data manipulation attacks. In the proposed scheme, the writing is performed through a gossip protocol, and reading is made using a data validation based on f-masking. The results obtained through simulations show that the proposed scheme improves data integrity, compared to other probabilistic quorum system.*

Resumo. *Embora os sistemas de quóruns com intersecção flexibilizada sejam mecanismos eficazes na replicação de dados em MANETs, em geral eles não consideram a existência de nós maliciosos nas suas operações. Tais sistemas, ao enfrentar ataques de manipulação de dados, têm a consistência e a disponibilidade de dados prejudicadas, invalidando muitas das operações de leitura no sistema. Este trabalho apresenta um sistema de replicação de dados para gerência de configuração e desempenho de MANETs baseado em quóruns probabilísticos, tolerante a ataques de manipulação de dados. No esquema proposto, a escrita dos dados é realizada por meio de um protocolo gossip, e as leituras usam uma validação de dados baseada em f-mascaramento. Os resultados obtidos por meio de simulações mostram que o esquema proposto melhora a integridade dos dados, comparado a outro sistema de quórum probabilístico.*

1. Introdução

As redes ad hoc móveis (*Mobile Ad Hoc Networks* (MANETs)) são compostas por um conjunto de dispositivos móveis (nós) que se comunicam entre si por um canal compartilhado de comunicação sem fio. Essas redes são dinâmicas, não dependem de uma infraestrutura fixa ou administração centralizada, e o seu funcionamento é mantido pelos próprios nós de uma forma auto-organizada e distribuída [Zhang et al. 2008]. Essas características são interessantes para algumas aplicações como o estabelecimento de comunicação entre dispositivos em operações militares, recuperação de desastres, compartilhamento de dados em conferências e computação ubíqua [Chlamtac et al. 2003].

Garantir a consistência e a disponibilidade de dados compartilhados e replicados é fundamental para algumas aplicações e serviços de gerencimanento nas MANETs [Tulone 2007]. Porém, pelas características dessas redes, garantir um alto nível de

consistência dos dados é uma tarefa difícil. Portanto, em geral são empregados mecanismos baseados em protocolos probabilísticos, que relaxam as restrições de consistência do ambiente [Luo et al. 2003, Gramoli and Raynal 2007, Tulone 2007]. Algumas aplicações que podem tolerar consistências relaxadas são os serviços de gerência de configuração e desempenho [Pei and Gerla 2001].

Nas redes tradicionais, os sistemas de quóruns [Malkhi and Reiter 1997] têm sido amplamente utilizados como um mecanismo efetivo de replicação de dados, garantindo tanto a consistência quanto a disponibilidade dos dados. Entre as vantagens de seu uso, comparado com a replicação passiva ou ativa convencionais, estão a economia de recursos computacionais e de comunicação, e o aumento da tolerância a falhas. Diversos esquemas de contrução para sistemas de quóruns têm sido propostos para as MANETs, geralmente considerando a consistência relaxada das informações. A maioria desses esquemas empregam os conceitos de quóruns probabilísticos, como o PAN [Luo et al. 2003], que garante com uma alta probabilidade a consistência entre as leituras e escritas dos dados replicados.

Entretanto esses esquemas baseados em quóruns, quando aplicados nas MANETs, não consideram a existência de entidades maliciosas. Por exemplo, alguns estudos mostram que o PAN é altamente vulnerável aos ataques de manipulação de dados [Mannes et al. 2009]. Esses ataques comprometem a integridade e a confiabilidade dos dados armazenados no sistema. Como as MANETs são altamente suscetíveis a ataques passivos e ativos [Wu et al. 2006], é muito importante que as aplicações de rede sejam tolerante a ataques, em especial àqueles comumente encontrados nesses redes, como manipulação de dados e falta de cooperação. Essa restrição é ainda mais primordial em serviços como a replicação de dados, por suportarem outras aplicações.

Esse trabalho apresenta um sistema de replicação de dados tolerante a ataques maliciosos para suporte às aplicações de gerência de configuração e desempenho das MANETs. Esse esquema usa os conceitos de quóruns probabilísticos [Malkhi et al. 2001] , sendo que sua estratégia de acesso de escrita é baseada no protocolo *gossip* [Leitao et al. 2007], e a sua estratégia de acesso de leitura é baseada nos conceitos dos quóruns f-mascaramento [Malkhi and Reiter 1997]. A avaliação é feita por meio de simulação com o Network Simulator 2 (NS-2). A métrica utilizada para a avalização do novo método é a integridade dos dados (*Data Integrity - DI*), que corresponde ao porcentual de leituras corretas realizadas pelos clientes

O restante desse trabalho está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados; a Seção 3 apresenta o modelo do sistema e a notação utilizada; a Seção 4 descreve o funcionamento do protocolo proposto; a Seção 5 discute os resultados da avaliação; por fim, a Seção 6 contém as conclusões e trabalhos futuros.

2. Trabalhos relacionados

Os sistemas de quóruns foram apresentados por Gifford [Gifford 1979] e Thomas [Thomas 1979], inicialmente como um esquema para votação que garante a consistência dos dados. Em seguida, vários autores estudaram melhorias na construção dos sistemas de quóruns, com objetivo de torná-los mais flexíveis ou confiáveis [Abraham and Malkhi 2005, Bazzi 2000]. Assim, foram propostos os sistemas de quóruns dinâmicos ou reconfiguráveis [Herlihy 1987, Naor and Wieder 2003],

porém uma avaliação conclui que tais esquemas não são aplicáveis às MANETs [Friedman et al. 2008], pois necessitam de muitas trocas de mensagens para a sua reconfiguração. Malkhi et. al [Malkhi et al. 2001] apresentaram os sistemas de quóruns probabilísticos, no qual propõe uma flexibilização nas restrições de intersecção dos quóruns tradicionais. Esses sistemas foram explorados e aplicados em outros paradigmas de redes, como as redes P2P e de sensores [Chockler et al. 2006, Miura and Tagawa 2006].

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.

Contudo esses sistemas aplicados nas MANETs não consideram a presença de nós maliciosos. Em [Mannes et al. 2009] o PAN é avaliado em cenários com ataques de falta de cooperação, temporização e manipulação de dados. Os resultados mostraram que, embora a sua construção resista aos primeiros ataques, ele é vulnerável ao ataque de manipulação de dados. Alguns autores propuseram soluções para a construção de sistemas de quóruns tolerantes a ataques maliciosos [Malkhi and Reiter 1997, Alvisi et al. 2000, Martin and Alvisi 2004], no qual o tamanho das intersecções devem ser grandes o suficiente para mascarar possíveis respostas maliciosas. Essas soluções, porém, não foram aplicadas nas MANETs. Este trabalho apresenta uma solução para a replicação de dados baseada em sistemas de quóruns que seja tolerante a ataques.

3. Notação e modelo do sistema

Esta seção apresenta a notação e as suposições assumidas para a definição do protocolo proposto. Esse protocolo considera uma rede *ad hoc*, formada por um conjunto de n nós móveis, identificados por $n_1, n_2, ..., n_{n-1}, n_n$. Esse conjunto de nós do sistema é representado por N. A Tabela 1 contém a notação utilizada no restante do artigo, e a Seção 3.1 discute algumas ameaças que afetam os serviços nas MANETs.

Assume-se, que todo nó $n_i \in N$ tem um endereço físico ou identificador único. Todas as comunicações entre os nós dependem de algum protocolo de roteamento, como o AODV [Perkins et al. 2003] ou DSR [Johnson et al. 2007]. Da mesma forma, assume-se que tanto as mensagens para suporte ao protocolo como as mensagens contendo os dados a serem armazenados são relativamente pequenas, de no máximo 128 *bytes*. Essa caracteristica é comum nas aplicações de gerência de rede que, em geral, utilizam mensagens pequenas para o controle e monitoramento da rede. Portanto, elas podem ser enviadas em pacote únicos.

Um subconjunto de nós é escolhido para armazenar os dados replicados. Esse conjunto de nós é chamado de sistema de armazenamento, representado por S, e corresponde ao sistema de quóruns. Como em [Luo et al. 2003], os nós-membros do sistema de armazenamento podem ser definidos antes da formação da rede, escolhendo os nós com mais recursos disponíveis, ou podem ser escolhidos dinamicamente com algum algoritmo

Tabela 1. Notações

Notação	Descrição
n_i	identidade de um nó do sistema
N	conjunto de nós do sistema
S	conjunto de servidores de armazenamento
M	conjunto de nós maliciosos
$\mathcal{Q}$	sistema de quóruns
Q_r	quórum de leitura
Q_w	quórum de escrita
$x\|\|y$	concatenação dos valores x e y
$\|N\|$	tamanho de um conjunto N
f	*fanout* do protocolo *gossip*
Δ_t	intervalo das propagações *gossip*
$\Re$	seleção aleatória
C	$\mid C \mid = f \wedge \mid Q_r - 1 \mid$
$dado_m$	dado malicioso

de eleição distribuída. Assume-se que todos os nós da rede sabem quais são os membros do sistema de armazenamento, que são chamados de "servidores". Neste trabalho, é considerado que o servidores podem falhar por *crash* ou apresentar um comportamento malicioso. Dessa forma, a quantidade de servidores falhos no sistema é representada por M.

3.1. Ameaças aos serviços nas MANETs

Diversos tipos de ataques passivos e ativos podem afetar os serviços fornecidos nas MANETs. Um nó malicioso pode, por exemplo, realizar uma escuta não-autorizada, descartar ou injetar pacotes na rede, modificar o conteúdo dos dados transmitidos, atrasar o encaminhamento dos dados ou, até mesmo, personificar outros nós autênticos. Geralmente, esses ataques afetam a confiabilidade e a eficácia das operações da rede [Djenouri et al. 2005].

Neste trabalho, foi considerado o ataque de manipulação de dados, que pode comprometer a integridade e a disponibilidade dos dados armazenados em um sistema de quóruns. Também foi considerado que os nós maliciosos podem comprometer outros nós e realizarem um ataque em conluio, afetando o desempenho e a eficácia dos sistema. Em um ataque de manipulação de dados, os nós maliciosos recebem um dado e o modificam, tanto nas operações de leitura como de escrita do sistema de armazenamento. Como consequência, um cliente pode aceitar como valor correto um dado comprometido, afetando a confiabilidade do sistema.

3.2. Sistemas de quóruns

Um sistema de quóruns típico é definido como um conjunto de subconjuntos de um universo finito $\mathcal{U}$ que possuem duas propriedades distintas: consistência e disponibilidade [Malkhi and Reiter 1997]. A propriedade de consistência garante que quaisquer dois subconjuntos (quóruns) devem ser intersectar - $\forall Q_1, Q_2 \in \mathcal{Q}, Q_1 \cap Q_2 \neq \emptyset$; a propriedade de disponibilidade garante que existe ao menos um subconjunto de nós corretos. Essa segunda propriedade garante a disponibilidade dos dados armazenados em um sistema de quóruns, que implica em uma restrição $|M| < \frac{|N|}{2}$.

Já em um sistema de quórum probabilístico, um quórum não é pré-determinado, mas escolhido de forma probabilística em cada interação [Malkhi et al. 2001]. Nesse caso, o sistema tenta garantir que dois quórum de leitura e de escrita, ambos selecionados aleatoriamente, se intersectem com uma dada probabilidade ε. Assim, dado um sistema de quórum $\mathcal{Q}$ e uma estratégia de acesso w, então $\forall Q_1, Q_2 \in \mathcal{Q}, \mathcal{P}(Q_1 \cap Q_2 \neq \emptyset) \geq 1 - \varepsilon$. A estratégia de acesso define a forma com que os quóruns de leitura e de escrita são selecionados.

Em [Malkhi et al. 1999], Malkhi et al. relatam que se a estratégia de escrita de um sistema de quóruns probabilístico estiver associada com um protocolo de disseminação de mensagens eficaz, a probabilidade de inconsistência nas leituras pode ser próxima a zero. Isso motivou o uso dos protocolos *gossip* [Leitao et al. 2007] para a disseminação das informações no sistema de armazenamento.

3.3. Protocolo Gossip

Os protocolos *gossip* têm sido amplamente utilizados para a disseminação de mensagens em uma rede. Nesses protocolos, quando um nó deseja disseminar uma mensagem, ele seleciona aleatoriamente f outros nós (chamados de "parceiros *gossip*"), e envia a mensagem para eles. Nesse caso, f é um parâmetro de configuração chamado de *fanout*. Ao receber uma mensagem pela primeira vez, um nó repete o mesmo processo, selecionado f nós e encaminhando a mensagem para eles. Como os nós são selecionados aleatoriamente, é possível que um nó receba a mesma mensagem várias vezes; nesse caso, ele simplesmente descarta as outras mensagens. Por isso, cada nó deve manter uma lista das mensagens que ele já recebeu e reencaminhou [Leitao et al. 2007]. Assim como a propagação de boatos na vida real, as informações disseminadas por um protocolo *gossip* se difundem rapidamente e com confiabilidade [Eugster et al. 2004]. Esses protocolos são altamente escaláveis e tolerante a falhas, mascarando falhas do tipo *crash* e de omissão.

As informações podem ser difundidas, tradicionalmente, seguindo uma das seguintes estratégias [Eugster et al. 2004]:

- Abordagem *eager push*: os nós enviam uma informação para outros nós selecionados aleatoriamente assim que a recebem pela primeira vez;
- Abordagem *pull*: os nós, periodicamente, consultam outros nós selecionados aleatoriamente, perguntando por novas informações recebidas;
- Abordagem *lazy push*: os nós que recebem uma nova informação enviam um identificador para um conjunto de nós selecionados aletoriamente. Se esses nós não possuem tal informação, eles a solicitam explicitamente;
- Abordagens híbridas: as abordagens anteriores podem ser combinadas, com o objetivo de aproveitar as vantagens de cada uma delas.

4. O protocolo proposto

Esta seção descreve o funcionamento do sistema de armazenamento e replicação de dados proposto. Inicialmente, é apresentada uma visão geral do sistema e as suas características. Em seguida, são descritas as operações de leitura e de escrita realizadas pelos clientes e servidores.

O sistema possui dois componentes: o cliente e o servidor. Sempre que um dado nó N_x deseja realizar uma operação de leitura ou escrita no sistema, ele é considerado um

"cliente". Dessa forma, todos os nós da rede podem agir como clientes, mesmo aqueles que fazem parte do sistema de armazenamento. Tanto nas leituras como nas escritas, os **clientes** enviam pedidos para um conjunto de **servidores**. Esses servidores são escolhidos aleatoriamente pelos clientes. No lado servidor, dois tipos de quóruns são considerados: de escrita (Q_w) e de leitura (Q_r). Um quórum de escrita é acessado por uma operação de atualização ou escrita, enquanto que um quórum de leitura é acessado por uma operação de consulta ou leitura.

A figura 1 ilustra a arquitetura do sistema composta por 20 nós. Desses nós, 10 nós formam o sistema de armazenamento, que corresponde ao sistema de quórum $\mathcal{Q} = \cup_{i=0}^{9} n_i$. Além disso, todos os nós podem agir como clientes do sistema. No exemplo apresentado, o nó n_{17} está realizando uma requisição, de leitura ou escrita, aos servidores do sistema de armazenamento.

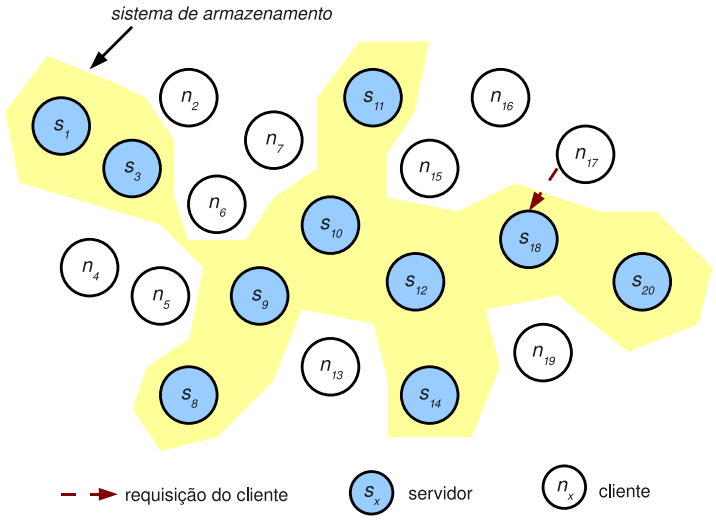

Figura 1. Sistema de armazenamento

4.1. Escrita

Quando um dado cliente n_x deseja realizar a escrita de um dado $dado_{n_x}$ no sistema de armazenamento, ele cria uma mensagem $msg = [\ n_x\ ||\ dado_{n_x}\ ||\ timestamp(dado_{n_x})\ ]$, em que $timestamp(dado_{n_x})$ é a versão do dado $dado_{n_x}$. Em seguida, ele seleciona f servidores, e envia a mensagem msg para eles. O valor f corresponde à quantidade inicial de servidores que receberão o dado atualizado. Esses servidores serão responsáveis pela difusão dos dados para os demais servidores do sistema.

O valor de f assumido pelos clientes pode ser diferente do valor assumido pelos servidores. Quanto maior for f na escrita do cliente, mais rápido os dados serão propagados. Além disso, em cenários com ataques, um maior valor de f também aumenta a probabilidade de servidores corretos participarem da fase inicial da escrita. Contudo, quanto maior o valor f no cliente, maior é o custo de comunicação necessário para realizar o armazenamento de um dado. Na avaliação do sistema, sempre é considerado o valor f igual a 2.

Ao receber uma requisição de escrita de um dado $dado_{n_x}$, os servidores o adicionam em um *buffer*. Periodicamente, esses servidores escolhem f servidores do sistema de quóruns e propagam para ele o conteúdo do *buffer*. Contudo, um dado $dado_{n_x}$ apenas é considerado correto por um servidor n_y, e armazenado em sua base local, se ele receber pelo menos f dados iguais, referente à mesma escrita. Caso um servidor n_y receba duas informações conflitantes, supostamente originadas de um mesmo cliente n_x, ele detecta uma ação maliciosa no sistema. Assim, ele pode decidir por perguntar explicitamente ao cliente emissor da escrita qual dos dados é correto, ou aguardar até receber uma segunda mensagem resolvendo o conflito de escrita.

O Algoritmo 1 resume o funcionamento do protocolo de escrita, em que o cliente n_x está realizando uma escrita no sistema de armanamento.

Procedimento 1 ESCRITA

nó cliente n_x ***enviando uma escrita***

1: $dest \leftarrow C \subset_{\Re} StS : |C| = |f|$ {*escolhe os nós para disseminar o dado*}
2: **for** $s_w \in dest$ **do**
3: $envia(atualiza(dado, valor, timestamp))$ {*envia a atualização*}
4: **end for**

nó servidor s_x ***ao receber uma escrita diretamente do cliente***

1: **if** $s_x \in M$ **then**
2: $dado_{recebido} \leftarrow dado_m$ {*modifica o dado recebido*}
3: **else**
4: $s_{x(confirmado)} \leftarrow 1$ {*dado é confirmado como confiável*}
5: **end if**
6: $dado_{local} \leftarrow dado_{recebido}$ {*atualiza o dado local*}
7: $buffer \leftarrow dado_{recebido}$ {*adiciona dado malicioso no buffer*}

nó servidor s_x ***ao receber uma escrita por meio do gossip***

1: **if** $s_{x(confirmado)} = 0$ **then**
2: $escritasRecebidas \leftarrow dado_{recebido}$ {*adiciona na lista de escritas recebidas*}
3: **end if**
4: **if** $escritas_{recebidas} \subset \ f\ resp_{iguais}$ **then**
5: $dado_{local} \leftarrow respostas_{iguais}$ {*atualiza o dado local*}
6: $s_{x(confirmado)} \leftarrow 1$ {*dado é confirmado como confiável*}
7: **else**
8: $consultaCliente(n_x)$ {*consulta o cliente*}
9: **end if**
10: $buffer \leftarrow dado_{recebido}$ {*adiciona o dado recebido no buffer*}

nó servidor s_x ***a cada*** Δ_t

1: **while** $buffer \neq \emptyset$ **do**
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: **for all** $s_w \in dest$ **do**
6: $envia(entrada, s_w)$ {*envia a atualização*}
7: **end for**
8: **end while**

4.2. Leitura

Quando um dado cliente n_x deseja ler um dado qualquer no sistema, ele deve definir um quórum de leitura (Q_r) e solicitar aos servidores desse quórum o dado desejado. O tamanho do quórum de leitura pode ser variável, dependendo das características do ambiente. Vários tamanhos de quóruns de leitura são considerados nas avaliações.

Em uma requisição de leitura, os servidores simplesmente respondem ao cliente o seu dado mais atualizado. O cliente ao receber as respostas dos servidores, deve compará-las, a fim de descartar as mensagens que possivelmente estejam desatualizadas ou sejam maliciosas. Para isso, o cliente guarda uma lista com as respostas recebidas, e ao receber todas as respostas, ou o *timeout* das leituras expirar, o cliente compara as leituras recebidas, e considera como correta aquela que aparecer mais vezes.

O Algoritmo 2 resume o funcionamento do protocolo de leitura, em que o cliente n_x está realizando uma leitura no sistema de armanamento.

Procedimento 2 LEITURA

nó cliente n_x ***requisitando uma leitura***

1: $dest \leftarrow C \subset_{\Re} StS : |C| = |Q_r|$ {*escolhe os nós do quórum de leitura*}
2: **for** $s_w \in dest$ **do**
3: $envia(ler_{dado})$ {*envia a requisição de leitura*}
4: $timer_{s_w} \leftarrow 0$ {*inicia um timer para as requisições de leitura*}
5: **end for**

nó cliente n_x ***ao receber uma resposta***

1: $contador \leftarrow contador + 1$
2: *msgRecebidas* $\leftarrow$ *msgRecebidas* $+ dado_{recebido}$ {*insere dado na lista de respostas*}
3: **if** $contador = Q_r$ **then**
4: $Commit(msgRecebidas)$
5: **end if**

5. Avaliação

Esta seção apresenta a avaliação do protocolo proposto diante de ataques de manipulação de dados. A Seção 5.1 apresenta as métricas e os cenários utilizados na avaliação e a Seção 5.2 discute os resultados das simulações.

5.1. Métricas e cenários

A métrica DI, Integridade dos Dados, foi definida para a avaliação do sistema diante de ataques de manipulação de dados. Ela corresponde ao porcentual de leituras corretas realizadas pelos clientes. Uma leitura é considerada correta se ela não foi modificada por um nó maliciosos, durante as fases de escrita e leitura dos dados. A métrica é definida como segue:

- Integridade dos dados (DI - *Data Integrity*): quantifica a probababilidade dos quóruns de escrita e de leitura se intersectarem, considerando apenas a quantidade de leituras corretamente obtidas pelo cliente. Sendo R o conjunto de todas as leituras realizadas, RD pode ser definido como:

$$DI = \sum_{i \in R} \frac{DI_i}{|R|} \quad \text{em que,} \tag{1}$$

$$DI_i = \begin{cases} 1 & \text{se } Q_w \setminus M \cap Q_r \setminus M \neq \emptyset \\ 0 & \text{caso contrário} \end{cases} \tag{2}$$

O protocolo proposto 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 sistema de armazenamento. 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. Esses parâmetros são os mesmos utilizados em [Luo et al. 2003]. As atualizações contidas no *buffer* são propagadas a cada 200ms, e o *fanout* é igual a 2, tanto nos clientes como nos servidores. O tamanho do quórum de leitura varia entre 4 e 7 servidores. A quantidade de atacantes ($|M|$) é de 20%, 28% e 36% dos nós do sistema de armazenamento, o que corresponde a 5, 7 e 9 nós atacantes, respectivamente. Esses mesmos valores foram utilizados em [Mannes et al. 2009].

Os pacotes com as requisições têm tamanho de 128 *bytes*, o suficiente para o tipo de dados que deseja-se replicar. 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.

5.2. Resultado das simulações

Essa seção apresenta os resultados obtidos com o esquema proposto, comparados aos resultados do PAN [Luo et al. 2003], analisado em [Mannes et al. 2009]. A Figura 2 ilustra a integridade dos dados no PAN, quando sob ataque de manipulação de dados.

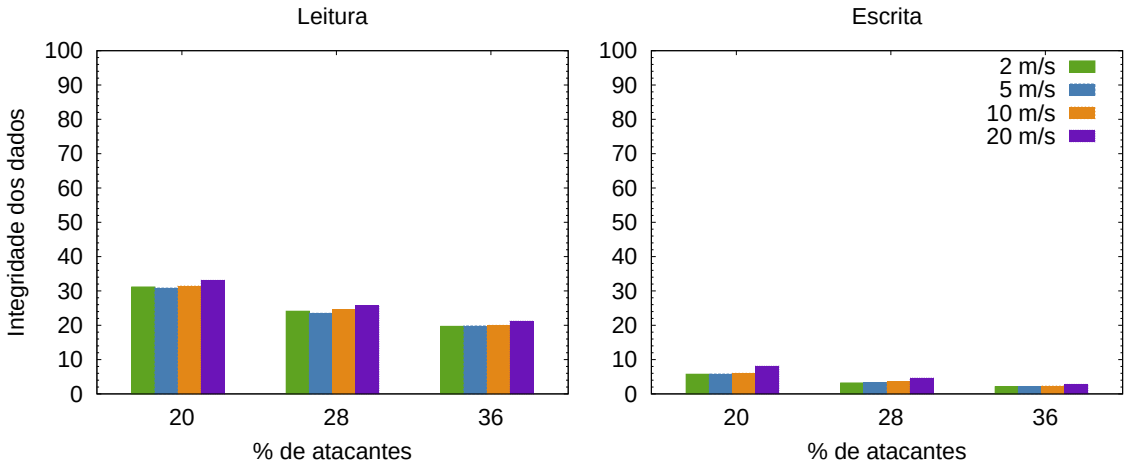

Figura 2. Integridade dos dados nas operações de leitura e escrita no PAN

Diante do ataque de manipulação de dados nas operações de leitura, o PAN atinge uma média de 25% de leituras recebidas que não foram comprometidas pelos nós maliciosos, sendo que o pior resultado obtido é na velocidade de 2m/s, com 36% de nós atacantes

na rede, que foi de 19,67%. O melhor resultado obtido é com 5% de nós atacantes, na velocidade de 20m/s, com 33,08% de leituras concluídas que não foram afetadas pelos nos maliciosos.

Já diante de ataques nas escritas, a integridade dos dados é muito mais comprometida, pelo motivo de que nas escritas, os nós maliciosos além de modificarem a escrita, propagam aos outros nós essas escritas maliciosas. Nesse caso, a integridade dos dados na escrita tem uma média de 3,5% leituras concluídas que não foram modificadas pelos nós maliciosos. O pior caso é com 9 atacantes, na velocidade de 2m/s, em que a integridade dos dados é de 2,08%, e o melhor caso é de 20m/s, com 5 atacantes, que foi de 8,05%.

A Figura 3 apresenta os resultados do novo esquema, que sugere a comparação dos dados recebidos em uma leitura pelo cliente, tem uma melhora considerável na quantidade de dados recebidos que não foram modificados por servidores maliciosos. Nesse caso, a integridade dos dados diante de ataques, que antes era em média 25%, com o esquema proposto, sobe para aproximadamente 70%, uma melhora de 50% na confiabilidade dos dados recebidos. A quantidade de dados que o cliente recebe como resposta a uma leitura influencia a qualidade da leitura considerada. Para isso, foram considerados diversos tamanhos de quóruns de leitura.

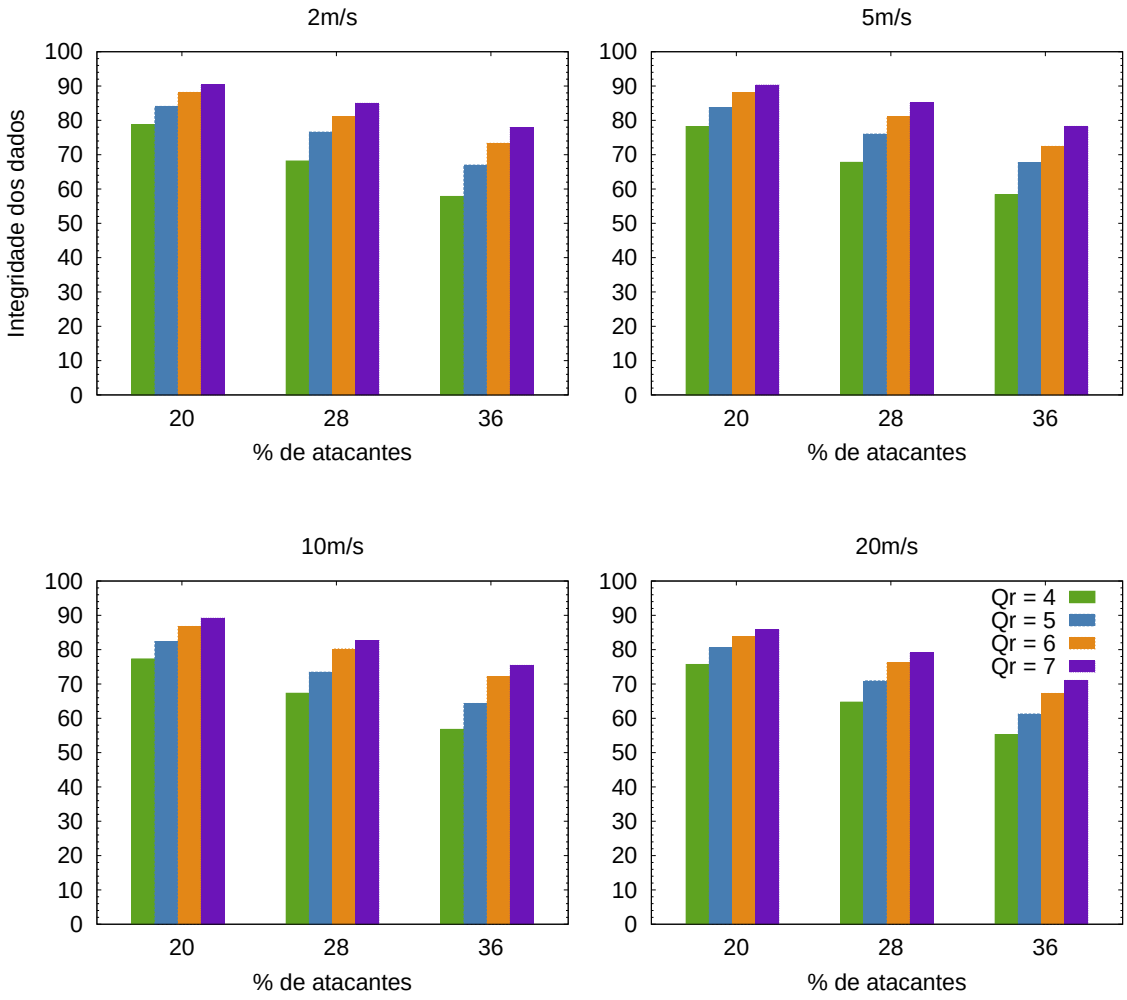

Figura 3. Integridade dos dados na operação de leitura

Percebe-se que quanto maior o quórum de leitura, melhor é a integridade dos dados recebidos. Isso porque o cliente possui mais resultados para comparar, e a chance de escolher um dado malicioso como resposta é menor. Também a probabilidade de receber dois dados iguais aumenta, o que aumenta a confiabilidade nesse dado, já que dois nós possuiam o mesmo valor.

Já diante de ataques na escrita, como apresentado na Figura 4, a integridade dos dados passa de 3,5% para 45% em média, um aumento de 41,5% na integridade dos dados. No melhor caso, o novo esquema atinge 59,7% de leituras que não foram manipuladas por nós maliciosos. Esse resultado corresponde ao cenário com velocidade de 5m/s e 20% de nós maliciosos no sistema. O pior caso é com velocidade de 5m/s e 36% de nós atacantes na rede, que corresponde a 30,2% de leituras não comprometidas por nós atacantes.

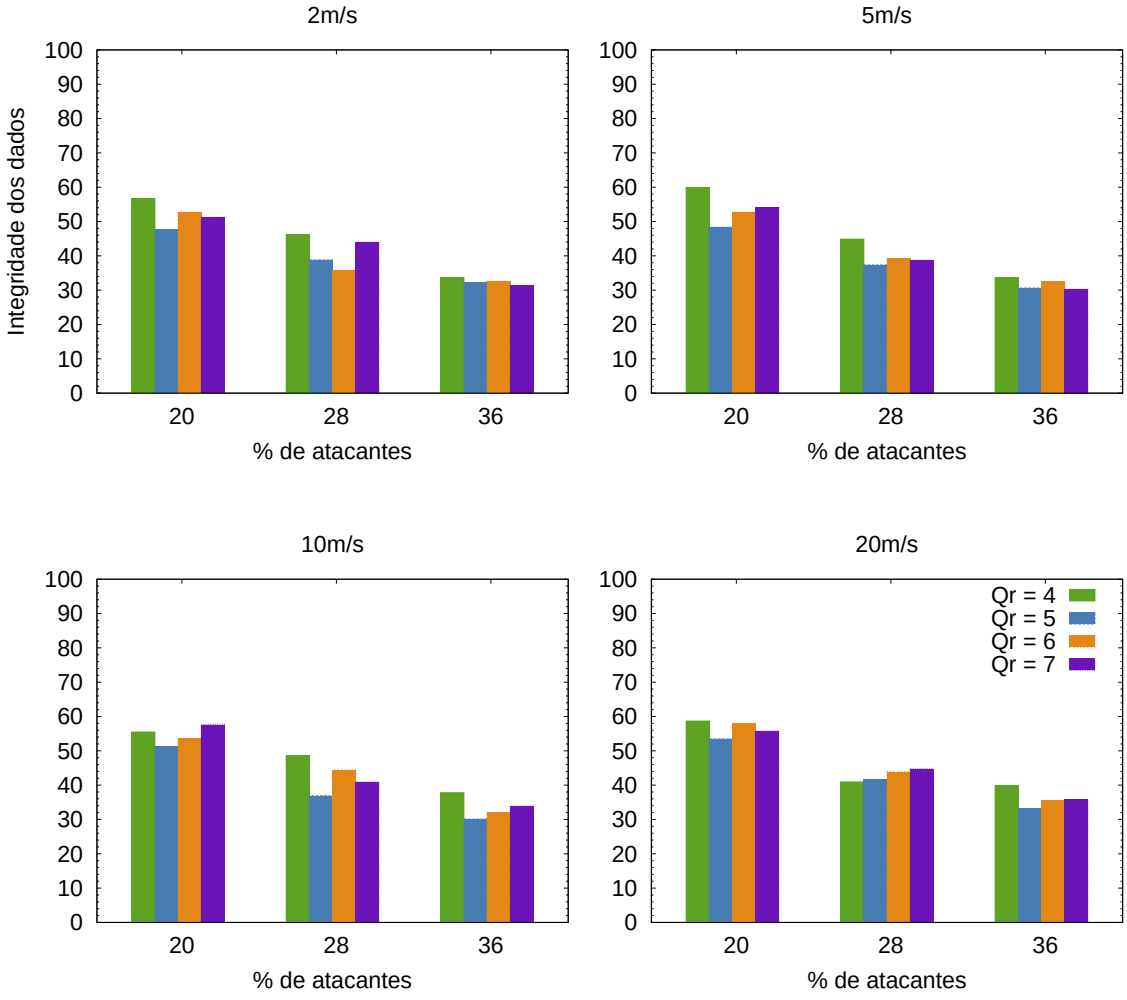

Figura 4. Integridade dos dados na operação de escrita

A escrita possui um impacto maior no esquema porque os dados maliciosos, assim como os corretos, são propagados rapidamente para outros nós na rede. Desta forma, os servidores precisam comparar os dados recebidos, e no caso de dados divergentes, resolver o conflito diretamente com o cliente. No caso da escrita, também foi simulado vários tamamnhos de quóruns de leitura, e os resultados mostram que em geral, um quórum menor garante uma melhor confiabilidade no sistema. Isso ocorre porque um menor quórum

faz com que a probabilidade de receber muitas leituras incorretas seja menor, ja que o número de respostas é limitado.

6. Conclusão e Trabalhos Futuros

Este trabalho propôs um sistema para a replicação de dados em MANETs tolerante a ataques de manipulação de dados. Para isso, utilizou-se os conceitos dos sistemas de quóruns probabilísticos, em que a estratégia de acesso aos quóruns deve garantir que a probabilidade de intersecção entre os quóruns não seja nula. Para isso, foi proposta uma estratégia de leitura baseada no f-mascaramento, e uma estratégia de escrita em que os servidores gravam os dados somente após confirmá-los. O novo esquema foi quantificado por meio de simulações, que consideraram a integridade dos dados nos resultados.

Os resultados obtidos por meio das simulações mostram que o esquema proposto garante uma maior integridade dos dados no sistema. Nas leituras, o esquema proposto aumenta a confiabilidade nos dados em 50%, garantindo em média 70% de leituras não comprometidas. Já na escrita, o esquema proposto aumenta aproximadamente 45% a integridade dos dados, um aumento de 41% comparado a confiabilidade do PAN. Comparou-se também o impacto de vários tamanhos de quórum de leitura nas soluções de leitura e de escrita do novo esquema proposto. Os resultados mostram que enquando na leitura é melhor um quórum de tamanho maior, na escrita, é recomendável o uso de um quórum pequeno, que possui uma melhor resiliência aos dados manipulados.

Como trabalhos futuros pretende-se estudar outras formas de realizar a escrita, para que a confiabilidade na escrita seja melhorada. Também pretende-se verificar a eficácia de mecanismos de reputação na rede, para que os nós maliciosos sejam conhecidos e punidos pelo comportamento malicioso, livrando a rede do impacto desses nós nas operações do sistema de quóruns.

Referências

Abraham, I. and Malkhi, D. (2005). Probabilistic quorums for dynamic systems. *Distributed Computing*, 18(2):113–124.

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.

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

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

Chockler, G., Gilbert, S., and Patt-Shamir, B. (2006). Communication-efficient probabilistic quorum systems for sensor networks. In *Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops (PERCOMW '06)*, page 111, Washington, DC, USA. IEEE Computer Society.

Djenouri, D., Khelladi, L., and Badache, N. (2005). A survey of security issues in mobile ad hoc and sensor networks. *IEEE Surveys and Tutorials*, 7(4):2–28.

Eugster, P. T., Guerraoui, R., Kermarrec, A.-M., and Massoulieacute, L. (2004). Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67.

Friedman, R., Kliot, G., and Avin, C. (2008). Probabilistic quorum systems in wireless ad hoc networks. In *Proceedings of the 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '08)*, pages 277–286, Washington, DC, USA. IEEE Computer Society.

Gifford, D. K. (1979). Weighted voting for replicated data. In *Proceedings of the 7th ACM Symposium on Operating Systems Principles (SOSP '79)*, pages 150–162, New York, NY, USA. ACM Press.

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

Herlihy, M. (1987). Dynamic quorum adjustment for partitioned data. *ACM Transactions on Database Systems (TODS)*, 12(2):170–194.

Johnson, D. B., Maltz, D. A., and Hu, Y. C. (2007). RFC 4728 - The Dynamic Source Routing protocol for mobile ad hoc networks (DSR).

Leitao, J., Pereira, J., and Rodrigues, L. (2007). Epidemic broadcast trees. In *Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems (SRDS '07)*, pages 301–310, Washington, DC, USA. IEEE Computer Society.

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., Mansour, Y., and Reiter, M. K. (1999). On diffusing updates in a byzantine environment. In *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS '99)*, page 134, Washington, DC, USA. IEEE Computer Society.

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, New York, NY, USA. ACM Press.

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

Mannes, E., da Silva, E., and dos Santos, A. L. (2009). Analisando o desempenho de um sistema de quóruns probabilístico para manets diante de ataques maliciosos. In *Anais do IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg '09)*, pages 71–84, Campinas. SBC.

Martin, J.-P. and Alvisi, L. (2004). A framework for dynamic byzantine storage. In *Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN '04)*, page 325, Washington, DC, USA. IEEE Computer Society.

Miura, K. and Tagawa, T. (2006). A quorum-based protocol for searching objects in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(1):25–37.

Naor, M. and Wieder, U. (2003). Scalable and dynamic quorum systems. In *Proceedings of the 22th Annual Symposium on Principles of Distributed Computing (PODC '03)*, pages 114–122, New York, NY, USA. ACM.

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

Perkins, C., Royer, E., and Das, S. (2003). RFC 3561 - Ad hoc On-demand Distance Vector (AODV) routing.

Thomas, R. H. (1979). A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems (TODS)*, 4(2):180–209.

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

Wu, B., Chen, J., Wu, J., and Cardei, M. (2006). *A survey on attacks and countermeasures in mobile ad hoc networks*, chapter 12, pages 103–136. Springer-Verlag, New York, NY, USA.

Zhang, C., Song, Y., and Fang, Y. (2008). Modeling secure connectivity of self-organized wireless ad hoc networks. In *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, Los Alamitos, CA, USA. IEEE Communications Society.