

Consenso FT-CUP em Redes MANETs: Uma Abordagem Prática *

Victor Franco Costa¹, Fabíola Gonçalves Pereira Greve¹, Sébastien Tixeuil²

¹Programa de Pós-Graduação em Mecatrônica – Universidade Federal da Bahia (UFBA)
CEP 40.170-110 – Salvador – BA – Brasil

²Université Paris VI, LIP6 & INRIA Grand Large, France

{vfcosta, fabiola}@dcc.ufba.br, tixeuil@lip6.fr

Resumo. *O FT-CUP é uma solução para o consenso tolerante a falhas em sistemas dinâmicos que, ao contrário do consenso clássico, não tem como requisito o conhecimento prévio dos participantes. Este trabalho complementa os resultados teóricos alcançados até então para a resolução do FT-CUP e analisa os aspectos práticos da sua realização em redes MANETs. São propostas implementações para os algoritmos sugeridos e realizadas simulações em cenários realistas. A partir dos resultados obtidos, chegou-se a um conjunto de parâmetros para os quais é possível a convergência do FT-CUP, definindo-se então as características de uma rede MANET onde o consenso pode ser resolvido de maneira aproximada.*

Abstract. *FT-CUP is a fault tolerant consensus for a network of unknown participants. It has been defined for taking into account the dynamic nature of self-organizing networks. This paper complements the theoretical results achieved until now for solving FT-CUP and analyzes the practical aspects of implementing it in a mobile ad-hoc network (MANET). The simulations experiments conducted show that the conditions for solving FT-CUP can actually be approximated in a MANET environment. Additionally, a set of parameters where FT-CUP converges has been identified.*

1. Introdução

O modelo de redes móveis *ad-hoc* (MANETs) possui diferenças essenciais em relação ao modelo clássico de sistemas distribuídos, e traz novos desafios para a especificação dos seus protocolos fundamentais, como o consenso. Redes MANETs são redes constituídas por nós móveis, cuja comunicação dá-se através de canais sem fio (*wireless*). Para que um nó possa se comunicar diretamente com outro, esse deve estar localizado no raio de transmissão do dispositivo de comunicação [Basile et al. 2003]. A mobilidade, a falta de estrutura da rede e as entradas e saídas deliberadas, fazem com que a dinamicidade da rede seja alta, tornando o conhecimento inicial dos participantes uma condição muito forte para ser adotada em MANETs. Por isso, no início da execução em uma MANET, considera-se que um processo não conhece o conjunto de participantes que compõem a rede, assim como desconhece a quantidade destes.

*Este trabalho tem o apoio do CNPQ e FAPESB-Bahia.

O consenso é um protocolo fundamental em sistemas distribuídos e diversos problemas são redutíveis ou equivalentes a ele. Informalmente, o consenso tem o objetivo de fazer com que todos os processos corretos do sistema decidam por um valor único proposto pelos mesmos. Em soluções para o consenso clássico, tais como [Chandra and Toueg 1996, Ben-Or 1983, Guerraoui 2004], o conhecimento inicial dos participantes é essencial, logo tais soluções não são adequadas ao modelo de redes MANETs.

Recentemente, o problema do consenso com participantes desconhecidos na rede foi proposto e estudado. Este novo problema foi designado CUP (*consensus with unknown participants*) e a sua extensão tolerante a falhas, FT-CUP (*fault tolerant consensus with unknown participants*) [Cavin et al. 2004, Cavin et al. 2005]. A solução para esse consenso contempla a definição de uma nova abstração, os *detectores de participação*. Estes são considerados oráculos distribuídos, associados a cada processo, que retornam um conhecimento parcial dos participantes existentes na rede. Algumas classes para estes detectores foram propostas. Cada classe estabelece um grafo de conectividade entre os participantes do sistema, definido a partir da relação de conhecimento estabelecida pelo detector. Foi mostrado em [Cavin et al. 2004] que o detector de participação minimal para resolver o CUP é o de classe *OSR*.

Em [Fischer et al. 1985], foi provado que o consenso clássico não pode ser resolvido em ambiente assíncrono e na presença de falhas. Por isso, é fundamental estabelecer requisitos de sincronia para a resolução do FT-CUP. Na solução apresentada em [Cavin et al. 2005], propõe-se uma solução para o FT-CUP utilizando um detector de falhas perfeito (classe \mathcal{P}) e o detector de participação *OSR*. Esta classe de detector só pode ser implementada num ambiente síncrono [Larrea et al. 2004] e não se adequa a redes dinâmicas.

Greve e Tixeuil apresentam uma proposta alternativa para solucionar o FT-CUP [Greve and Tixeuil 2007]. Nesta, os autores consideram os requisitos mínimos de sincronia, já identificados para a resolução do consenso no modelo clássico, e buscam os requisitos mínimos para a conectividade do grafo de conhecimento que possibilitam resolver o problema. Como demonstrado em [Chandra and Toueg 1996], o requisito mínimo de sincronia resume-se ao detector de falhas da classe $\diamond S$. Esses detectores possuem propriedades não-confiáveis e podem ser implementados em sistemas dinâmicos, como MANETs, redes de sensores e P2P (*peer-to-peer*).

Este trabalho estuda o consenso FT-CUP proposto por Greve e Tixeuil através de uma abordagem prática. Nosso intuito é o de avaliar o comportamento do protocolo numa rede MANET de topologia arbitrária, onde as condições teóricas de conectividade de conhecimento exigidas pelo FT-CUP não são necessariamente satisfeitas. Para tanto foi considerado uma implementação simples de detector de participação, que origina grafos de conhecimento arbitrários e que, não necessariamente, satisfazem as propriedades estabelecidas para a resolução do consenso. Desta forma, foi possível avaliar o grau de convergência do consenso FT-CUP e a sua conseqüente latência, mesmo em situações adversas. Tal avaliação do comportamento do FT-CUP deu-se a partir de experimentos de simulação, variando-se tanto os parâmetros da MANET, quanto os do FT-CUP.

O resto deste artigo está organizado da seguinte maneira. A seção 2 introduz o problema do consenso clássico e o consenso para redes desconhecidas; a seção 3 apresenta o modelo de implementação do FT-CUP utilizado neste trabalho; a seção 4 apresenta os

resultados e a análise de desempenho; a seção 5 apresenta as conclusões.

2. O Problema do Consenso

Modelo do Sistema. Considera-se um sistema distribuído formado por um conjunto finito Π de $n > 1$ processos, que se comunicam através de canais confiáveis (sem perdas de mensagens) e de forma assíncrona (sem que sejam feitas hipóteses temporais sobre as ações dos processos e canais). Considera-se que $f < n$ processos podem falhar por parada (*crash*), através de um colapso brusco ou saída deliberada (*switched off*). O valor n é desconhecido pelos processos, já o valor f é do conhecimento de todos. Considera-se ainda a existência de um protocolo de roteamento confiável tal que, se o processo p conhece o processo q , então p é capaz de enviar mensagens para q de forma confiável.

2.1. Consenso Clássico

A definição formal do consenso clássico consiste nas seguintes propriedades [Chandra and Toueg 1996]:

- **Terminação.** Todo processo correto decide por um valor;
- **Validade.** Se um processo decide por um valor, ele foi proposto por algum processo;
- **Acordo.** Todos os processos corretos decidem pelo mesmo valor.

As primeiras soluções para o consenso em ambientes assíncronos ofereciam garantias probabilísticas. Estas soluções usavam um oráculo aleatório para gerar as estimativas propostas pelos processos [Ben-Or 1983]. O oráculo aleatório possui uma função RANDOM que retorna aleatoriamente um valor 0 ou 1. Posteriormente, foi proposta a abstração de detectores de falhas [Chandra and Toueg 1996]. Tais detectores reúnem as condições de sincronia necessárias e suficientes para a resolução determinística do consenso em ambientes assíncronos com participantes conhecidos.

2.2. Consenso em Redes Desconhecidas

Apenas poucos trabalhos abordam o consenso em redes com participantes desconhecidos [Cavin et al. 2004, Cavin et al. 2005, Greve and Tixeuil 2007]. Em MANETs, abordar o consenso com o conhecimento prévio dos participantes não é realista devido a alta dinamicidade da rede. Entretanto, é necessário algum conhecimento sobre os participantes para que se possa executar qualquer algoritmo distribuído não trivial. Por isto, o conceito de detectores de participação (PD) foi proposto para resolver este problema em redes com características auto-organizáveis.

Detectores de Participação. Os detectores de participação (PD) são oráculos distribuídos que fornecem informações sobre os participantes do sistema. Seja PD_p o detector de participação associado ao processo p . Quando consultado por p , PD_p retorna um subconjunto de processos de Π com os quais ele pode colaborar. Estes detectores devem satisfazer as propriedades: (i) *inclusão da informação*: a informação retornada por PD é não decrescente ao longo do tempo; (ii) *exatidão da informação*: PD não comete erros.

A informação retornada por todos os PDs associados aos processos enriquece o sistema com um grafo de conectividade por conhecimento. Este grafo é orientado, pois a relação de conhecimento não é necessariamente bidirecional. Assim, se $q \in PD_p$, então não necessariamente $p \in PD_q$. A partir das características do grafo de conhecimento, algumas classes de detectores de participação foram propostas por [Cavin et al. 2004] para resolução do CUP. Para resolver o FT-CUP, Greve e Tixeuil introduzem novas classes de detectores de participação. A mais importante para este trabalho é:

k-Redutível a Único Poço (k-OSR). O grafo orientado G_{di} satisfaz as seguintes condições:

1. O grafo G , obtido de G_{di} , é conexo;
2. O grafo direcionado acíclico, obtido pela redução de G_{di} às componentes k -fortemente conexas, tem apenas uma componente poço (um poço é um nó com grau de saída 0);
3. Para $k > 1$, considere quaisquer duas componentes k -fortemente conexas G_1 e G_2 , se existe um caminho entre G_1 e G_2 , então existem k caminhos disjuntos entre G_1 e G_2 .

Uma componente G_c de um grafo G_{di} é k -fortemente conexa se, para qualquer par (p, q) de nós em G_c , existem k caminhos disjuntos entre p e q . Uma componente G_s do grafo G_{di} é considerada componente poço quando não existem caminhos partindo de nós em G_s para qualquer outro nó de G_{di} . Sendo assim, um grafo k -OSR formado por apenas uma componente corresponde a um grafo do tipo k -SCO.

2.3. FT-CUP com Requisitos Mínimos de Sincronia

De acordo com o grau de conectividade do grafo estabelecido pelo detector, Cavin *et al.* [Cavin et al. 2004] provou que os requisitos de conectividade necessários e suficientes para resolver o CUP (*Consensus with Unknown Participants*) em ambiente assíncrono e em cenários sem falhas resume-se ao detector de participação da classe OSR (equivalente ao 1-OSR). Em seguida, os autores mostram que este detector não é suficiente para resolver o FT-CUP (*Fault-Tolerant CUP*). Mostram que para resolver o FT-CUP com o detector OSR são necessárias fortes condições de sincronia, representadas pelo detector de falhas perfeito (\mathcal{P}). Entretanto, estas condições de sincronia confrontam com as características naturais das redes MANETs: a alta dinamicidade, descentralizada e auto-organizável. Além disso, mesmo com estas condições, o consenso uniforme não pode ser solucionado.

Greve e Tixeuil [Greve and Tixeuil 2007] utilizaram os requisitos mínimos de sincronia (representado pelo detector de falhas $\diamond\mathcal{S}$) e, a partir destes requisitos, identificaram os requisitos mínimos de conectividade para resolver o problema. Mostrou-se que, se a quantidade de falhas f é menor do que k ($f < k < n$), o detector k -OSR é suficiente e necessário para resolver o FT-CUP. Para tanto, propõem-se três algoritmos: COLLECT, SINK, CONSENSUS. Cada um deles é executado em sequência, individualmente por cada processo do sistema.

COLLECT. É usado pelo processo p para expandir o seu conhecimento sobre os participantes da computação, a partir de uma busca no grafo. No início do algoritmo, o processo p consulta o seu detector de participação para obter PD_p . Em seguida, num procedimento de descoberta do grafo, p irá requisitar novas informações de conhecimento (visões) a novos processos identificados, até que nenhuma nova informação possa ser obtida. Ao término do algoritmo, ele terá armazenado em Π_p o conjunto maximal de participantes que ele pode alcançar no grafo de conhecimento.

SINK. Através deste algoritmo, cada nó pode saber se pertence à componente poço (definida pelo grafo k -OSR). Nesse algoritmo, cada nó p envia mensagens para todos os nós em Π_p (encontrados pelo COLLECT), visando comparar o seu conjunto com o de cada processo em Π_p . Seja $q \in \Pi_p$ um desses processos. Se q possui o conjunto $\Pi_q = \Pi_p$, então q responde ACK à solicitação de p , senão ele responde NACK. Se $|\Pi_p| - f$ nós

respondem com ACKs, o nó p deduz que ele pertence à componente poço, já que apenas na componente poço os nós possuem a mesma visão do conjunto de processos no sistema. Se o nó p recebe algum NACK, ele deduz que não pertence à componente poço. Note que se o grafo for k -SCO, todos os nós devem ser considerados pertencentes à componente poço, pois esta componente coincide com o próprio grafo do sistema.

CONSENSUS. Este último algoritmo a ser executado é responsável pela realização do acordo. Inicialmente, cada nó executa o SINK para obter uma visão parcial do sistema e decidir se pertence à componente poço. Ele admite dois comportamentos para os nós. Apenas os nós identificados como participantes da componente poço executam a fase agreement para obter o consenso. Sendo assim, qualquer consenso subjacente, baseado em algum oráculo (*detector de falhas*, *eleição de líder* ou *aleatório*), é suficiente para resolver o consenso, desde que exista uma maioria de processos corretos na componente poço [Chandra et al. 1996]. Precisamente, a componente poço deve possuir pelo menos $2f + 1$ nós. Os outros nós (pertencentes as componentes k -fortemente conectas) não participam do consenso. Eles apenas enviam uma mensagem para os processos conhecidos requisitando o valor do consenso e aguardam até que a componente poço obtenha o acordo.

3. Implementação do FT-CUP

Nesta seção será apresentado o modelo de implementação do FT-CUP proposto neste trabalho e os cenários utilizados para analisar o comportamento do protocolo no ambiente de redes MANETs. O experimento foi realizado através do OMNeT++, um simulador orientado a eventos discretos, em conjunto o *Mobility Framework*, uma extensão que implementa o modelo das redes MANETs.

3.1. Modelagem dos Algoritmos

Uma visão geral do sistema encontra-se ilustrada na Figura 1(a), onde pode-se identificar os algoritmos do FT-CUP descritos na seção 2. Os algoritmos foram implementados seguindo uma estrutura modular, onde os módulos se comunicam segundo uma estrutura pré-definida. O FT-CUP contém os seguintes módulos: PD, COLLECT, SINK e CONSENSUS. Cada um representa, respectivamente, os algoritmos do FT-CUP *detector de participação*, *collect*, *sink* e *consensus*. Nas camadas inferiores ao FT-CUP, foram utilizados os módulos auxiliares ROUTER e MAC. Estes representam, respectivamente, o algoritmo de roteamento e a camada MAC de comunicação [Tang and Gerla 2001]. O protocolo de roteamento e difusão utilizado neste trabalho é baseado no *flooding* [Obraczka et al. 2001]. Em um estudo recente [de Oliveira et al. 2007], vários protocolos de difusão para MANETs foram avaliados e o *simple flooding* obteve os melhores resultados para a taxa de entrega de mensagens. Neste protocolo, a confiança e tolerância a falhas podem ser consideradas devido a sua alta redundância. Porém, o *flooding* causa o problema conhecido como tempestade de *broadcast* (*the broadcast storm problem* [Tseng et al. 2002]), que também aumenta a latência. O *flooding* foi selecionado por questões de simplificação. Mas, pretende-se, como trabalho futuro, avaliar o uso de outros protocolos de roteamento confiável para fins de melhoria na latência.

Na Figura (b), é representado o módulo CONSENSUS. O submódulo CONSENSUS.MEDIATOR é responsável por determinar quando o nó deve participar ou não do consenso clássico. Para o consenso clássico, foi utilizado o protocolo de consenso genérico

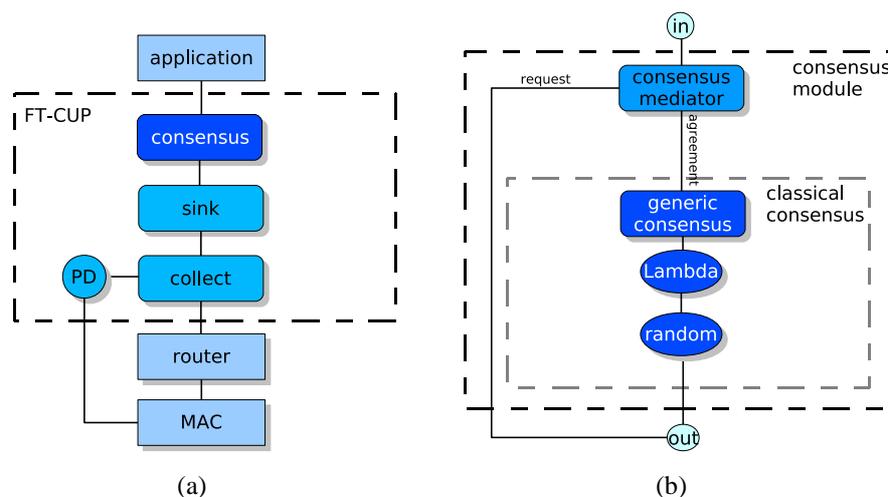


Figura 1. Modelo do FT-CUP (a) e modelo específico para o módulo CONSENSUS (b).

(GENERIC_CONSENSUS) proposto por Raynal e Guerraoui [Guerraoui 2004]. O módulo RANDOM implementa o oráculo Ben Or's de forma trivial [Ben-Or 1983].

Detectors de Participação. A implementação do detector de participação é fundamental para o funcionamento do FT-CUP. Porém, não existem propostas de implementação de detectores que garantam as propriedades especificadas, a saber, grafo k -SCO, k -OSR, etc. Se uma implementação para estes detectores, considerando-se o modelo de MANETs, é possível ou não continua sendo uma questão em aberto. Sendo assim, este trabalho busca avaliar a possibilidade de convergência do consenso mesmo utilizando um detector de participação que não satisfaz as propriedades do detector k -OSR.

Escolheu-se usar um detector de participação trivial para os nós. Este detector utiliza a característica de *broadcast* local inerente à camada MAC de redes MANETs para construir uma visão local formada pelos vizinhos de 1 nível de um dado nó. Esta implementação simples foi escolhida para este trabalho para avaliar o comportamento do FT-CUP nas situações de rede com pouco conhecimento sobre os nós.

PD-1hop. O detector proposto, denominado PD-1hop, retorna para os processos a lista dos nós que se encontram no seu alcance de transmissão (vizinhos). O detector inicia enviando mensagens do tipo "hello". Os vizinhos vão receber a mensagem e adicionar o nó fonte ao seu conjunto Π estimado. Esse algoritmo termina quando um tempo (*timeout*) é atingido. Durante este tempo os nós podem emitir novas mensagens "hello" em resposta a mensagens "hello" recebidas por outros nós. Vale ressaltar que o detector PD-1hop definido é implementado de forma independente dos parâmetros do FT-CUP e apenas com base nas relações de vizinhança real da rede. Num trabalho anterior realizado pelos autores [Costa and Greve 2007] foram considerados detectores mais robustos e com características definidas a partir da conectividade k do FT-CUP.

3.2. Modelo de Falhas.

No modelo de falhas adotado para as simulações, o processo falha por parada. A falha ocorre depois da execução do detector de participação pois, de acordo com o algoritmo

Tabela 1. Parâmetros de simulação

Parâmetros de simulação	
Simulador	OMNeT++ (3.4)
Quantidade de nós	10 à 50
Área	300x300, 400x400 e 500x500 m^2
Alcance <i>wireless</i>	125m e aleatório (25m, 50m, 125m, 250m)
Tempo de simulação	50 s
Repetições	30
Padrão de Mobilidade	Random Waypoint
Velocidade dos nós	de 0 à 10 m/s
Tempo de pausa	até 2s

do FT-CUP, se o processo falha antes da construção do grafo de conhecimento, ele simplesmente não participa desta construção e não será considerado como parte do sistema. A falha não afeta as camadas inferiores, como por exemplo a camada de roteamento, uma vez que o objetivo é testar a camada do FT-CUP. Sendo assim, fazemos com que a camada de roteamento tenha as melhores condições de execução possíveis.

3.3. Cenários

Ambiente da MANET. A escolha dos parâmetros associados ao ambiente de simulação (redes móveis *ad-hoc*) baseou-se nas considerações realizadas por [Kurkowski et al. 2005], que faz uma crítica aos trabalhos que simulam algoritmos para MANETs, mostrando como devem ser selecionados os parâmetros para que esses sejam realistas. A tabela 1 descreve os parâmetros gerais utilizados nas simulações. A *quantidade de nós*, os valores para a *área* e *alcance do dispositivo de comunicação wireless* foram selecionados visando definir a densidade da rede. Utilizamos inicialmente o raio do alcance de transmissão fixado em 125m. Testes preliminares foram realizados com diversas densidades (definimos a densidade através da variação da área). Em cenários de rede muito esparsa, o consenso não apresentou boas taxas de convergência, dado que frequentemente ocorria o particionamento da rede. A partir de determinada densidade, é visto que obtemos boas taxas de convergência. Sendo assim, selecionamos três valores de densidade: esparsa (área de 500x500 m^2), normal (área de 400x400 m^2) e densa (área de 300x300 m^2). Esses valores selecionados para a área e alcance de transmissão permitem uma conectividade entre os nós suficiente para evitar esse particionamento. Para simular cenários onde possam ocorrer conexões unidirecionais no grafo de conhecimento gerado pelo detector de participação, foi necessário um cenário onde nós possuem alcances de transmissão diferentes, onde estas conexões ocorrem naturalmente. Neste cenário, denominado rede híbrida, é associado aleatoriamente para cada nó um valor de alcance de transmissão entre os valores 25m, 50m, 125m e 250m. Com isso, o alcance médio dos nós na rede é 112.5m. Para a rede híbrida, foi utilizado a área de 500x500 m^2 , a mesma da rede esparsa.

Observando as considerações sobre os trabalhos que utilizam simulações em MANETs apresentados em [Kurkowski et al. 2005], pode-se usar o cálculo para a média de vizinhos para definir a média esperada de vizinhos por nó da rede nas simulações. Considerando a rede densa, para 10 nós, espera-se que cada nó tenha em média 5.45 vizinhos. Para 50 nós, a média é de 27.27 vizinhos. Para a rede normal, com 10 nós, espera-se que

cada nó tenha em média 3.07 vizinhos. Com 50 nós, a média é de 15.34 vizinhos. Na rede esparsa, para 10 nós, espera-se que cada nó tenha em média 1.96 vizinhos. Para 50 nós, a média é de 9.82 vizinhos. Para a rede híbrida calculou-se que, com 10 nós, espera-se que cada nó tenha em média 1.59 vizinhos. Para 50 nós, a média é de 7.95 vizinhos.

O *tempo de duração* de cada simulação é de 50 segundos, tempo suficiente para que houvesse uma convergência para um valor de decisão. O *padrão de mobilidade* usado é o *Random Waypoint* com a *velocidade* variando de 0 à 10 *m/s* e *tempo de pausa* de até 2s. Esta velocidade foi selecionada com o intuito de minimizar a interferência da mobilidade na convergência do consenso.

Ambiente do FT-CUP. Para os parâmetros específicos do FT-CUP, f (*máximo de falhas tolerável*) e Pf (*porcentagem real de falhas sobre f*), considera-se uma combinação de valores, para que se possa analisar seu impacto no FT-CUP e determinar as melhores configurações para o consenso. Dado que a verdadeira conectividade da rede, em termos de conhecimento, não pode ser pré-definida, escolheu-se não fixar um valor para k . Nas simulações, apenas iremos trabalhar com uma quantidade f de falhas na rede, correspondente a uma porcentagem dos nós que podem falhar, e espera-se nesse caso, que $f < k$, embora tal relação não possa ser garantida.

O parâmetro f é definido como uma porcentagem sobre n . Este parâmetro é fornecido como entrada para a execução das simulações, visto que os algoritmos do FT-CUP dependem dele para executar corretamente. Os valores considerados foram as porcentagens 0%, 30% e 50% em relação à n . Sendo assim, existem cenários que toleram uma quantidade de falhas que obedece os valores $f = 0.1$, $f = 0.3$ e $f = 0.5$ em relação à n . O Pf foi definido como um percentual em relação à f . Os valores considerados foram de 0%, 50% e 100% em relação à f , fazendo com que existam cenários sem falhas ($Pf = 0$), com metade das falhas possíveis ($Pf = 0.5$) e com o máximo de falhas tolerável ($Pf = 1$). O detector utilizado nas simulações é o PD1-hop, definido na seção 3.1, sendo usado um *timeout* de 2s. A combinação dos parâmetros f , Pf e da densidade da rede definem conjuntos de simulações. Cada conjunto é simulado por 30 vezes, sendo que o que se vê nos gráficos são os valores médios obtidos através dessas repetições.

4. Resultados e Análise de Desempenho

Para a análise de desempenho, algumas métricas foram escolhidas. A métrica *participantes detectados* refere-se à porcentagem média dos nós encontrados pelos detectores de participação associados a cada nó. A métrica *convergência para poço* representa a porcentagem dos nós que fazem parte do poço e que conseguem determinar que estão na componente poço. A *convergência do FT-CUP* refere-se à porcentagem de nós que terminam o consenso com uma decisão. A métrica *verificação do acordo* representa a porcentagem das execuções em que a propriedade acordo do consenso clássico (definida na seção 2) não é violada, ou seja, ao término da execução, apenas um valor é decidido por todos os nós. A *latência do FT-CUP* mostra a média dos tempos gastos para a convergência do FT-CUP. Nesta seção serão mostrados alguns gráficos resultantes das simulações relevantes para as nossas conclusões, omitindo os outros gráficos. Porém, estes evidenciam comportamentos similares aos observados nos gráficos inseridos no artigo e induzem as mesmas conclusões.

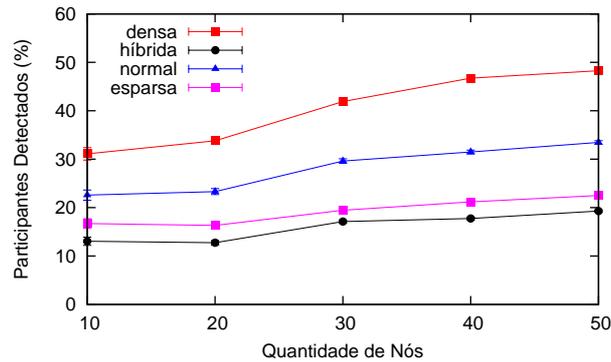


Figura 2. Resultado do detector PD-1hop em função da quantidade de nós.

4.1. Detector de Participação

A figura 2 representa os resultados de simulação para a métrica *participantes detectados*. Esta traduz efetivamente os valores de média de vizinhos esperados apresentado na seção 3.3. A variação em relação a esses valores é consequência da mobilidade dos nós. Esta mobilidade interfere no conhecimento dos nós, uma vez que novos vizinhos podem se mover para o alcance de transmissão de novos nós. Pode-se observar quatro curvas no gráfico: *densa* (alta densidade), *normal* (densidade normal), *esparsa* (baixa densidade) e *híbrida* (nós com diferentes alcances de transmissão). Neste gráfico, é visto que o desempenho do detector de participação é rigorosamente dependente da densidade da rede. A medida que se aumenta a densidade, aumenta-se o grau de conhecimento retornado por PD. No cenário híbrido, o resultado é semelhante ao resultado para a rede esparsa. Os valores de f e Pf não interferem no detector de participação PD-1hop e os gráficos gerados por todos os valores destes parâmetros são semelhantes ao da figura 2.

4.2. Convergência para Poço

A figura 3 representa a métrica *convergência para poço*. Os valores representados nos gráficos são oriundos da execução do algoritmo SINK do FT-CUP.

O gráfico 3(a) compara os resultados desta métrica em relação as configurações de densidade utilizadas. Neste gráfico pode-se perceber que quanto maior a densidade da rede, maior a porcentagem de nós caracterizados como participantes da componente poço. Observe também que, em um mesmo valor de densidade, a porcentagem diminui com o aumento dos nós. Através da análise anterior do resultado do detector de participação, vimos que, uma maior densidade tem como consequência o maior grau de conectividade. Sendo assim, é maior a probabilidade da componente poço ter um maior número de participantes.

O gráfico 3(b) compara os resultados em relação aos valores de Pf utilizados. Quanto maior o valor de Pf , maior a porcentagem de convergência. Isso ocorre devido a estratégia do algoritmo COLLECT e SINK em que cada nó p espera por $|\Pi_p| - f$ mensagens. Sendo assim, quando efetivamente os f nós falham ($Pf = 1$), ocorre a situação em que cada nó recebe as respostas esperadas dos nós corretos da rede. Quando $Pf = 0$, um nó p sempre ignora mensagens de f nós corretos. Estes f nós poderiam aumentar o conhecimento de p (através do COLLECT) e fazer com que este identifique-se como participante do poço.

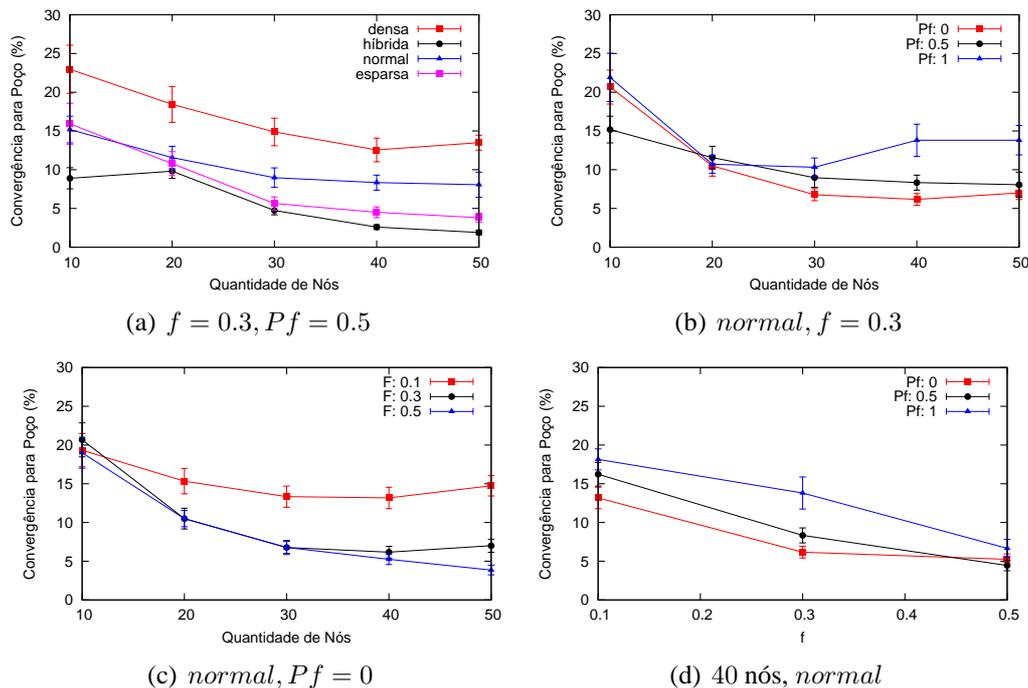


Figura 3. Convergência para poço em função da quantidade de nós (a), (b), (c) e para 40 nós em função de f (d).

O gráfico 3(c) representa a métrica em questão comparando os resultados dos valores de f utilizados. Através dele, observa-se que quanto menor o f , maior a convergência, ou seja, quando menos mensagens são ignoradas, a convergência para poço é maior. Isso confirma a justificativa anterior em relação ao gráfico 3(b). O gráfico 3(d) sintetiza essas observações comparando os valores de f e Pf para o valor fixo de 40 nós.

4.3. Verificação da Propriedade Terminação

Na figura 4 é mostrada a métrica *convergência do FT-CUP*. Observando o gráfico 4(a), é visto que, quanto maior a densidade maior a taxa de convergência. Isso ocorre devido ao maior grau de conectividade na rede.

Os gráficos 4(b) e 4(c) representam as curvas de convergência para os valores de Pf na densidade normal e híbrida, respectivamente. Pode-se observar nestes gráficos que uma quantidade de falhas menor (menor valor de Pf) produz um melhor desempenho, comportamento contrário ao ocorrido na métrica convergência para poço. Neste caso, pode-se concluir que a ocorrência de falhas interfere na execução do consenso. As falhas podem ocorrer em nós participantes da componente poço e ocasionar em bloqueio da execução do consenso clássico. Observando as curvas dos gráficos 4(b) e 4(c), conclui-se que, a depender da densidade da rede, o valor de Pf causa um determinado grau de interferência na convergência. Para redes com maior densidade, o valor de Pf não interfere significativamente na convergência. Sendo assim, pode-se dizer que, em redes densas, é possível obter a convergência total do FT-CUP mesmo na presença da quantidade máxima de falhas, limitada pelo valor de f . Além disso, em alguns casos a convergência é maior para um maior valor de Pf , assim como acontece na convergência para poço. Conclui-se disso que, não adianta usar o FT-CUP com uma tolerância a falhas (representada aqui pelo valor de f) maior do que a quantidade de falhas esperada no ambiente. Essa definição

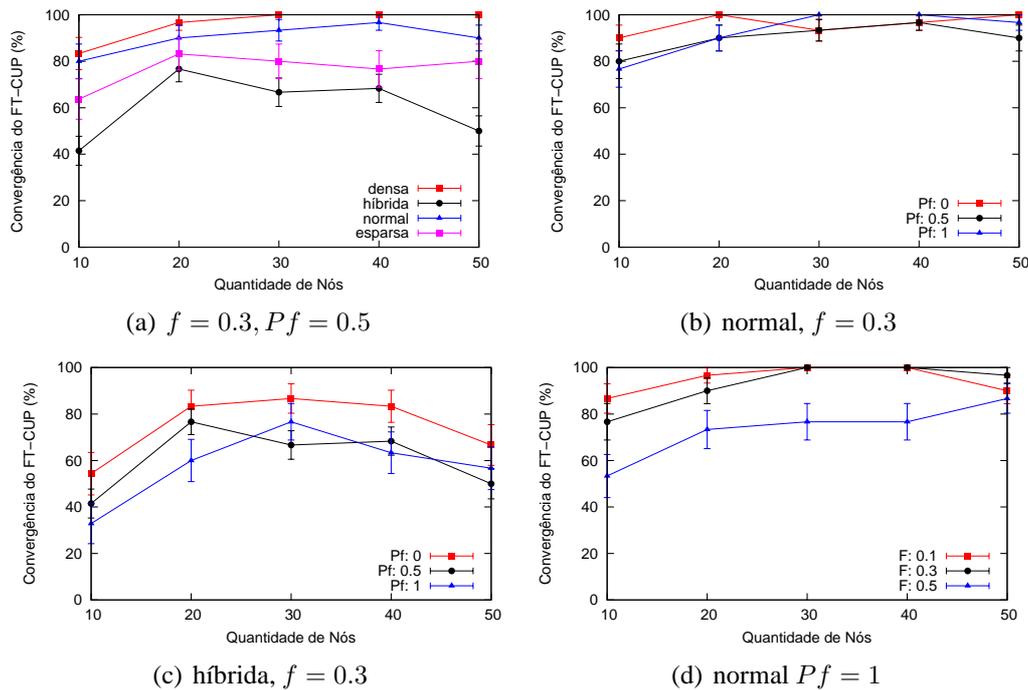


Figura 4. Convergência do FT-CUP em função da quantidade de nós.

inadequada pode ser prejudicial nos casos de execuções livres de falhas ou com falhas significativamente abaixo do esperado. O gráfico 4(d) representa as curvas de convergência para os valores de f na densidade normal onde é visto que quanto menor o valor de f , maior a convergência.

4.4. Verificação da Propriedade Acordo

Na figura 5 encontra-se representado os resultados para a métrica *verificação do acordo*. Em todas as situações (dos gráficos (a), (b), (c) e (d) e dos gráficos aqui omitidos), a porcentagem média das execuções que não violam a propriedade acordo superou o valor de 90%. A formação de mais de uma componente poço é a principal razão para que o acordo seja violado. Outra possibilidade seria um particionamento da rede, fazendo com que existam dois ou mais grupos de nós completamente isolados. Observe que a ocorrência destes fenômenos é pequena porém não é irrelevante.

Os gráficos 5(a), 5(b), 5(c) e 5(d) evidenciam que o comportamento desta métrica frente as diversas variações dos parâmetros usados nas simulações é bastante semelhante. A diferença mais relevante encontrada seria para a variação de f , observada no gráfico 5(d). Quanto maior o f , menor a quantidade de execuções que violam o acordo.

4.5. Latência

A figura 6 representa os resultados da métrica *latência do FT-CUP*. Essa latência varia em torno de 15 segundos. O protocolo de roteamento utilizado (*flooding*) faz com que exista um alto tráfego de mensagens na rede, ocasionando em tempos altos para o FT-CUP. Isso acontece também devido a alta complexidade dos algoritmos do FT-CUP (COLLECT, SINK e CONSENSUS), que é, no pior caso, $O(n^2)$. Portanto, esta métrica deve ser analisada apenas qualitativamente. Acredita-se que com o uso de um protocolo de roteamento mais adequado para MANETs este tempo diminua significativamente.

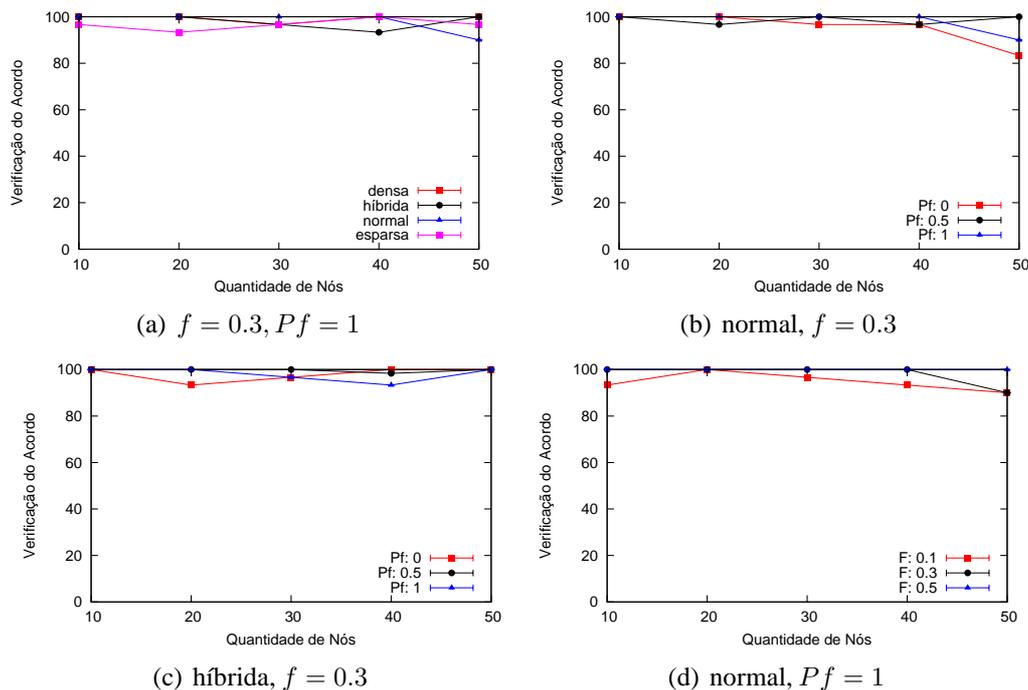


Figura 5. Verificação do acordo em função da quantidade de nós.

No gráfico 6(a), é visto que a latência aumenta com o menor valor de f , uma vez que para menores valores de f temos uma maior componente poço, o que aumenta a complexidade do consenso. Usando o mesmo raciocínio, no gráfico 6(b), a latência é maior para uma rede mais densa e no gráfico 6(c) a latência é maior para um maior valor de Pf . O gráfico 6(d) evidencia a diminuição da latência com o aumento de f .

5. Conclusões

O consenso FT-CUP é uma variação do problema do consenso para sistemas dinâmicos, como as redes MANETs. O FT-CUP não considera o conhecimento prévio dos participantes e utiliza uma abstração, o detector de participação, para obter um conhecimento parcial sobre a rede. Este trabalho estudou aspectos práticos da implementação do FT-CUP proposto em [Greve and Tixeuil 2007] no ambiente de MANETs. O FT-CUP utiliza uma abordagem modular, onde um algoritmo de consenso clássico é utilizado como parte de sua solução. O consenso genérico de [Guerraoui 2004] foi utilizado neste trabalho por permitir o uso das principais abstrações para resolução do consenso em ambientes assíncronos: detector de falhas, eleição de líder e oráculo aleatório. Neste trabalho, usou-se apenas o oráculo aleatório para a execução do consenso clássico. Uma implementação simples de um detector de participação (PD-1-hop) foi utilizada para avaliar o comportamento do FT-CUP em redes com conhecimento bastante limitado sobre os nós participantes.

Os itens a seguir definem as condições práticas onde o FT-CUP pode ser resolvido em uma MANET, além de características importantes observadas no comportamento do FT-CUP.

1. A escolha do detector de participação e do parâmetro f é essencial para a convergência do FT-CUP;

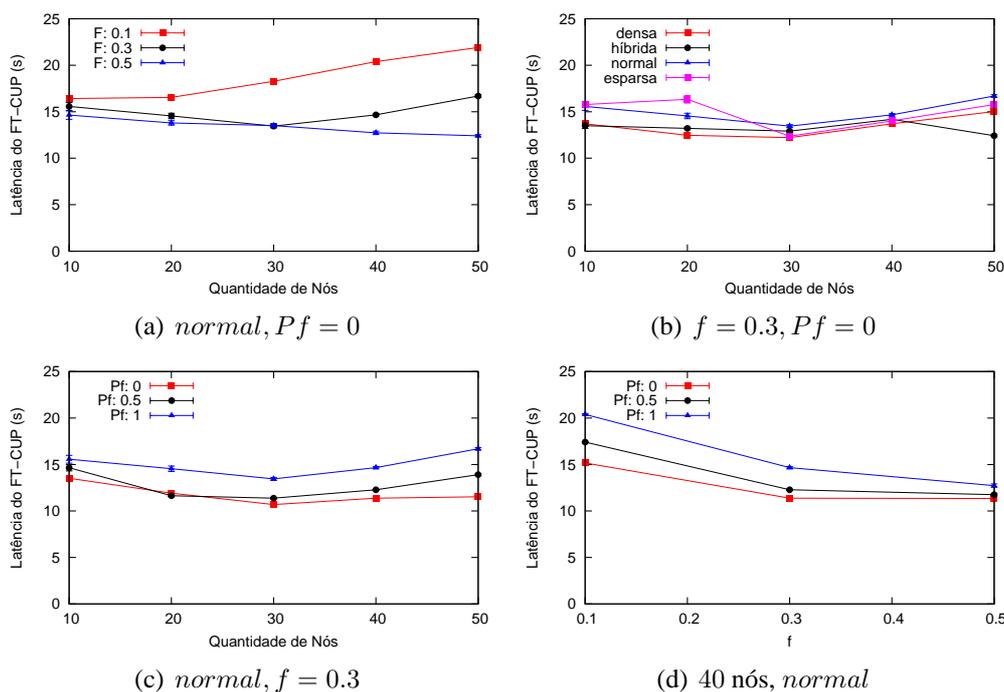


Figura 6. Latência do FT-CUP em função da quantidade de nós (a), (b), (c) e para 40 nós em função de f (d).

2. Através das simulações pode-se observar que mesmo um detector simples constrói um grafo de conhecimento que se aproxima dos requisitos teóricos para resolver o FT-CUP, representados por um grafo k -fortemente conexo com apenas uma componente poço;
3. Em redes com densidade alta, a convergência do FT-CUP obteve valores próximos a 100%, mesmo na presença de falhas. Conclui-se que, a partir de determinada densidade temos os requisitos necessários para a resolução do FT-CUP, mesmo na presença de uma quantidade significativa de falhas;
4. É possível resolver o FT-CUP mesmo com densidades menores. Nestes casos, deve-se tolerar menos falhas utilizando para f um valor mais adequado;
5. Em todas as simulações, mesmo nas redes mais esparsas, a porcentagem das execuções que não violam a propriedade de acordo superou o valor de 90%, sendo que nas redes densas atingiu 100% na maioria das situações. Conclui-se que, de acordo com a densidade da rede, o FT-CUP pode ser resolvido de maneira aproximada sem a violação desta propriedade.

Para trabalhos futuros, pretende-se simular o consenso com implementações para o detector de falhas e eleição de líder adequados ao ambiente utilizado. Como demais trabalhos futuros, pretende-se investigar mecanismos para garantir uma melhor taxa de convergência do FT-CUP e garantir que, mesmo em cenários desfavoráveis, a propriedade de acordo não seja violada.

Referências

Basile, C., Killijian, M., and Powell, D. (2003). A survey of dependability issues in mobile wireless networks. Technical report, LAAS CNRS, Toulouse, France.

- Ben-Or, M. (1983). Another advantage of free choice (extended abstract): Completely asynchronous agreement protocols. In *PODC '83: Proceedings of the second annual ACM symposium on Principles of distributed computing*, pages 27–30, New York, NY, USA. ACM.
- Cavin, D., Sasson, Y., and Schiper, A. (2004). Consensus with unknown participants or fundamental self-organization. In *Third Int. Conf. on Ad hoc Net. and Wireless (ADHOC-NOW 2004)*, pages 135–148, Vancouver, Canada.
- Cavin, D., Sasson, Y., and Schiper, A. (2005). Reaching agreement with unknown participants in mobile self-organized networks in spite of process crashes. Technical report, Ecole Polytechnique Federale de Lausanne.
- Chandra, T. D., Hadzilacos, V., and Toueg, S. (1996). The weakest failure detector for solving consensus. *J. ACM*, 43(4):685–722.
- Chandra, T. D. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *J. ACM*, 43(2):225–267.
- Costa, V. and Greve, F. (2007). Aspectos práticos da realização do consenso ft-cup em redes móveis ad-hoc. In *WTF 2007 VIII Workshop de Testes e Tolerância a Falhas realizado em conjunto com o SBRC 2007*, pages 1–14.
- de Oliveira, T. B., Costa, V. F., and Greve, F. (2007). On the behavior of broadcasting protocols for manets under omission faults scenarios. In *Lecture Notes in Computer Science*, volume 4746, pages 142–159. Springer.
- Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382.
- Greve, F. and Tixeuil, S. (2007). Knowledge connectivity vs. synchrony requirements for fault-tolerant agreement in unknown networks. In *DSN '07: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 82–91, Washington, DC, USA. IEEE Computer Society.
- Guerraoui, R.; Raynal, M. (April 2004). The information structure of indulgent consensus. *Transactions on Computers*, 53(4):453–466.
- Kurkowski, S., Camp, T., and Colagrosso, M. (2005). Manet simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61.
- Larrea, M., Fernandez, A., and Arevalo, S. (2004). On the implementation of unreliable failure detectors in partially synchronous systems. *IEEE Transactions on Computers*, 53(7):815–828.
- Obraczka, K., Viswanath, K., and Tsudik, G. (2001). Flooding for reliable multicast in multi-hop ad hoc networks. *Wirel. Netw.*, 7(6):627–634.
- Tang, T. and Gerla, M. (2001). Mac reliable broadcast in ad hoc networks. In *Proc. of the IEEE Military Communications Conference*, pages 1008–1013.
- Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P. (2002). The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2–3):153–167.