Detectores de Defeitos para Redes Wireless Ad Hoc

Giovani Gracioli e Raul Ceretta Nunes

[1]GMICRO/CT – Universidade Federal de Santa Maria (UFSM)
Campus Camobi - 97105-900 – Santa Maria/RS

{giovani,ceretta}@inf.ufsm.br

Resumo. *Uma rede wireless ad hoc é caracterizada pela ausência de qualquer infra-estrutura física, sendo que os nodos presentes na rede podem se comunicar diretamente sem a existência de um ponto de acesso. Devido a essa propriedade, a especificação de serviços de detecção de defeitos para redes ad hoc torna-se um desafio. Este artigo identifica os principais problemas que dificultam a migração para redes wireless ad hoc de algoritmos de detecção de defeitos projetados para redes fixas, apresenta algoritmos já propostos e discute como eles estão mutando para tratar esses problemas.*

1. Introdução

Um mecanismo de tolerância a falhas é normalmente dependente de um mecanismo de detecção de defeitos [Jalote 1994], que nada mais é do que um algoritmo distribuído que fornece informações sobre suspeitas de defeitos em componentes monitoráveis [Felber et al. 1999].

Com o intuito de obter uma solução eficiente aplicável à LANs e/ou WANs, diversos algoritmos para detecção de defeitos foram propostos ao longo dos anos [Chandra and Toueg 1996, Aguilera et al. 1997, Renesse et al. 1998, Felber et al. 1999]. Entretanto, o número de redes independentes (*ad hoc*) compostas por computadores móveis com capacidade de comunicação sem fio (*wireless*) cresce rapidamente e as soluções para LANs e WANs nem sempre podem ser reaproveitadas, pois redes *wireless* ad hoc são caracterizadas pela escassez de recursos (alcance de transmissão, consumo de energia, etc.) e pela mudança freqüente na topologia da rede, principalmente oriundo da mobilidade dos nodos e da maior probabilidade de ocorrer defeitos na comunicação entre os *hosts* [Mateus and Loureiro 2005].

Recentemente detectores de defeitos para redes *wireless* ad hoc foram propostos [Wang and Kuo 2003, Tai et al. 2004, Hutle 2004, Friedman and Tcharny 2005], mas não há conhecimento de uma avaliação das suas principais características. Este artigo revisa a literatura e identifica os principais problemas que dificultam a migração dos algoritmos de detecção de defeitos de sistemas distribuídos com nodos fixos para redes *wireless* ad hoc. Para tal o artigo descreve cada uma das propostas, salientando como cada uma resolve os problemas identificados, e discute as principais características de cada uma.

O restante desse artigo é organizado da seguinte maneira. A sessão 2 apresenta uma visão geral sobre conceitos de detectores de defeitos para sistemas distribuídos com nodos fixos e os principais algoritmos existentes. A sessão 3 mostra os principais problemas encontrados nas redes *wireless* ad hoc. Na sessão 4 algumas das estratégias existentes para detectores de defeitos para redes *wireless* ad hoc são descritas. A sessão 5 discute as estratégias apresentadas na sessão anterior. Finalmente, conclusões e trabalhos futuros são apresentados na sessão 6.

2. Detectores de Defeitos para Redes com Nodos Fixos

Um algoritmo de detecção de defeitos deve estar ciente que pode ocorrer falsas detecções, mas deve tentar garantir o máximo de exatidão no momento em que o defeito é detectado. Por isto o comportamento dos detectores de defeitos é especificado por duas propriedades [Chandra and Toueg 1996]: a abrangência (*completeness*) e a exatidão (*accuracy*). Abrangência se refere a capacidade de detectar defeitos de nodos que realmente falharam, enquanto que a exatidão se refere a capacidade de não cometer falsas suspeitas. Nesta seção revisaremos quatro modelos básicos para detecção de defeitos muito utilizados: o push, pull, Heartbeat e gossip. Estes detectores de defeitos podem monitorar processos, objetos ou nodos (um processo/objeto por nodo), mas, sem perda de generalidade, serão descritos a seguir como monitorando nodos apenas.

Dois modelos de detecção muito conhecidos são o Push e o Pull [Felber et al. 1999]. No modelo Push cada nodo monitorado envia periodicamente mensagens do tipo '*I am alive!*' ao nodo monitor (detector de defeitos), o qual suspeita de um nodo monitorado quando não recebe a mensagem em um certo intervalo de tempo T (*timeout*). No modelo Pull o monitor periodicamente envia mensagens do tipo '*Are you alive*?' aos nodos monitorados e aguarda mensagens do tipo '*Yes, I am.*'. Quando o monitor não recebe a resposta em um certo intervalo de tempo T (*timeout*), o monitor insere o nodo monitorado numa lista de suspeitos de terem falhado.

O modelo Heartbeat [Aguilera et al. 1997], diferentemente dos modelos Push e Pull, não utiliza *timeouts*. Neste modelo um nodo monitor p mantém para cada nodo monitorado q um contador de *heartbeats*. A idéia básica é detectar defeitos observando o contador. Periodicamente cada nodo monitorado envia uma mensagem '*I am alive*' (*heartbeat*) aos nodos monitores que, ao receber a mensagem, incrementam o contador correspondente àquele monitorado. Caso o contador não seja mais incrementado uma suspeita é levantada.

No modelo Gossip [Renesse et al. 1998] cada nodo da rede é um monitor/monitorado e mantém uma lista contendo o endereço e um inteiro (contador de heartbeat) para cada nodo do sistema. A cada intervalo *Tgossip*, um nodo escolhe aleatoriamente um ou mais vizinhos para enviar a sua lista. A lista recebida é unida com a lista que o receptor possui e o maior valor do contador de heartbeat presente nas listas é mantido na lista final. Cada nodo mantém o instante do último incremento do contador de heartbeat. Se o contador não for incrementado em um intervalo de *Tfail* unidades de tempo então o membro é considerado suspeito. Sua principal vantagem é tolerar a perda de mensagens, mas o tempo de detecção de defeitos aumenta se a probabilidade de perda de mensagens aumenta [Renesse et al. 1998].

3. Redes Wireless Ad Hoc

Em uma rede *wireless* ad hoc os nodos são capazes de se comunicar diretamente com outros sem a necessidade da criação de uma infra-estrutura de rede tal como *backbones* ou pontos de acesso [Mateus and Loureiro 2005]. Os nodos em uma rede ad hoc se comunicam sem uma conexão física, formando uma rede '*on the fly*', no qual os dispositivos fazem parte da rede apenas durante a comunicação ou, no caso de dispositivos móveis, apenas enquanto estão dentro do alcance de transmissão. Uma rede ad hoc pode ser formada não somente por computadores, mas também por qualquer aparelho que tenha um

dispositivo *wireless*, como por exemplo PDAs, celulares ou nodos sensores.

Abaixo são identificados os principais problemas em redes *wireless* ad hoc que dificultam a migração de algoritmos de detecção de defeitos em sistemas distribuídos com nodos fixos [Tai et al. 2004, Wang and Kuo 2003, Hutle 2004, Friedman and Tcharny 2005].

Problema da Mobilidade: em uma rede *wireless* o conceito de mobilidade é muito importante e o algoritmo de detecção de defeitos deve considerar essa propriedade. Os nodos estão sempre em movimento. Hora eles fazem parte dos vizinhos, hora eles estão fora do alcance de transmissão. O serviço de detecção de defeitos não deveria suspeitar de um nodo somente porque ele saiu momentaneamente da vizinhança.

Problema na Qualidade da Comunicação Wireless: esse problema abrange a maior vulnerabilidade a perda de mensagens e desempenho da rede (largura de banda, atraso na comunicação, latência, etc). O detector de defeitos deve ser capaz de trabalhar sem violar as propriedades de abrangência e exatidão, que são desejáveis para um detector.

Problemas de Bateria e Consumo de Energia: como são alimentados por bateria, os nodos possuem uma séria limitação de energia e o modelo de troca de informações e dados da rede deve ser otimizado para consumo mínimo de energia, maximizando o tempo de operação de um nodo. Achar meios de economizar energia, descobrindo algoritmos cada vez melhores de disseminação de informação pela rede é um grande desafio.

4. Estratégias para Detecção de Defeitos em Redes Wireless Ad Hoc

Recentemente existem algumas propostas de detectores de defeitos para redes *wireless* ad hoc [Tai et al. 2004, Wang and Kuo 2003, Hutle 2004, Friedman and Tcharny 2005], dentre as quais pode-se identificar duas estratégias de operação: detectores baseados em *cluster* e detectores baseados no modelo *gossip*.

4.1. Detector Baseado em Formação de Cluster

O detector de defeitos proposto em [Tai et al. 2004] é baseado em uma arquitetura de comunicação em *cluster*, ou seja, num círculo com o raio igual ao alcance de transmissão do nodo central. Observe que qualquer nodo dentro do *cluster* está a um *hop* do seu nodo central, chamado de *clusterhead* (CH). Um nodo que está a uma distância de um *hop* de dois *clusterheads* é chamado de *gateway* (GW). Os nodos que não são nem CHs e nem GWs são chamados de membros ordinários (OM). Para resolver o problema da mobilidade, os *clusters* são controlados e reconfigurados de maneira independente e dinâmica, um em relação a outro. O detector possui dois algoritmos: um responsável pela formação dos *clusters* e outro pelo serviço de detecção de defeitos (SDD).

O algoritmo de formação de *cluster* (AFC) é uma variante do algoritmo proposto em [Gerla and Tsai 1995] com algumas características próprias: (i) assegura que um GW é afiliado a um e somente um *cluster* e que existirão vários nodos candidatos a GW; (ii) cria CHs substitutos (*Deputy Clusterheads* - DCHs) e *backup gateways* (BGWs) que deixam o detector mais flexível a falhas dos nodos; e (iii) permite que as informações coletadas no primeiro *round* do AFC sejam utilizadas também no primeiro *round* do algoritmo do SDD[1]. Depois que o CH for identificado pelo AFC, ele identifica cada mem-

[1]Depois do primeiro round, os algoritmos de formação de cluster e do SDD executarão separadamente.

bro pertencente ao seu *cluster* e transmite a todos (*broadcast*) a organização do conjunto. Cada membro terá uma visão inicial de todo o *cluster* local e o CH saberá de que nodos deverá esperar as mensagens durante a execução do SDD.

O Algoritmo do SDD proposto em [Tai et al. 2004] trabalha no modelo Heartbeat e consiste em 3 fases. Na primeira, cada membro do *cluster* local C envia ao CH uma mensagem heartbeat que contenha o seu NID (identificador do nodo) e um bit indicador. Na segunda, todo nodo envia para o seu CH um relatório de todos os heartbeats dos nodos que ele conseguiu ouvir durante a primeira fase. Na terceira, analisando as informações coletadas na primeira e segunda fases, o CH identifica os nodos suspeitos e então transmite uma mensagem de atualização para o *cluster*, indicando sua decisão. Um nodo v é determinado falho se e somente se o CH não receber nem o heartbeat de v na primeira fase nem o relatório de v na segunda fase e nenhum dos relatórios recebidos pelo CH na segunda fase reflete o conhecimento do heartbeat pertencente ao v. Um CH, por sua vez, é julgado suspeito se e somente se o DCH não receber nem o heartbeat do CH na primeira fase nem o relatório do CH na segunda fase, e nenhum dos relatórios que o DCH receber contém informações sobre o heartbeat do CH e ainda se o DCH não receber a mensagem de atualização dos estados enviada pelo CH na terceira fase.

Um problema na comunicação dentro de um *cluster* é que múltiplos vizinhos podem responder a um pedido de um nodo simultaneamente, resultando em um desperdício de energia. Para tratar desse problema, cada nodo, assim que receber uma mensagem de um vizinho, ajusta um período de espera em função do seu NID (único na rede), permitindo que cada nodo tenha um período de espera diferente e balance seu consumo de energia.

4.2. Detectores Baseados no Modelo Gossip

Wang e Kuo [Wang and Kuo 2003] proporam um SDD baseado no modelo *gossip*. O esquema de detecção possui duas fases: a interna e a externa. Os nodos da rede são representados a partir de um grafo. Na fase interna do algoritmo, as arestas dos nodos são marcadas com uma direção e na fase externa com a direção oposta. As mensagens são transmitidas na direção em que a aresta se encontra. A idéia básica é fazer com que os nodos sejam automaticamente e dinamicamente gerados no grafo de conectividade na fase interna para recolher as mensagens gossip dos nodos da sua vizinhança, depois a mensagem é difundida na maneira inversa. A cada *Tgossip* as direções das arestas são invertidas. Para manter a informação sobre os vizinhos a um-hop, foi usado um tipo de mensagem chamado de mensagens *HELLO*. Essa mensagem é enviada periodicamente e assincronamente por cada nodo para declarar a sua presença, tratando assim do problema da mobilidade. A informação sobre a vizinhança é atualizada por mensagens *HELLO* e as informações sobre defeitos através de mensagens gossip.

Friedman e Tcharny [Friedman and Tcharny 2005] também criaram uma adaptação do modelo de detecção de defeitos *gossip* para redes ad-hoc. O algoritmo usa um contador de *heartbeat* que é incrementado toda vez que um novo *heartbeat* de algum vizinho é recebido. Na teoria, a cada T unidades de tempo um *heartbeat* deveria ser recebido, mas na prática isso não acontece devido ao problema da mobilidade ou devido ao aumento do caminho (maior quantidade de nodos presente na rede) e com isso o tempo T pode ser excedido. Para evitar isso, o algoritmo permite a passagem de no máximo Y

heartbeats para não suspeitar de um nodo entre o envio de dois *heartbeats* consecutivos. O algoritmo espera que os nodos estejam em movimento, hora fazendo parte da rede, hora estando fora do alcance de transmissão. O principal objetivo é encontrar um valor de *Y*, com o passar dos *rounds*, que mesmo que um nodo vá para fora do raio de ação ele não seja suspeito, pois o valor de *Y* estará ajustado para o tempo desse movimento.

Outro detector baseado no modelo *gossip* foi proposto por *Hutle* [Hutle 2004]. Neste algoritmo também usou-se uma alternativa para ajustar o tempo de uma detecção de defeito. Um contador de distância é usado para ter uma estimativa da atual distância entre dois nodos. Inicialmente um nodo *p* envia para outro nodo *q* a sua distância contendo o valor zero (pois o nodo somente conhece ele mesmo). O nodo *q* incrementa em um essa distância ao ser recebida. Após, o nodo *q* repassa a sua distância até *p* para um outro nodo que pertence a sua vizinhança, que também a incrementa ao receber. Assim, pode-se ter uma estimativa de quantos *hops* de distância está um nodo até outro. Para detectar um defeito é mantido um conjunto de todos os processos que o detector de defeitos não suspeita, chamado de lista de detecção. Conseqüentemente, um processo irá suspeitar de outro se este não estiver na sua lista de detecção.

5. Discussão

As estratégias para construção de detectores de defeitos para redes *wireless* ad hoc apresentadas na sessão anterior tentam resolver os problemas citados na sessão 3. Abaixo, são comentadas as principais características de cada estratégia.

Um detector de defeitos com uma arquitetura de comunicação baseada em *cluster* permite que haja uma visão sobre a hierarquia dos nodos na rede (CH, GW e OM). Permite também, que um algoritmo de detecção de defeitos seja executado paralelamente dentro do *cluster* [Tai et al. 2004]. Em cada *cluster* existe uma divisão das tarefas, o AFC é responsável por tratar do problema da mobilidade e escalabilidade dos nodos, enquanto o algoritmo do SDD é responsável em detectar o defeito. O resultado de uma detecção dentro de um *cluster* é enviado para outros *clusters* através dos GWs, de maneira que seja resistente ao problema da vulnerabilidade a perda de mensagens [Tai et al. 2004]. Essa estratégia explora a redundância de mensagens que é comum em redes *wireless* ad hoc, o que atenua os efeitos do problema da vulnerabilidade a perda de mensagens. Isto deixa o SDD mais eficiente, robusto e ajuda a contornar os problemas de conectividade causados pela gerência de energia em cada nodo (*sleep/wakeup*).

Os detectores de defeitos baseados no protocolo gossip são flexíveis por não depender de uma hierarquia fixa, ou seja, independe de uma topologia [Renesse et al. 1998]. A configuração dos nodos é feita dinamicamente através da troca de mensagens, tratando do problema da mobilidade. Em geral, são menos eficientes do que abordagens hierárquicas [Tai and Tso 2004], mas possuem a característica de serem robustos, pois cada nodo difunde as suas informações para um ou mais de seus vizinhos fazendo com que essas informações possam chegar para um outro nodo que não faz parte da sua vizinhança, tornando a detecção abrangente. Se necessário, podem sacrificar o alcance do nodo pela velocidade ou eficiência [Tai and Tso 2004].

6. Conclusão e Trabalhos Futuros

Este artigo identificou os problemas que dificultam a migração de detectores de defeitos de redes com nodos fixos para redes *wireless* ad hoc, bem como apresentou e discutiu as principais estratégias de detecção que tratam dos problemas identificados. Pode-se perceber que os algoritmos Push e Pull não são usados em redes *wireless* ad hoc, mas sim algoritmos baseados em cluster ou *gossip*. Isso se deve ao fato da periodicidade do envio das mensagens (*timeout*) que aumenta a quantidade de mensagens transmitidas, o que aumenta o consumo e a perda de mensagens. Como os detectores de defeitos analisados não diferem mobilidade de defeito, estamos explorando novos algoritmos em um simulador.

Referências

Aguilera, M. K., Chen, W., and Toueg, S. (1997). Heartbeat: A timeout-free failure detector for quiescent reliable communication. In *Workshop on Distributed Algorithms*, pages 126–140.

Chandra, T. D. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267.

Felber, P., Défago, X., Guerraoui, R., and Oser, P. (1999). Failure detectors as first class objects. In *Proceedings of the International Symposium on Distributed Objects and Applications (DOA'99)*, pages 132–141, Edinburgh, Scotland.

Friedman, R. and Tcharny, G. (2005). Evaluating failure detection in mobile ad-hoc networks. *Int. Journal of Wireless and Mobile Computing*.

Gerla, M. and Tsai, J. (1995). Multicluster, mobile, multimedia radio network. *Journal of Wireless Networks*, 1(3):255–265.

Hutle, M. (2004). An efficient failure detector for sparsely connected networks. *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2004), Innsbruck, Austria*.

Jalote, P. (1994). *Fault tolerance in distributed systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Mateus, G. and Loureiro, A. (2005). *Introdução à computação móvel, 2ª Edição*.

Renesse, R. V., Minsky, Y., and Hayden, M. (1998). A gossip-style failure detection service. Technical Report TR98-1687.

Tai, A. T. and Tso, K. S. (2004). Failure detection service for ad hoc wireless networks applications: A cluster-based approach. Technical Report IAT-302184, IA Tech, Inc., Los Angeles, CA.

Tai, A. T., Tso, K. S., and Sanders, W. H. (2004). Cluster-based failure detection service for large-scale ad hoc wireless network applications. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN 2004)*.

Wang, S.-C. and Kuo, S.-Y. (June 2003). Communication strategies for heartbeat-style failure detectors in wireless ad hoc networks. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN 2003), (San Francisco, CA)*, pages 361–370. IEEE Computer Society.