

3º SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES (3º SBRC)

IMPLEMENTAÇÃO DE UM ALGORITMO DE ROTEAMENTO ADAPTATIVO PARA OS GATEWAYS INTERREDES-PUC.

LUIS CARLOS TREVILIN.

SUMARIO:

O encaminhamento eficiente de pacotes entre gateways de redes interligadas constitui-se num problema relevante dentro do projeto de protocolos inter-redes.

Este trabalho apresenta a descrição do método utilizado na implementação de um procedimento distribuído adaptativo para encaminhamento de pacotes no protocolo inter-redes-PUC, utilizando recursos de endereçamento em grupo, para construção e atualização das tabelas de encaminhamento nos gateways.

I. INTRODUÇÃO

Na implementação do protocolo Inter-redes-PUC (1) não nos preocupamos, em princípio, com o problema de roteamento de pacotes pelos gateways. Na realidade, assumimos na época que as tabelas de encaminhamento nos gateways seriam estáticas, deixando esta questão para ser resolvida posteriormente.

Doutorando em informática pela PUC-RJ; professor assistente do depto. de Computação e Estatística da UFSCar; com interesse em redes de computadores e sistemas distribuídos, sistemas de informação e ensino de informática.

Este trabalho apresenta uma proposta de implementação de um método adaptativo de atualização dessas tabelas, baseado no algoritmo de Dijkstra (Shortest-Path-First), atualmente empregado na política de roteamento de pacotes da rede Arpa (2).

Inicialmente faremos uma breve apresentação do método, ressaltando suas principais características e estruturas de dados, exigidas para sua implementação. Em seguida, apresentaremos a estrutura do protocolo Inter-redes-PUC, tal como foi implementado num gateway. Apresentaremos, depois, os procedimentos desenvolvidos para a implementação do método e o protocolo a ser utilizado na sua operação, e, por fim, apresentaremos algumas conclusões sobre os resultados do trabalho.

2. O METODO DE ROTEAMENTO PROPOSTO

Conforme mencionado na introdução, este método baseia-se na técnica utilizada atualmente pela rede Arpa, na qual, cada nó rede, durante um certo intervalo de tempo, o custo médio de transmissão de um pacote, em termos de seu retardo médio, obtido através das medições do número médio de pacotes na fila de transmissão e do tamanho médio destes pacotes. Toda a vez que houver uma variação significativa deste "custo", em relação ao custo anterior, ele envia a todos os demais nós da rede um pacote contendo o valor do novo custo, medido nas linhas onde houve variação. Todos os nós possuem informações sobre a topologia da rede, logo, podem calcular qual a linha de saída mais indicada para cada nó de destino.

A razão de se ter escolhido este método baseia-se no fato de que, do ponto de vista dos gateways, tem-se uma rede do tipo

'store-and-forward', que tem por nós os gateways e por linhas de comunicação as redes-PUC intermediárias. Quando um computador hospedeiro de uma rede A deseja se comunicar com outro em uma rede D, passando pelas redes B e C, a operação é feita através de gateways que ligam AB, BC e CD e que operam nesse caso como nós de uma sub-rede de comunicações do tipo armazenar-e-encaminhar, semelhante neste aspecto a rede Arpa.

O método para cálculo das linhas de saída é resultado de modificações efetuadas no algoritmo de Dijkstra para cálculo do caminho mínimo entre a origem e todos os destinos em ordem não decrescente de custo. O processo de busca destes caminhos envolve o uso de uma matriz de adjacência que oferece as conexões da rede, bem como os custos entre cada par de nós conectados, para gerar uma árvore de caminhos mínimos do nó de origem para cada destino e a partir daí, uma tabela dos nós de saída é construída para ser utilizada como indicadora de roteamento.

Os algoritmos que implementam este método serão tratados nas seções seguintes.

A atualização dessa árvore, e conseqüentemente, da tabela de encaminhamento, quando o nó recebe um pacote de atualização, se processa da seguinte maneira: - Um nó ao notar que a variação no custo de transmissão por uma determinada linha, até seu vizinho, foi superior ao limite mínimo, envia um pacote de atualização com o próximo número de sequência para todos os demais nós da rede, num processo de comunicação por enchente. Cada nó ao receber um tal pacote, verifica se seu número de sequência é o esperado e, se for, copia seu conteúdo e o repassa para todos os nós adjacentes sem alterar seu conteúdo e, em seguida, procede a

atualização de suas tabelas. Senão, o pacote é descartado, pois consiste de uma cópia de um pacote antigo. Este procedimento garante a entrega da informação a todos os nós da rede sem uma sobrecarga sensível de cópias desses pacotes.

Um nó que recebe um pacote de atualização com alterações de custo, procede a alteração de sua tabela de encaminhamento, comparando, numa tabela de precedências, o custo novo para se encaminhar pela rota antiga e o custo de se encaminhar pela melhor rota alternativa. Este será então o novo caminho até ele e até os nós abaixo dele na sub-árvore de precedências formada a partir dele.

O tempo de medição até uma próxima possível atualização leva em consideração que as mudanças de custo numa determinada linha não ocorrem numa frequência maior que as medições. Na rede Arpa, este tempo é estimado em 10segs (2), entretanto, este parâmetro deve ser função das características da rede que se está projetando.

As principais características desse método são:

- Utiliza poucos recursos do sistema para manutenção e processamento das informações sobre as condições da rede, uma vez que as atualizações só são encaminhadas se houver uma variação real dessas condições. Além disso, o tamanho do pacote de atualização não é tão grande, o que não sobrecarrega nem os buffers dos nós nem as linhas, o que poderia introduzir uma sobrecarga de tráfego nos gateways.
- Responde rapidamente às mudanças da topologia, uma vez que todos os nós recebem os pacotes de atualização dos custos, num

intervalo pequeno de tempo, sendo que o processamento dessas atualizações é rápido.

- Em função do processo de medição e estabelecimento dos custos de transmissão, este método contribui para a detecção e controle de congestionamento através da determinação rápida de rotas alternativas (quando estas existirem, obviamente), aliviando as filas de transmissão do nó congestionado.

- Por utilizar o algoritmo de Dijkstra, que procura por caminhos de custo mínimo entre uma origem e todos os destinos, em ordem não decrescente de custo, este método procura encaminhar os pacotes pelos caminhos efetivamente mais curtos, até os nós de destino tendo em vista os custos para seu encaminhamento através de cada linha.

Detalhes sobre medidas de performance deste método, implementado na rede Arpa, podem ser obtidas em (2).

3.A ESTRUTURA DO PROTOCOLO INTERREDES-PUC NUM GATEWAY

O protocolo inter-redes-PUC está implementado num gateway através de tres módulos principais (fig. 2): - O módulo Super-BW, encarregado de encaminhar os pacotes recebidos pelo gateway, para as filas de transmissão de pacotes das redes apropriadas, as quais o gateway está conectado, de acordo com suas tabelas de encaminhamento; - O módulo TX, de transmissão, que verifica se há um descritor de pacotes na fila de transmissão e, nesse caso transmite-o via quadros de nível 2 (nível de enlace); - E o módulo de recepção RX, que recebe quadros do nível 2, oriundos da

rede que ele está recebendo e os insere num buffer de recepção disponível para o Super-GW.

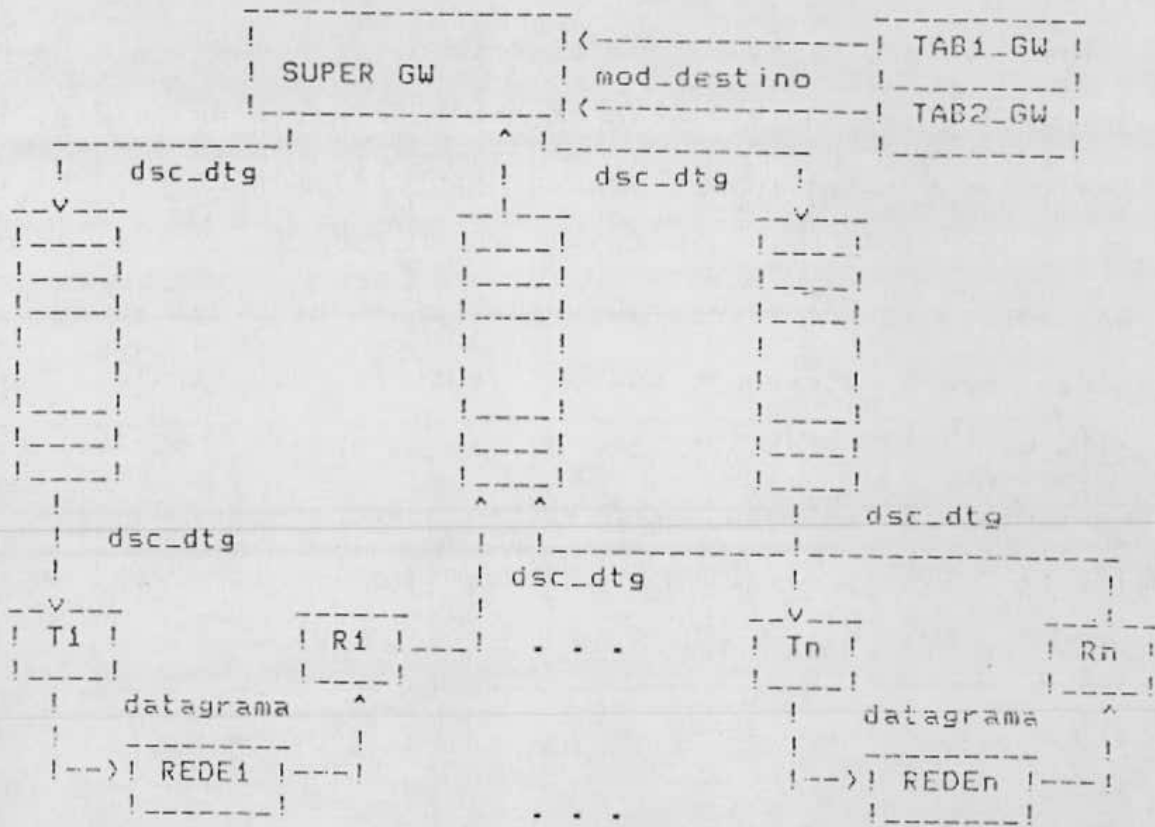


fig.2 - Estrutura do protocolo IR-PUC num gateway.

Existem num gateway tantos módulos de TX e RX quantas forem as redes às quais ele estiver conectado. Seu funcionamento ocorre da seguinte forma: - Uma mensagem gerada num módulo da rede A, dirigida a um módulo da rede C, é difundida pela rede A, para o grupo de gateways a ela conectados. Os módulos RX recebem esta mensagem e verificam se o gateway tem, em sua tabela de encaminhamento, uma saída para a rede solicitada (rede C no caso). Se tiver, ele então copia o conteúdo da mensagem para um buffer alocado para ela e monta um descritor dessa mensagem que é inserido na fila de recepção, a ser tratada pelo Super-GW. Este

analisa cada descritor de mensagem da fila e, de acordo com a tabela de roteamento, remove o descritor desta fila e o insere na fila de transmissão da rede de saída.

As tabelas de encaminhamento contém informações sobre que redes aos quais o gateway esta conectado, e à quais ele serve como intermediário. A iniciação dessas tabelas, a princípio, era feita quando o sistema era gerado, permanecendo estáticas daí por diante.

Analisando-se este problema sob o ponto de vista de que são poucas as redes interconectadas, o emprego de técnicas mais sofisticadas de atualização destas tabelas parece ser desnecessário. Entretanto, considerando-se a possibilidade do crescimento desses sistemas, o uso dessas técnicas mostra perfeitamente válido.

4. IMPLEMENTAÇÃO DO MÉTODO SPF NOS GATEWAYS DA IR-PUC.

A implementação do método modificado de Dijkstra compreende três tarefas principais: - A primeira, de desenvolver os algoritmos para iniciação, atualização e geração das tabelas de encaminhamento de mensagens dos gateways; - A segunda, de adaptar os módulos de transmissão, supervisão e recepção dos gateways, para que possam trabalhar com estes novos procedimentos; - E por fim, a terceira, de implementação do procedimento de medição e do protocolo de atualização desses custos medidos nas filas de transmissão.

Apresentaremos aqui os algoritmos que foram utilizados na iniciação e atualização das tabelas de precedências entre os nós e a geração do diretório de encaminhamento.

- Desenvolvimento dos algoritmos:

- Algoritmo de montagem da matriz de precedências dos gateways da IR, de acordo com a matriz de adjacência com os custos assinalados nos elementos não nulos.

- 1- Declare todos os nós como 'não na lista';
 - 2- Coloque o nó de origem como nó de raiz da árvore de precedências;
 - 3- Enquanto lista # de vazio faça:
 - 3.1- Remova da lista o nó mais próximo em distancia total da origem e coloque-o na árvore;
 - 3.2- Para cada vizinho do nó removido faça:
 - 3.2.1- Se o nó não está na árvore, então, se está na lista, então
 - 3.2.1.1- atualize sua distancia da raiz,
 - 3.2.1.2- coloque-o na lista;
 - 3.2.2- Fim Para;
 - 3.3- Fim Enquanto;
- 4- Fim.

- Algoritmo de geração do diretório (tabela de encaminhamento), a partir da tabela de precedências.

1- Faça: diretório(nó de origem) = 0,

2- Para cada nó da tabela de precedências, a partir do nó na posição 2 da tabela faça:

2.1- Se o predecessor do nó na tabela = nó de origem
então,

2.1.1- Diretório (nó) := nó

senão,

2.1.2- Procure na tabela de precedências para os nós nas posições inferiores até a do nó em questão, qual a posição do nó predecessor do nó em questão;

2.1.3- Faça diretório (nó em questão) :=
diretório (nó predecessor);

2.2- Fim Para;

3- Fim.

- Algoritmo de atualização da tabela de precedências, a partir de um vetor de atualização de custos.

1- Para cada par (nó_dest, Custo_novo) de atualização, em que houve alteração significativa de custo (retardo), no canal que liga a origem a ele, faça:

1.1- Se o seu predecessor na tabela for o nó origem da atualização, ou, se o custo novo decresceu em relação ao custo medido anteriormente, então:

- 1.1.1- Se o seu predecessor na tabela é a origem, então, atualize sua distancia até a raíz, através dela;
- 1.1.2- Lance o nó de destino como nó pendente e atualize a matriz de adjacência;
- 1.1.3- Repita as operações seguintes até não haver mais nós pendentes:
 - 1.1.3.1- Procure um nó pendente, e para cada um de seus sucessores faça:
 - 1.1.3.1.1- Se seu predecessor for o nó pendente como seu predecessor na tabela, então:
 - 1.1.3.1.1.1- Procure dentre todos os seus possíveis nós predecessores, aquele que lhe de menor distancia da raíz e tome-o como seu novo predecessor;
 - 1.1.3.1.1.2- Assinale o nó sucessor do nó pendente como um nó também pendente;

senão:

1.1.3.1.1.3- Verifique qual a menor distancia até a raiz entre seu predecessor na tabela e o nó pendente, e tome o de distancia menor como seu predecessor;

1.1.3.1.1.4- Se houve troca, assinale este nó como um nó pendente;

1.1.3.1.2- Fim Se;

1.1.3.2- Fim da Repetição;

1.2-Fim Para;

2- Fim.

As duas fases seguintes serão omitidas por serem dependentes da implementação. Cabe apenas ressaltar que o processo de medição da variação dos comprimentos das filas de transmissão ocorre em intervalos definidos previamente, medindo-se a cada transmissão, o comprimento da fila e calculando, ao final do intervalo, qual seu comprimento médio.

Quanto ao protocolo de atualização, o formato do pacote de atualização está assim estruturado:

<--- 1 octeto --->

1	!	Tipo do Pacote: (02H)	!
2	!	Gateway de Origem	!
3	!	Numero de Sequência	!
4	!	Num. Pares de Atualiz	!
5	!	Gateway Vizinho 1	!
6	!	Custo Novo 1	!
	!	o	!
	!	o	!
	!	o	!
n-1	!	Gateway Vizinho m	!
n	!	Custo Novo m	!

- Descrição dos campos:

- 1..Tipo do Pacote: Tipo do datagrama de atualização. Valor (02H).
Indica que é um dtg. de roteamento.
- 2..Gateway de Origem: Indica de qual nó partiu o dtg. com as medições.
- 3..Número de Seq.: É a maneira de se controlar a enchente de pacotes na rede, isto é, toda vez que um pacote chegar ao gateway com um número de sequência diferente do esperado, ele é descartado pois consiste de uma duplicata ou encarnação de um pacote anterior.
- 4..Número de Pares de Atualiz.: Indica quantos pares de bytes seguem com informações.

5..m: Conjunto de Pares (Nó Vizinho, Custo Novo): com identificação para cada par, do gateway vizinho para o qual foi notada a variação de custo de transmissão, e qual o valor deste novo custo.

5. CONCLUSOES.

A implementação do processo de roteamento adaptativo distribuído na inter-redes-PUC, encontra-se em fase final de programação e testes. Resultados mais palpáveis de sua performance poderão ser obtidos quando for completada esta fase e puderem ser colhidos resultados da simulação que deverá ser feita em seguida.

Acreditamos, entretanto, que, tal como nas redes que utilizam este método, ele deverá ser bastante eficiente para operar na inter-redes-PUC, posto que o número de gateways a serem interligados não será tão grande a ponto de resultar numa matriz de adjacência que venha a ocupar um espaço excessivo de memória ou num sistema cuja atualização de suas tabelas seja muito lenta.

Este trabalho foi desenvolvido como parte da disciplina Redes de Computadores do depto. de informática da PUC-RJ, durante o segundo período de 1984 sob a orientação do professor Daniel Schwabe.

6. BIBLIOGRAFIA.

- (1). MENASCÉ, D.A. 'Especificação de um protocolo para ligação de redes locais'.
Anais do XI SEMISH, Viçosa, julho de 1984.
- (2). MACQUILLAN, et alli 'An overview of the new algorithm for the Arpanet'.
IEEE- Transaction on Communications, 1979.
- (3). MEDEIROS, L.; TREVELIN, L.C.; DUARTE FILHO, N.L.; SABOIA, R. 'Protocolo de Comunicação Inter-redes-PUC: - Manual de Implementação'.
Depto. de Informática - PUC-RJ, relatório interno, julho de 1984.
- (4). TREVELIN, L.C. 'Algoritmo de Roteamento do Protocolo Inter-redes-PUC: - Manual de Implementação'.
Depto. de informática - PUC-RJ, relatório interno, janeiro de 1985.
- (5). BOGOS, D.R., et alli: 'PUP: An Internetwork Architecture'.
P.A.R.C. - XEROX, Ca, July - 1979.