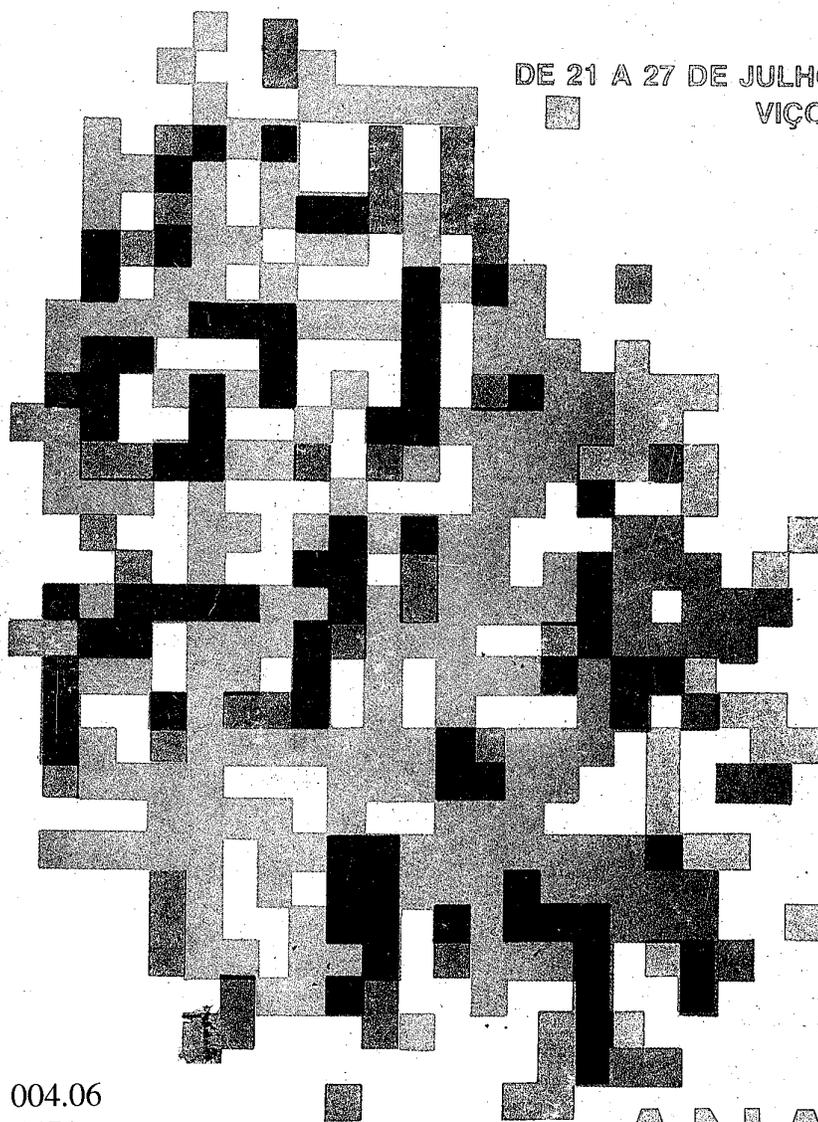


IV CONGRESSO

DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

DE 21 A 27 DE JULHO DE 1984
VIÇOSA — MG



004.06
S471
1984
V.1

ANAIIS

VOL. I

IV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO
VIÇOSA 21 a 27 de julho de 1984

ANAIS
VOLUME I

TRABALHOS APRESENTADOS
XI SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE

EDITORES:
R.S. BIGONHA, L.J. BRAGA-FILHO, A.M. OLIVEIRA & C.E. RECH

PROMOÇÃO: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO
UNIVERSIDADE FEDERAL DE VIÇOSA - UFV

PATROCÍNIO: CAPES CNPq CTI FINEP SEI

ESPECIFICAÇÃO DE UM PROTOCOLO PARA INTERLIGAÇÃO DE REDES LOCAIS

DANIEL A. MENASCE

SUMÁRIO

A interligação de redes locais ou geograficamente distribuídas, tem por objetivo aumentar o nível de partilhamento de recursos de cada uma das redes interligadas. Este artigo contém uma especificação de um protocolo para interligação de redes locais que permitam endereçamento de grupo ou global.

ABSTRACT

The interconnection of local or geographically distributed networks is motivated by the desire to increase the level of resource sharing of each component network. This paper presents a specification of a local network interconnection protocol. The protocol assumes that each local network allows for group or global addressing.

* Ph.D. em Ciência da Computação pela UCLA; Professor Associado do Departamento de Informática da PUC/RJ; redes de computadores, bancos de dados e avaliação de desempenho. Este trabalho foi parcialmente financiado pela EMBRATEL sob contrato nº CDCD-002/82.

1 - INTRODUÇÃO

A interligação de redes, locais ou geograficamente distribuídas, tem em geral os mesmos objetivos de compartilhamento de recursos que motivam a construção de redes isoladas [1]. Outras motivações podem existir no caso de redes locais com passagem de permissão, caso em que um conjunto de redes interligadas pode apresentar um desempenho melhor do que uma única rede com o mesmo número total de estações [2]. Este artigo contém uma especificação de um protocolo para interconexão de redes locais. As únicas hipóteses feitas sobre o protocolo de acesso ao meio das redes locais envolvidas é que eles permitam endereçamento de grupo ou global. O protocolo aqui descrito está sendo implementado no Laboratório de Redes da PUC/RJ para interconectar diversas REDPUCs [3 e 4].

A seção dois deste artigo descreve o contexto no qual se insere o Protocolo Interredes, bem como os serviços por ele oferecidos. A seção três contém uma descrição informal do protocolo. A sintaxe do protocolo é apresentada na seção quatro e a semântica na próxima seção. O artigo conclui com algumas considerações sobre interconexões de redes locais com redes geograficamente distribuídas.

2 - O AMBIENTE DO PROTOCOLO INTERREDES (PI) E OS SERVIÇOS OFERECIDOS

O protocolo interredes (PI), situa-se no nível 3 da arquitetura em camadas dos protocolos de uma estação de uma rede local. O nível 1 é o nível físico e o nível 2 é o nível de acesso ao meio. É o nível 2 que caracteriza o tipo de rede local: passagem de permissão (p. ex. REDPUC [2]), CSMA/CD (p. ex. ETHERNET [5]), anel de Newhall (p. ex. [6]).

As diversas redes locais são interligadas através de "gateways", que são ligados às diversas redes que são por eles interconectados. A figura 2.1 ilustra a arquitetura aqui descrita.

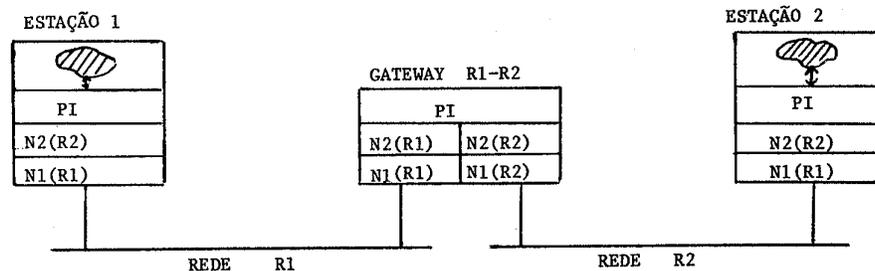


FIGURA 2.1 - ARQUITETURA INTERREDES

A figura indica que o "gateway" deve ter tantas implementações distintas dos níveis 1 e 2 quantos forem os tipos de redes diferentes aos quais eles se interliga. A interface entre o nível 3, nível interredes, e a nível de acesso N2 (R_i) em uma estação da rede local depende da rede R_i ao qual está ligada a esta estação.

Os serviços oferecidos pelo PI ao nível 4 são:

1. envio não confiável de datagramas entre duas estações que podem estar situadas em redes distintas.
2. endereçamento a nível de estação, transportando o datagrama para a estação de destino na rede de destino.
3. roteamento. A fim de tornar o roteamento eficiente o PI mantém memórias "cache" de endereços de "gateway" associados a redes de destino.

De uma maneira mais precisa, tais serviços podem ser caracterizados da seguinte forma:

```

type dtg = record (* datagrama *)
    end_dtg : integer;
    tam_dtg : integer
end;
type modulo = byte;
type rede = byte;

procedure envia_dtg_mod(rede_destino : rede;
    modulo_destino: modulo;
    tipo_protocolo: byte;
    datagrama : dtg);

```

Descrição: esta primitiva envia o datagrama armazenado em <datagrama.end dtg> e cujo tamanho é <datagrama.tam_dtg> para o modulo <modulo_destino> na rede <rede_destino>. O parâmetro <tipo_protocolo> indica qual o tipo de protocolo de nível 4 que está originando o datagrama. Isto permite a coexistência de diversos protocolos de transporte distintos oferecendo serviços diferentes para diversas classes de aplicações.

```

procedure recebe_dtg(endereço_buffer : integer;
    tam_buffer : integer ;
    tipo_protocolo : byte;
    var modulo_origem : modulo ;
    var rede_origem : rede ;
    var tam_dtg: integer;
    var cret: (tem_dtg,não_tem,buffer_pequeno));

```

Descrição : verifica se existe algum datagrama para ser recebido do nível 3 para o processo que implementa o protocolo de tipo <tipo_protocolo> . Se houver, ele deve ser ar

mazenado no buffer de endereço $\langle \text{endereço_buffer} \rangle$, de tamanho $\langle \text{tam_buffer} \rangle$. O tamanho do datagrama recebido é retornado no parâmetro $\langle \text{tam_dtg} \rangle$. Se $\langle \text{tam_dtg} \rangle$ for maior que $\langle \text{tam_buffer} \rangle$ então $\langle \text{cret} \rangle$ retorna o valor "buffer_pequeno", caso contrário retorna "tem_dtg". Se não existir datagrama a ser recebido $\langle \text{cret} \rangle$ retorna com o valor "não_tem". O parâmetro $\langle \text{modulo_origem} \rangle$ indica o módulo no qual está o processo transmissor e $\langle \text{rede_origem} \rangle$ indica a rede à qual pertence aquele módulo.

3 - DESCRIÇÃO INFORMAL DO PROTOCOLO

Considere a figura 3.1 para ilustrar o funcionamento do protocolo

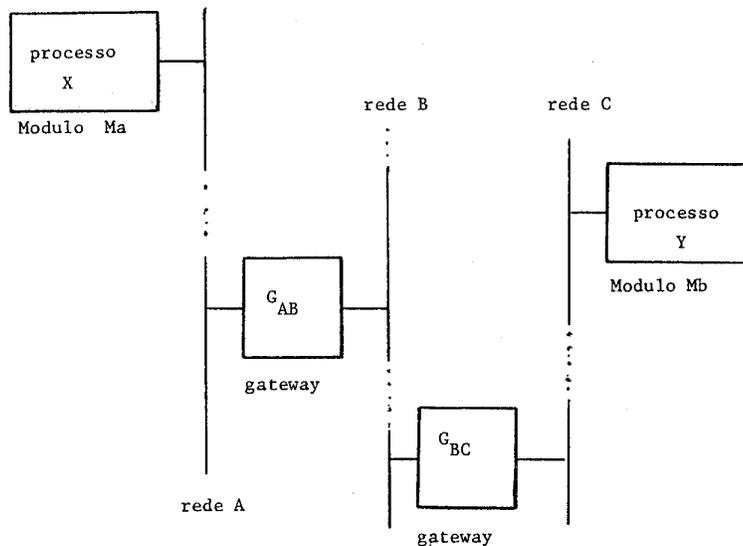


FIGURA 3.1 - EXEMPLO DE REDES INTERLIGADAS

Vamos supor que o processo X no módulo Ma da rede A resolva enviar um datagrama ao processo Y no módulo Mb da rede C. O processo X se utiliza da primitiva `envia_dtg_a_mod` com parâmetros $\langle \text{rede_destino} \rangle = C$, $\langle \text{modulo_destino} \rangle = Mb$.

Nos protocolos interredes para redes geograficamente distribuídos é comum supor que as estações ou computadores hospedeiros mantenham uma tabela completa de roteamento. Esta tabela indica neste caso para qual "gateway" deve ser enviado o datagrama. O IP da ARPANET é um exemplo disto [7]. Em redes locais, as estações podem ser microcomputadores com pouca capacidade de memória. Portanto, o protocolo aqui proposto não exige que a estação seja capaz de manter toda uma tabela de roteamento. A princípio, a estação não precisa armazenar esta tabela. Se a estação tiver capacidade de armazenamento suficiente ela pode con

ter uma "cache" de "gateways", cuja funcionamento será descrito a seguir.

O PI consulta a sua "cache" de "gateways" e verifica se existe alguma entrada para a rede C. Uma entrada nesta tabela "cache" é um par (rede_destino, gateway_de_saída). Se houver a entrada, o datagrama será enviado ao "gateway" correspondente encontrado na tabela (gateway GAB, no caso). Se não houver nenhuma entrada correspondente à rede C, o PI envia o datagrama para o grupo de módulos que são "gateways" na rede A (em geral, pode haver mais de um). Neste datagrama, entre outras coisas, haverá uma indicação da rede de destino. Assim, todos os "gateways" que receberem este datagrama na rede A consultam a sua tabela de encaminhamento entre redes. Em somente um deles haverá uma indicação de um caminho para a rede de destino. No nosso exemplo, o "gateway" GAB verifica que ele deve enviar o datagrama para o "gateway" GBC, e responder com um datagrama especial ("ack-gate") informando ser ele o responsável pela conexão com a rede C, fazendo com que o módulo Ma atualize sua "cache". O "gateway" GBC, por sua vez envia o datagrama recebido de GAB para o módulo Mb na rede C.

As "caches" utilizam uma disciplina LRU ("Least Recently Used"), para armazenar informações. Com isso, podemos garantir maior eficiência na utilização da rede, pois menos mensagens serão processadas pelos "gateways".

Em algumas situações, tais como inexistência de redes interligadas (só existe uma rede local) ou ainda em aplicações que estão circunscritas a uma única rede, não seria conveniente ter todo um processamento de nível 3, desnecessariamente. O primeiro octeto de todo datagrama interredes, é usado para indicar a existência ou não de um cabeçalho de nível 3; em caso positivo, ele indica o tipo do datagrama.

4 - SINTAXE DO PROTOCOLO

A seguir são apresentados os diversos tipos de datagramas interredes (DI) que compõem o protocolo interredes.

4.1 - DTG_MOD

Este datagrama é usado para enviar dados a um módulo (estação) de destino conforme requisitado pelo primitiva envia_dtg_a_mod.

<----- 1 octeto ----->

TIPO
REDE de ORIGEM
MODULO de ORIGEM
REDE de DESTINO
MODULO de DESTINO
TIPO de PROTOCOLO
~ DADOS ~

Os campos REDE de ORIGEM, MODULO de ORIGEM, REDE de DESTINO e MODULO de DESTINO são auto-explicativos. O campo TIPO conterá o valor 03H se a estação endereçar o DTG_MOD ao grupo de "gateways" na rede de origem, e 06H caso contrário.

O campo TIPO de PROTOCOLO identifica o tipo de protocolo de nível 4.

4.2 - ACK_GATE

Este DI é utilizado por um "gateway" para indicar a estação de origem que foi ele que aceitou um DI do tipo DTG_MOD e que, portanto, ele será o responsável pelo seu encaminhamento.

<----- 1 octeto ----->

TIPO (= 04H)
REDE de DESTINO
NUMERO do GATEWAY

4.3 DTG_NULO

Este é o datagrama nulo que indica que o nível interredes é inexistente.

<----- 1 octeto ----->

TIPO (= FFH)
~ DADOS ~

5. A SEMANTICA DO PI

Esta seção descreve em pseudo-APSL [8] a semantica do PI. A semantica do PI em uma estação da rede local é diferente da semantica no "gateway". Estas descrições estão apresentadas nas duas subseções que se seguem.

5.1 - O PI EM UMA ESTAÇÃO DA REDPUC

A especificação abaixo assume que o PI está sendo executado em uma rede de identificação REDEORG e no módulo MODORG.

espera _comandos:

```
<< + envia_dtg_a_mod(rede_dest, mod_dest, protocolo, datagrama)>>
```

```
    (DTG_MOD.rede_origem := REDEORG;
     DTG_MOD.modulo_origem := MODORG;
     DTG_MOD.rede_destino := rede_dest;
     DTG_MOD.tipo_protocolo := protocolo;
     DTG_MOD.dados := datagrama;
     consulta_cache(rede_dest, gateway, achou);
     if achou
     then begin
         DTG_MOD.tipo := 06H;
         envia_datagrama DTG_MOD para modulo <gateway>
     end
     else begin
         DTG_MOD.tipo := 03H
         envia_datagrama DTG_MOD para grupo de gateways
     end)
     → espera_comandos;
```

```
<< + ACK_GATE (rede_dest, gateway) >>
    ( coloca a entrada (rede_dest, gateway) na cache de gateways
      → espera_comandos;
```

```
<< + DTG_MOD (rede_origem, modulo_origem, rede_destino, modulo_destino, protocolos, dados)>>
    (passa dados e informações sobre origem e destino para o protocolo adequado de
     nível 4)
     → espera_comandos;
```

```

<< + recebe_dtg (endbuff,tbuff,protocolo,mod_org,rede_org,tamdtg,cret) >>
  (if não tem datagrama para "protocolo"
   then cret := não tem
   else if tamanho do datagrama > tbuff
     then cret := buffer_pequeno
     else begin
       cret := tem_dtg;
       tam_dtg := tamanho do datagrama;
       mod_org,rede_org ← conforme indicado no datagrama;
       copia dados do datagrama a partir de endbuff
     end) → espera_comandos;

```

5.2 - O PI EM UM "GATEWAY" DA REDPUC

A especificação abaixo considera a execução do PI no "gateway" G da rede R.

espera_dtg:

```

<< + DTG_MOD (rede_origem,modulo_origem,rede_destino
              modulo_destino,protocolo,dados) >>
  (if rede_destino = R
   then (* chegou à rede de destino *)
     envia DTG_MOD para modulo <modulo_destino>
   else begin (* rede intermediária *)
     consulta tabela de encaminhamento e obtem gateway de saída Gs;
     if achou (* rede na tabela *)
       then begin
         if tipo = 03H
           then begin
             envia ACK_GATE <rede_destino,G> para modulo_origem;
             DTG_MOD.tipo := 06H;
           end
         envia DTG_MOD para modulo Gs
       end
     end) → espera_dtg;
  end

```

6 - CONSIDERAÇÕES FINAIS

O protocolo interredes aqui apresentado pode ser também utilizado no caso em que redes locais e redes públicas estão sendo interligadas. Neste caso existirão em algumas das redes locais da interrede, "gateways" que de um lado apresentam uma interface com uma rede local e que de outro implementam o protocolo X.25 do CCITT para a rede pública. Maiores detalhes sobre arquiteturas de interconexão de redes locais e redes públicas podem ser encontradas em [7].

Finalmente, é necessário tecer alguns comentários sobre controle de congestão nos "gateways". Em determinadas situações, um determinado "gateway" de interrede pode vir a receber um tráfego bastante intenso. O controle de fluxo das redes locais às quais ele se liga fará com que datagramas para ele encaminhados comecem a ser descartados. É interessante nesta hora diminuir o tráfego destinado a este "gateway" sem contudo ter que tornar o protocolo mais complexo. Duas medidas estão sendo consideradas. Ambas poderão vir a ser implementadas.

A primeira consiste em fazer com que um "gateway" que descarte um datagrama comunique o seu estado de congestão ao módulo origem do datagrama. Este por sua vez, para diminuir o tráfego de datagramas para o mesmo módulo destino do datagrama descartado (e que portanto, passam pelo "gateway" congestionado), colocará no fim da sua fila de transmissão todos os datagramas para aquele destino.

A segunda medida consiste na transmissão por parte de um "gateway" congestionado G, de uma mensagem global indicando o seu congestionamento para todas as redes a ele conectadas. Ao receber tal mensagem, uma estação deve passar as mensagens destinadas ao "gateway" G para o fim da fila.

7 - REFERÊNCIAS BIBLIOGRÁFICAS

- 1 - Postel, "Internetwork Protocol Approaches", IEEE Transactions on Communications, vol. COM-28, Nº 4, Abril 1980.
- 2 - Menascê, D.A. e L. Lellis P. Leite, "Performance Evaluation of Isolated and Interconnected Token Bus Local Area Networks", a ser publicado.
- 3 - Soares Gomes, L.F. e D.A. Menascê, "Um Protocolo para Redes Locais do Tipo Difusão", Anais do IX SEMISH, Ouro Preto, Julho de 1982.
- 4 - Soares Gomes, L.F., "Projeto e Desenvolvimento de Protocolos para Redes Locais de Computadores", Tese de Doutorado, Depto. de Informática, PUC/RJ, dezembro de 1983.
- 5 - Metcalf, R. e D. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks", CACM 7(19), Janeiro de 1980.
- 6 - Farber, D.J. e K. C. Larson, "The System Architecture of the Distributed Computer System - the Communication System", Proc. of the Symposium on Computer Communications Networks and Teletraffic, 1972, pp. 21-23.
- 7 - Information Sciences Institute/USC, "Internet Protocol", IEN: 111, Agosto de 1979
- 8 - Menascê, D.A. e D. Schwabe, "Redes de Computadores: Aspectos Técnicos e Operacionais", Ed. Campus, 1984