TITULO : CONTROLE DE CONGESTIONAMENTO E FLUXO NAS CAMADAS DE REDE E TRANPORTE EM AMBIENTE OSI

AUTORES : CRISTINA KOKIEL/DANIEL DALAROSSA

ENDERECO: DIGIREDE INFORMATICA LTDA
AV. ANGELICA 2582, 4o. ANDAR, SAO PAULO, SP, CEP 01228
TEL: 011-259-1233 R. 269

CURRICULUM VITAE:

CRISTINA KOKIEL E BACHAREL EM CIENCIAS DE COMPUTACAO PELA UNIVERSIDADE DE SAO PAULO, POSSUI EXPERIENCIA NA AREA DE COMUNICACAO DE DADOS, MAIS ESPECIFICAMENTE EM CONFIGURADORES E PROTOCOLOS DE NIVEIS SUPERIORES DE REDES DE LONGA DISTANCIA.

TOTAL DE PAGINAS: 26

RESUMO:

ESTE TRABALHO TEM COMO OBJETIVO DESCREVER OS ALGORITMOS DE CONTROLE DE CONGESTIONAMENTO E FLUXO DEFINIDOS PELA DIGIREDE E PRESENTES NAS CAMADAS DE REDE E TRANSPORTE DE SUA ATUAL ARQUITETURA DE REDE DENOMINADA SISTEMA DIGIREDE DE INTERCONEXAO DE NOS (SDIN).

PALAVRAS CHAVE: MODELO OSI/ISO, CAMADA DE REDE, CAMADA DE TRANSPORTE

1. ARQUITETURA DE REDE DIGIREDE - VISAO GERAL

Desde que a automacao de servicos tornou-se uma realidade, tem surgido inumeras solucoes para diferentes problemas nas areas bancaria, industrial, comercial e de escritorio. Um novo problema que surge decorrente disso, é a interconexao entre os sistemas das diversas solucoes existentes.

A Digirede sentindo essa tendencia a nivel mundial, propos-se a desenvolver uma solucao para este problema. Esta, a nivel de rede, deveria apresentar as seguintes caracteristicas:

- Ser um sistema voltado para aplicacoes transacionais;
- Tornar baixo o custo para futuras interligacoes com outras redes que nao Digirede;
- Permitir aplicacoes independente de parametros como: topologia, localizacao fisica da aplicacao e perifericos, protocolos de comunicacao, etc.

As caracteristicas de um sistema de comunicacao que suporta de maneira adequada as necessidades acima citadas, levaram a Digirede a desenvolver o Sistema de Comunicacao Digirede (SCD), com uma proposta de arquitetura de rede baseada no modelo de referencia para Interconexao de Sistemas Abertos da ISO, denominado Sistema Digirede de Interconexao de Nós (SDIN).

1.1. Servicos e Protocolos do SDIN

Camada de Aplicacao

O servico de aplicacao oferecido pelo SDIN está dividido em:

- Servico sem conexao: adequado as aplicacoes tipicamente transacionais;
- Servico com conexao: adequado as aplicacoes que transmitem grande volume de dados sendo que controle de fluxo e sequenciamento sao caracteristicas fundamentais;
- Servico de transferencia de arquivos: uma ferramenta que facilita a transferencia de arquivos entre nós Digirede ou nao.

Os servicos previstos nesta camada sao classificados como os do "Kernel" do conjunto CASE (Common Application Service Elements) e os servicos do FTAM (ISO DP 8571).

Camada de Apresentacao

Na versao atual a camada de apresentacao é nula.

Camada de Sessao

- Servico com conexao: foi implementado o "subset" BCS com "functional units" Kernel e duplex (ISO DIS 8327)

- Servico sem conexao: protocolo baseado em ISO WD TC97/SC21 N45

Camada de Transporte

Oferece servicos com conexao e sem conexao.

- Servico Sem Conexao: utiliza o servico sem conexao da camada de rede, baseado em ISO DIS 8348 DAD1 mais um protocolo baseado em ISO DP 8602 TC97/SC6 N3223;

- Servico Com Conexao: utiliza o servico sem conexao da camada de rede, baseado em ISO DIS 8348 DAD1 mais o protocolo classe 4 da recomendacao ISO 8073.

Camada de Rede

Na camada de rede o SDIN utiliza para sub-rede Digirede o protocolo Internetwork Protocol Adaptado (IPA), baseado no Internetwort Protocol da ISO. Possui algoritmo de roteamento nao adaptativo estatico e facilidade da definicao de rotas alternativas. Modificacoes foram feitas no IP para adaptar-se as particularidades da sub-rede Digirede.

Para roteamento de mensagens para outras sub-redes, adotou-se o protocolo Internetwork Protocol (ISO DIS 8473).

Os protocolos de rede acima apresentados sao protocolos para tipo de servico datagrama, este escolhido pelo fato de ser adequado ao perfil das aplicacoes transacionais.

Camada de Enlace

Os protocolos de enlace disponiveis no SDIN sao:

- HDLC/ABM com opcoes 2, 8, 12;
- PLD (Protocolo de Linha Digirede), um HDLC/ABM na versao assincrona;
- enlace X25;
- enlace VME, um protocolo de enlace que tem como meio fisico o bus VME, de grande throughput;
- HDLC/NRM para linha multi-ponto.

2. CONTROLE DE FLUXO E CONGESTIONAMENTO - NECESSIDADES

2.1. INTRODUCAO

Nos capitulos seguintes procuraremos tratar os seguintes pontos:

- Caracterizar a necessidade de politicas de controle de fluxo/congestionamento dentro de redes comutadas por pacote e que oferecem servico tipo datagrama;
- Avaliar os atuais procedimentos existentes no SDIN voltados ao controle de fluxo/congestionamento;
- Propor novas politicas baseadas em experiencias anteriores observadas nas redes Arpanet e Cyclades.

Vamos distinguir duas expressoes que serao utilizadas neste trabalho:

- Controle de fluxo: sao procedimentos tomados entre entidades pares em nós distintos (via protocolo) ou entre entidades das camadas num mesmo nó (via interface de servico) para compatibilizar a taxa de transmissao de dados de uma entidade com a taxa de recepcao de outra entidade, tendo como beneficio direto a nao perda de dados e indireto a capacidade de se minimizar a ocorrencia de congestionamento numa placa de CPU, na maquina, na rede, etc, dependendo do ambiente em que o controle de fluxo é aplicado.
- Controle de congestionamento: sao procedimentos aplicados em conjunto com os procedimentos de controle de fluxo (utilizando-os inclusive como ferramenta) para impedir o congestionamento dos recursos relacionados acima.

Alguns termos sao caracteristicos do ambiente Digirede e tambem serao utilizados neste trabalho, sao eles:

- nob (nivel de ocupacao de buffers): é um numero no intervalo de 1 a 4 (inclusive) que reflete o nivel de ocupacao dos buffers do ambiente onde reside o software de rede.
- GCNO (gerenciador do nó): é a entidade de gerenciamento que interage com as entidades de cada camada/subcamada de forma a trocar informacoes relativas a configuracao, monitoracao/controle, determinacao de problemas/recuperacao.
- PU (Processo Usuario): é um usuario dos servicos de comunicacao que processa informacao para uma dada aplicacao.

Nó de comunicacao: um nó é um conjunto de um ou mais computadores com software associado, periféricos, terminais, etc, que forma uma entidade autonoma capaz de processar, armazenar e transferir informacao.

Um nó pode ser classificado segundo dois aspectos: de acordo com sua interligacao física com outros nós (topologia) e de acordo com uma determinada troca de informacao entre dois nós (comunicacao).

a) topologia

Um nó dependendo de sua interligacao pelo meio físico com outros nós pode ser classificado como:

1) nó de acesso: é o nó cujo meio físico o interconecta com somente um outro nó. Esse tipo de nó nao tem funcao de "relay"/"routing" implementada, isto é, ele nao tem funces que tratam de retransmissao/roteamento de mensagens.

2) Nó de comutacao: é o nó cujo meio físico o interconecta com mais de um nó. Esse tipo de nó deve ter funcoes de "relay"/"routing" implementada.

b) Comunicacao

Um nó em relacao a uma determinada comunicacao pode ser classificado como nó terminal ou nó intermediario.

Considerando uma determinada troca de informacoes entre dois nós, aqueles que estao atuando como a fonte inicial ou destino final dos dados sao chamados de nós terminais.

Nem todos os nós atuam como nós terminais, quando o meio físico nao interconecta todos os nós diretamente, alguns nós podem atuar como nós intermediarios, simplesmente passando a informacao que está sendo trocada entre dois nós terminais para outros nós (funcao de "relay").

Um determinado nó pode portanto, atuar simultaneamente como nó terminal ou como nó intermediario.

- IPADU ("Internetwork Protocol Adapted Data Unit"):

É a unidade de dados do protocolo IP Adaptado que reside na subcamada 3/1 do SDIN.

2.2. MOTIVACAO

Dentro das modalidades de comutacao atualmente empregadas em redes de computadores, a comutacao de pacotes é a que recebe maior destaque, exatamente pela sua habilidade em poder compartilhar dinamicamente os recursos da rede entre seus usuarios, procurando nao reservar uma porcao significativa de recursos da rede a nenhum usuario em particular.

Dentro dessa modalidade, redes "puramente datagrama" (Digirede) sao as que mais se prejudicam em relacao às redes "puramente circuito-virtual" quando da elevada carga de pacotes na rede, em outras palavras, o controle de recursos de uma rede datagrama em situacao de carga é muito mais complexo (e portanto requer atencao especial) do que numa rede circuito-virtual. Essa caracteristica é certamente levada em consideracao quando da escolha do tipo de servico a ser oferecido numa rede publica de transmissao de dados (vide Renpac, Transpac, Tymnet). Na verdade a justificativa que normalmente se dá para a adocao da filosofia de circuito-virtual numa rede publica é que a qualidade do servico que a rede se compromete a oferecer aos usuarios finais fica muito mais garantida e muito mais facil de se obter, em oposicao á filosofia datagrama.

Resumindo: redes tipo datagrama sao por natureza mais suscetiveis a falhas em situacoes de carga do que redes tipo circuito virtual e essas falhas podem se refletir de forma significativa na qualidade do servico oferecido aos usuarios se nenhum procedimento de controle de congestionamento é tomado.

Uma descricao do que pode acontecer quando nao há nenhum controle de congestionamento na rede é dada a seguir: numa situacao de carga nos nós de comutacao, o "round trip time" dos pacotes na rede (tempo de "passeio" dos pacotes na rede) aumenta, isso faz com que:

- na camada de transporte que oferece servico com conexao comecem a surgir os "falsos alarmes" (time-out) e a retransmissao de pacotes supostamente perdidos é iniciada;

- o elemento humano defronte a um terminal, apos uma mensagem de "falha na comunicacao" relativa á sua solicitacao de servico transacional repita a operacao, na esperanca de obter sua resposta.

Em ambos os casos (servico com conexao e sem conexao) o numero de copias de pacotes já em transito na rede tende a aumentar, isso implica em que:

a) o throughput é reduzido a uma fracao do normal;

b) conexoes sao desfeitas devido ao "round trip time" (RTT) ter aumentado bruscamente; vale observar que mesmo com os melhores algoritmos adaptativos de retransmissao na camada de transporte, uma variacao subita na carga da rede pode causar uma demora na convergencia do RTT atual para o real, causando as desconexoes;

c) a alocacao de recursos na rede entre usuarios que competem entre si torna-se pobre, permitindo desigualdade na distribuicao dos recursos;

d) o throughput pode cair de niveis reduzidos para zero, e em alguns casos uma situacao de "deadlock" pode se observar.

O que desejamos a nivel de proposta de controle de congestionamento envolvendo camadas de rede e transporte é que:

- nunca haja descarte de mensagens por congestionamento;

- o "throughput" da rede permaneca em niveis toleraveis.

2.3. SITUACAO ATUAL

Serao descritos nos itens seguintes quais sao os procedimentos atualmente previstos nas camadas de rede e transporte do SDIN relativos a controle de fluxo e controle de congestionamento.

2.3.1. Camada de Rede

Na versao 1.0 do SDIN está implementado na camada de rede um protocolo de controle de congestionamento.

A premissa basica do protocolo é que a cada alteracao de nob no ambiente em que a camada de rede reside, esta deve avisar os nos vizinhos de tal alteracao; quando se recebe um aviso de alteracao de nob, deve-se passar essa informacao ao processo GCNO.

Concluimos entao que o protocolo passa toda a responsabilidade do controle da geracao de fluxo do nó ao processo GCNO, isto é, este deve de alguma forma "avisar" os geradores de fluxo (gerenciadores de aplicativos) para que "diminuam" seu fluxo para um dado destino. Essa filosofia traz as seguintes questoes á discussao:

a) Será que é realmente funcao do processo GCNO esse tipo de tarefa?

b) Podemos garantir que os algoritmos e procedimentos adotados em cada gerenciador (PU) vao corresponder ás expectativas do sistema de comunicacao, no sentido de manter o throughput no nivel maximo sem congestionar o(s) nó(s) que se manifestou(aram) critico(s)?

c) Nao seria mais eficiente o proprio SDIN tomar a atitude de "estrangular" o fluxo para um dado destino ao inves do GCNO, para evitar o atraso e assincronismo entre avisar os gerenciadores e o procedimento ser efetivamente iniciado? e o que fazer com os pacotes que já estao dentro do SDIN em vias de ser transmitidos?

2.3.2. Camada de Transporte

Aqui um aspecto a ser analisado é o algoritmo de retransmissao de pacotes utilizado pelo protocolo de transporte classe 4. Na versao 1.0 a temporizacao Ti do protocolo permanece estatica na conexao, isso traz os seguintes inconvenientes:

a) No caso de um congestionamento temporario na rede, os "falsos alarmes" tendem a aparecer, isto é, a temporizacao Ti estoura indicando perda do pacote (quando na verdade ele ainda está na rede) e uma retransmissao é efetuada; se o congestionamento permanecer durante o intervalo N * Ti, a conexao será liberada;

b) O fato de o RTT nao ser obtido dinamicamente, faz com que na situacao descrita em a) a camada de transporte seja um elemento colaborador no agravamento do congestionamento, pois coloca pacotes na rede (duplicados) desnecessariamente; isso afeta inclusive o throughput das conexoes.

Com relacao ao controle da janela de recepcao, parece razoavel a ideia de diminuir e posteriormente fechar a janela quando nob aumenta: W = n para nob = 1, W = n/2 para nob = 2, W = 0 para nob >= 3, sendo n parametro de configuracao. Talvez um refinamento que se poderia pensar seria em se aumentar o numero de fases do nob (por exemplo 1 a 6), e fazer que o fechamento de janela fosse menos brusco, como a seguir:

nob	% sobre W
1	100
2	80
3	50
4	20
5	0
6	0

Um outro aspecto negativo da versao é o fato de sempre se transmitir AK TPDU a cada DT TPDU recebido (V = 1); isso inunda a rede com pacotes se existem varias conexoes estabelecidas entre os varios nos de acesso; colocando de outra forma: embora consigamos melhor "performance" para V = 1, sabemos que em situacao de carga da rede fazer V = 1 é altamente indesejavel, pois o "overhead" trazido pelo protocolo (transmissao de AK TPDUs) é significativo.

3. PROPOSICAO PARA A CAMADA DE REDE

Dentre as varias politicas de controle de congestionamento existentes, destacamos duas que se mostraram bastante eficientes em suas aplicacoes:

- "choke packets" da rede Cyclades [5];

- "source quench" de rede Arpanet.

a) "Choke packets"

Consiste basicamente em monitorar o tamanho das filas associadas aos enlaces disponiveis para a camada de rede, de forma que em se atingindo um determinado limite (por exemplo 80% da capacidade maxima) continuar encaminhando os pacotes de dados que chegam (pelo enlace em situacao critica) mas ao mesmo tempo gerar um pacote "choke" (estrangulador) ao originador do pacote de dados; essa informacao pode ser passada à camada de transporte no nó terminal que pode entao regular seu fluxo de transmissao de dados.

b) "Source Quench"

A ideia aqui é semelhante á exposta em a), só que ao inves de monitorar as filas dos enlaces é monitorado o nivel de ocupacao de buffers do nó como um todo; caso o nob atinja um nivel critico (por exemplo nob >= 3) um pacote de "source quench" (extincao da fonte) é enviado para cada novo pacote de dados que é encaminhado; tal informacao tambem é passada á camada de transporte.

Os procedimentos existentes na camada de rede da versao 1.0 utilizam, sob um certo aspecto, as tecnicas de "source quench"e "input buffer limit"; a principal discussao é a forma como se bloqueia os geradores de fluxo, daí a necessidade de uma nova proposicao.

Passamos entao a propor um protocolo de controle de congestionamento que substitui o comentado em 2.3.1.

3.1. Procedimento

Serao descritos a seguir procedimentos relativos á geracao do CC IPADU (veja 3.2) somente.

A entidade de rede atuará de forma a monitorar o tamanho das filas dos enlaces e o nivel de ocupacao de buffers do nó onde reside.

Nao haverá distincao a respeito do tipo de nó a utilizar os procedimentos, isto é, nó de acesso e comutacao tomarao as mesmas atitudes.

O procedimento a ser tomado é:

a) Quando se recebe um NPDU de dados do IP adaptado de uma entidade de rede par e nob é igual ou maior que 3 entao p1 é realizado.

b) Se sobre o PDU de dados tiver de ser aplicado as funcoes de encaminnamento (isto é, o destino do PDU nao é este nó) é analisado o tamanno atual da fila do enlace que tal PDU eventualmente vá ser colocado; se tal tamanho corresponder a pelo menos k% (k é configurado) do tamanno maximo da fila (que atualmente é configurado) e nob for menor que 3, p2 é realizado.

c) Quando se recebe a primitiva SN_UNIDATArequest [4], para cada PDU de dados que vier a ser obtido (mais de um se segmentacao for feita), b) é realizado, mas sem ser levado em consideracao a restricao do nob e ao inves de p2 é realizado p3.

Acoes mencionadas no procedimento:

p1 - É enviado um IPADU de controle de congestionamento (veja 3.2) com campo DG = 0, DA = source address do IPADU de dados, TA = dest address do IPADU de dados. Tal IPADU terá prioridade sobre quaisquer outros IPADUS no sentido de ser encaminnado por um dado enlace antes que qualquer outro IPADU (eventualmente enfileirado para ser encaminnado pelo mesmo enlace).

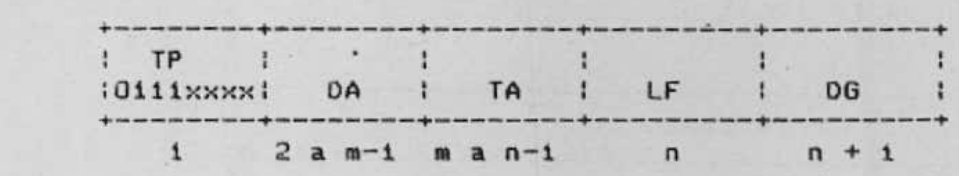
A geracao só será feita se o campo "segment offset" do header do IPADU de dados correspondente for igual a zero, ou seja, tal pacote é o primeiro segmento do pacote original.

p2 - mesma acao p1, com campo DG = 1

p3 - É gerada a primitiva descrita em 3.3, com causa = rota congestionada.

3.2. Estrutura e codificacao do IPADU de controle de congestionamento

A estrutura do CC IPADU (Controle de Congestionamento) é mostrada a seguir:



TP : Type protocol, identifica IPADU de controle de congestionamento (bits de 8 a 5); os bits 4 a 1 sao reservados.

DA : Destination Address, é o endereco do originador do IPADU de dados, ao qual este CC IPADU deve ser encaminhado.

TA : Target Address, é o endereco que constava como destino no IPADU de dados.

LF : Lifetime, descrito em [1].

DG : Diagnostic, contem o motivo da geracao deste IPADU, como a seguir:

0 : nob critico
1 : rota congestionada

3.3. Servico adicional

O protocolo descrito aqui terá reflexo na interface de servico 3/1 (-) 3/3 e 3/3 -) 4 sob a forma da primitiva N_L_CONGEST indication, descrita a seguir (SN_L_CONGEST indication é analoga a N_L_CONGEST indication):

Primitiva / parametro	N_L_CONGEST indication
end. destino	x
diagnostico	x

"end destino": é o endereco do nó que havia sido passado como parametro "end destino" na primitiva N_UNIDATA request anteriormente gerada.

"diagnostico": indica o motivo da geracao desta primitiva; as possibilidades para este parametro sao:

- congestionamento em nó da rede.

- congestionamento em rota da rede.

Algumas observacoes a respeito de 3.1, 3.2, 3.3:

a) Da forma como foi colocado em 3.1., o seguinte problema tem grande probabilidade de ocorrer: se um mesmo nó terminal está gerando IPADUs de dados com frequencia para um mesmo destino, basta um CC IPADU para esse nó ao inves de um para cada pacote de dados: se o nó que gera tais CC IPADUs estava com nob critico, esse tipo de situacao colabora ainda mais para degradar a situacao do nó.

b) Para evitar possivel "starvation" dos IPADUs normais (como deixa transparecer a acao 3.1.p1), podemos transmitir um IPADU normal a cada n CC IPADUs transmitidos (n configurado).

c) Para deixar os procedimentos tomados pelas entidades em toda a rede mais estaveis, podemos aplicar uma histerese ao tamanho k definido em 3.1.b.

d) A razao da escolha em se definir a primitiva N_L_CONGEST indication ao inves de utilizar a N_REPORT_indication é:

- a primitiva N_REPORT_indication é gerada só quando a correspondente primitiva N_UNITDATA request é descartada na rede, ao contrario do que N_L_CONGEST_indication informa.

e) Com relacao á interface com o processo GCNO, parece conveniente reportar ao GCNO os CC IPADUs recebidos.

3.4. Solucao Alternativa

Uma outra possibilidade que podemos apontar para o controle de congestionamento é aquela que utiliza o proprio IPADU de dados do IPA para levar informacao sobre o estado da rede.

Em linhas gerais o que se pretende é:

a) Adicionariamos mais um campo no IPADU de dados do IPA, chamado "carga da rede", de tamanho 1 byte.

b) Para todo IPADU de dados em transito na rede seria adotado o seguinte procedimento: se o nob do nó que detem tal PDU é critico ou o enlace que tal PDU será encaminhado está critico e essas informacoes mapeadas num numero (por exemplo 0 a 100) é maior que o valor do campo "carga da rede" do IPADU, entao atribui-se o valor registrado no nó ao referido campo. Dessa forma saberiamos, quando da entrega desse pacote ao destino, como está a situacao da rede, pelo menos no que refere á rota que tal PDU percorreu.

c) Na primitiva "N_UNITDATA_indication" gerada no nó destino a partir do IPADU de dados referido em b), seria adicionado o campo "carga da rede", que é o simples repasse do valor contido no campo analogo do IPADU de dados.

Esse tipo de informacao seria util ao transporte pois:

d) A cada TPDU recebido da entidade par seria verificado o estado da rede e poder-se-ia ajustar (inclusive de forma bastante refinada) o tamanho da janela de transmissao da entidade par, até mesmo fechando a janela, caso necessario.

Vamos analisar agora vantagens e desvantagens da solucao anterior que utiliza CC IPADU (chamaremos de S1) em relacao á solucao agora apontada (chamaremos de S2).

Vantagens de S1 em relacao a S2:

a) A deteccao de congestionamento na rede é sensivelmente mais rapida em S1 que em S2. Suponha que haja uma conexao entre dois nós de acesso A e B estabelecida sem que haja troca de dados; se no intervalo entre os AK TPDUS (por W [3]) de A para B um nó de comutacao ficar congestionado (o que nao é pouco provavel pois W é da ordem de minutos) e se A iniciar repentinamente uma transferencia de dados, entao pode haver descarte de mensagens em tal nó de comutacao até que B saiba do fato e feche a janela para A.

Conclusao: existe uma inercia grande para se detectar congestionamento, ou seja, para a fonte geradora (que é o alvo de nossas atencoes neste trabalho) ficar ciente em S2.

Algumas consideracoes:

1) Mesmo em S1 quando A iniciasse a transmissao repentina, é bem provavel que o CC nao chegaria a tempo para bloquear a transmissao sem haver descarte no nó de comutacao. O que queremos dizer é que mesmo em S1 pode haver descarte;

2) o grande problema de S2 (para o caso de descarte) é que nos baseamos no proprio pacote de dados do IPA para levar a informacao de congestionamento e dessa forma, havendo descarte, A nao receberia um AK IPDU fechando a janela e haveria entao retransmissao por A, podendo haver até mesmo quebra da conexao AB. Em S1 isso nao acontece pois mesmo havendo descarte, A seria bloqueado.

   Fica claro entao que a diferenca de S1 para S2 neste caso é que em S1 bloqueia-se a retransmissao muito mais rapidamente que em S2.

b) Com S1 poderiamos passar a informacao de congestionamento tambem para o servico sem conexao: com S1 os proprios recursos de comunicacao sao suficientes para passar a informaco de congestionamento á fonte (no servico sem conexao) enquanto que em S2 deve haver um protocolo entre PUs para bloquear a fonte.

g) Quando da interconexao de redes, se há transmissao de dados entre um nó de subrede Digirede para um nó de outra subrede, em caso de problemas num nó da subrede Digirede, S1 tende a resolver o problema.

Desvantagens de S1 em relacao á S2:

h) Um problema grave de S1 é o fato de se consumir recursos da rede para se bloquear a fonte, isto é, na geracao do CC IPADU utilizam-se buffers, CPUs, enlaces, enquanto que em S2 quase nenhum recurso adicional é requerido.

i) Quando da interconexao de redes, se há transmissao de dados entre um nó de outra subrede para um nó da subrede Digirede, em caso de problemas em um nó da subrede Digirede, S2 tende a resolver o problema.

j) Sob condicoes normais, S2 nos permite fazer um controle mais refinado do fluxo de dados; S1 funciona á base do "XON/XOFF".

k) O procedimento de retomada da transmissao de dados por parte da fonte é mais simples e efetivo no caso S2 que em S1. Uma possivel solucao seria: apos ter fechado a janela da fonte, esperariamos os AK TPDUs (por W) gerados pela fonte; a partir daí conseguiríamos saber o estado da rede e eventualmente iniciar abertura de janela da fonte.

l) A nivel de complexidade de implementacao dos procedimentos S1 e S2, podemos afirmar que S2 é sem duvida bem menos complexo que S1; o custo de implementacao de S2 é praticamente zero.

3.5. Solucao Desejada

Apos termos descrito algumas alternativas para controle de congestionamento envolvendo camadas de rede e transporte, vamos sugerir aqui o uso da melhor solucao. A melhor solucao deverá ser simples (no sentido de implementacao, compreendimento), efetiva (no sentido de atingir os objetivos comentados em 2.2) e que nao consuma recursos de rede em demasia para seu funcionamento.

Colocado esses pontos verificamos que ambas as solucoes (S1 e S2) entram em conflito com tais caracteristicas.

Solucao / Caracteristicas	S1	S2
simples	nao	sim
efetiva	sempre	nem sempre
consumo de recursos de rede	grande	muito pequeno

Tendo em vista que nem S1 ou S2 satisfazem todos os requisitos, sugerimos a utilizacao de ambas as solucoes, abrindo mao do requisito "simplicidade".

porque da adocao do modelo misto:

- S2 sozinha nao deve ser adotada pois já vimos que nao é efetiva em todos os casos

- Para minimizar o problema do "consumo dos recursos de rede", teriamos que ajustar o valor comentado em 3.1.b (K) para alguma coisa em torno de 90%. de modo que em situacoes normais, isto é, sem cargas de fluxo repentinas de varios nós, o proprio controle previsto em S2 resolveria, e em ultimo caso S1 seria acionado.

Observacao:

Poderiamos acrescentar um campo "user data" na estrutura da CC IPADU para saber qual entidade de transporte deve ser ativada. Mas, se um CC IPADU destina-se ao servico sem conexao (por exemplo), o servico com conexao nao seria avisado, pois a geracao do CC é seletiva; assim, optou-se por gerar a primitiva N_L_CONGEST indication para todas as entidades de transporte ativas.

3.6. Rotas Alternativas

Até agora nos mantivemos afastados da discussao sobre rotas alternativas. Quando nos itens anteriores nos referimos a por exemplo "quando o numero de mensagens na fila de um enlace atingir k% da capacidade, devemos enviar CC IPADU", estavamos omitindo a possibilidade de uso de enlace alternativo. Isso é estranho pois vai contra a atual filosofia de uso do enlace alternativo: utiliza-se o enlace alternativo quando da indisponibilidade do enlace principal. Entenda-se indisponibilidade como queda fisica do enlace ou fila associada ao enlace cheia.

Concluimos entao que em amoas as solucoes apontadas (S1 e S2) devemos, antes de tomar alguma atitude relativa á indicacao de congestionamento num dado enlace, verificar se há enlace alternativo; o status de congestionamento seria coletado do ultimo enlace alternativo conseguido.

Dessa forma a filosofia que seguimos com relacao á enlaces alternativos é a de distribuicao de carga, ao contrario da utilizacao de enlaces alternativos como "backup" (só usar em caso de indisponibilidade dos enlaces principais).

4. PROPOSICAO PARA A CAMADA DE TRANSPORTE

As alteracoes que sugerimos aqui estao ligadas ao protocolo definido em 3. e as otimizacoes que julgamos necessarias.

4.1.2.1. Utilizacao do N_CONGEST_indication (NCOGind)

O protocolo que preve o servico com conexao de transporte deveria utilizar a primitiva NCOGind da seguinte forma:

Na recepcao de um NCOGind sao localizadas todas as conexoes ativas cuja entidade de transporte par reside no endereco recebido em NCOGind; para cada conexao a atitude a ser tomada é reduzir a janela de transmissao a zero e passar a contar os AK TPDUs recebidos; quando o numero de AK TPDUs recebidos atingir n (configurado), assume-se o tamanho da janela antigo, ou seja, a situacao anterior ao recebimento do NCOGind.

Observacoes:

- é garantido o recebimento dos n AK TPDUs pelo temporizador W do protocolo de transporte.
- no sentido contrario, isto é, a recepcao de dados continua normalmente; com a aplicacao do procedimento 4.1.2.2 os AKs transmitidos devido á recepcao de DT TPDUs nao deverao sobrecarregar ainda mais o nó que causou a geracao do NCOGind.

O protocolo que prove o servico sem conexao de transporte deveria ignorar os NCOGind, pois vamos assumir por simplicidade que o servico transacional nao sobrecarrega a rede com solicitacoes (na verdade essa é uma hipotese que na pratica se verifica).

4.1.2.2. "Ack/Credit group size V"

Este procedimento consiste em transmitir um AK TPDU apos um dos seguintes eventos ter ocorrido:

a) V DT TPDUs foram recebidos.

b) Nao foram recebidos V DT TPDUs e se passou um tempo AL (mesma notacao do IS 8073) apos a recepcao do primeiro DT TPDU (este por sua vez recebido apos o envio do ultimo AK TPDU).

O procedimento consiste em se temporizar AL [3] apos o recebimento do primeiro DT TPDU e somente cancelar tal temporizacao apos a transmissao do AK TPDU (quando ocorrer um dos eventos mencionados em a ou b).

O objetivo deste procedimento é reduzir o "overhead" de protocolo na transmissao de AK TPDUs.

Uma sugestao é fazer V = 1 para conexoes em situacao normal de transferencia de dados, e conforme o aumento do trafego na rede incrementamos o valor de V, porem observando sempre a condicao de que o V seja menor ou igual ao valor da janela de transmissao da sua entidade par.

Alem do objetivo citado acima, conseguimos com V = n (n>1) sobrecarregar menos a rede com pacotes do transporte.

4.1.2.3. Temporizador T1 dinamico

A proposicao a ser feita é baseada em resultados obtidos no TCP da rede Arpanet [2].

A ideia aqui é fazer com que o temporizador retransmissao de pacotes (T1) seja adaptativo, isto é, seja reavaliado e alterado conforme haja variacoes na carga da rede ou quando a entidade de transporte par resida em outra rede. Exemplos que cobrem essas duas situacoes sao citadas a seguir:

a) É uma situacao comum em redes de computadores as flutuacoes que ocorrem na carga da rede e que afetam diretamente o RTT da rede. Essas flutuacoes estao ligadas intimamente ao ambiente de aplicacao que utiliza os servicos de comunicacao. Um exemplo disso é de uma rede que dá suporte a aplicacao bancaria; sabemos que um horario de pico na rede é das 11:00 ás 12:00 hs, onde, devem ser confirmadas as aplicacoes em over-night, fundo de renda fixa, etc. Quando o protocolo nao preve esse fato (T1 fixo, caso SDIN v1.0), observamos retransmissoes desnecessarias e ás vezes até quebra de conexoes.

b) Suponha que uma rede local seja conectada á rede SDIN, cujo RTT medio é da ordem de 500 ms. Uma entidade de transporte dessa rede que deseja estabelecer uma conexao com uma entidade par do SDIN se nao "souber reconhecer com quem está estabelecendo a conexao" e mantiver T1 fixo certamente nao conseguirá seu objetivo.

O procedimento a ser aplicado é descrito a seguir:

Para cada DT TPDU transferido deve-se medir o RTT (tempo entre a transmissao de um DT TPDU até o recebimento do AK TPDU). Baseando-se nas medidas pode-se computar o RTT medio (Smoothed Round Trip Time SRTT).

$$SRTT_i = A.SRTT_{i-1} + (1-A).RTT$$

onde i é o numero da mensagem transmitida.

Tendo o SRTT, calcula-se o Retransmission Time Out (RTO).

$$RTO = \min(Ubound, \max(Lbound, B.SRTT))$$

onde Ubound e Lbound sao os valores maximo e minimo permitido para cada conexao.

A é o fator de media e

B é o fator de variancia cujos valores recomendados sao 0,8 ~ 0,9 e 1,3 ~ 2,0 respectivamente

Repare que o valor de Lbound deve ser Lbound = Elr + Erl + Ar + X segundo a notacao [3].

Existem alguns problemas no uso do Ti:

A primeira dificuldade é na escolha do valor inicial do SRTT. Antes de qualquer troca de mensagem entre duas entidades de transporte nao há informacao de quanto será o valor do RTT, supondo que a topologia nao é connecida pelas duas entidades de transporte.

É muito frequente ocorrer casos em que a escolha arbitraria desse valor é muito menor ou maior comparado com o valor real. O mesmo acontece com o valor de RTO. Como resultado, haverá retransmissoes desnecessarias ou haverá esperas longas antes da retransmissao se o primeiro pacote for perdido, alem disso a convergencia será lenta.

Quando o valor inicial é muito pequeno, retransmissoes excessivas podem causar congestionamento temporario na rede antes que o Ti convirja para o valor correto. Por outro lado um valor inicial muito grande significa uma possivel resposta lenta para o usuario mas nao causa nenhum tipo de prejuizo para a rede como no caso anterior.

É ilustrado um exemplo da velocidade da convergencia quando valor do SRTT inicial (S0) nao é apropriado.

Escolhendo A = 0,85, B = 2 e assumindo S0 = 3 segundos e todos os valores RTT medidos = 1 segundo, sem perdas de pacotes:

$$SRTT_n = A^n \cdot S0 + RTT(1-A^n)$$

$$SRTT_{19} = 1,1 \text{ seg}$$

Por outro lado escolhendo S0 = 1 seg e todos os RTT = 3seg sem perda.

$$SRTT_{19} = 2,9 \text{ seg}$$

Para uma conexao com poucos pacotes a serem transferidos a convergencia nao é possivel.

O segundo problema é como medir RTT. A medicao é trivial quando nao há perdas de pacotes. Quando ocorre perda de pacotes a medicao correta de RTT torna-se impossivel porque quando um AK TPDU é recebido apos n retransmissoes, o transmissor de dados nao pode afirmar qual das n + 1 copias transmitidas está sendo reconhecida. Esse fato afeta diretamente o calculo do SRTT.

Os exemplos abaixo podem esclarecer melhor o problema acima citado.

Assumindo que um AK TPDU foi recebido apos n retransmissoes:

1. Supondo que o RTT é tomado como o tempo decorrido desde o envio da 1a. copia do pacote até finalmente receber o AK TPDU, isto é, o tempo medido para detecao da perda (n x RTO) mais o tempo real de RTT e usando esse valor para calcular SRTT e depois RTO, temos:

   RTO = min (Ubound, max (B.SRTT, Lbound))

   $$RTT_{i+1} = n.RTO + real\_RTT = n.B.SRTT_i^{*} + real\_RTT$$

   $$SRTT_{i+1} = A.SRTT_i + (1-A).RTT_{i+1}$$

   $$= (A.SRTT_i + (1-A)real\_RTT) + (1-A).n.B\_SRTT$$

   parte desejada | parte indesejada

   Portanto, a perda de apenas um pacote pode causar uma grande variacao no valor de SRTT e perdas multiplas e consecutivas podem fazer com que os valores de RTO e SRTT crescam até o valor de Ubound.

   Desde que os valores de Ubound e Lbound sejam fixos e escolhidos respectivamente grande e pequeno para que $B.SRTT_S$ se situe entre os seus limites.

2. Se o RTT for medido a partir do envio da ultima copia até a recepcao do AK TPDU, o resultado vai ser menor que o valor real de RTT desde que o AK TPDU recebido corresponda a uma copia transmitida anteriormente, a ultima copia. O SRTT vai convergir para um valor errado.

3. Se a medida da RTT nao for usada para ajustar SRTT quando ocorre retransmissao, o valor de SRTT nao mudará.

   Se o valor original do RTO é menor que o RTT real ou o atraso na rede crescer repentinamente o RTO vai estacionar sempre no mesmo valor que é pequeno, resultando em retransmissoes desnecessarias.

Considerando os exemplos acima pode-se afirmar que nos casos 2 e 3 haverá sempre retransmissoes que sao altamente indesejaveis, enquanto que no caso 1 isso nem sempre ocorre e quando ocorrer há possibilidade para convergir para o novo valor correto.

Existe outro fator significativo que influencia no calculo do SRTT: é o tamanho do pacote a ser transmitido.

Segundo [3] o Ti é calculado da seguinte maneira:

Ti = Elr + Erl + Ar + X

onde Elr: atraso maximo esperado de local p/ remoto

Erl: atraso maximo de remoto p/ local

Ar : tempo de ACK remoto

X : tempo de processamento

Os valores de Elr, Erl e Ar sao fixos e de conhecimento de ambos os lados de comunicacao e sao calculados baseados em TPDUs de 128 bytes. O atraso Elr pode aumentar drasticamente quando há mensagens grandes a serem transmitidas (digamos 4 Kbytes) nesse caso o Elr pode ser até 32 vezes maior que o valor original.

A sugestao é calcular o SRTT por segmentos de 128 bytes. Desse modo pode se calcular o RTO levando em consideracao o tamanho do pacote.

Outro ponto a ser discutido é quando há envio para a rede de uma serie de pacotes que perfazem uma janela e o V da entidade par remota é maior que 1. Quando chega um AK TPDU da entidade remota confirmando n mensagens anteriormente enviadas, como proceder para calcular o SRTT?. Isso é particularmente problematico se o valor de V for grande.

Uma solucao seria temporizar todas as mensagens enviadas e quando receber um AK TPDU, o RTT é o tempo decorrido deste o envio da 1a. mensagem com temporizacao ativada até a chegada do ACK. Deve-se cancelar todas as temporizacoes de pacotes anteriores ao pacote confirmado pelo AK TPDU.

Na solucao apresentada o calculo do RTT independe do valor de V remoto. Depende apenas do tempo decorrido entre envio de um pacote até recepcao de um AK TPDU valido, isto é a propria definicao de Ti.

É necessario criar uma tabela para guardar todas as temporizacoes cujo tamanho será de no. total de conexoes x tamanho da janela de cada conexao. Esse tamanho nao é fixo pois o tamanho da janela varia conforme as condicoes de trafego na rede e do nod da entidade de transporte par.

Para contornar esse problema sugerimos o seguinte:

Manter no maximo duas temporizacoes ativas por conexao. Toda vez que se enviar um pacote e houver algum recurso disponivel (ie uma temporizacao nao ativa) deve-se ativar a temporizacao. No pior caso conseguiriamos medir apenas um RTT por janela de pacotes enviados.

5. REFERENCIAS

[1] ISO DIS 8473 - Internetwork Protocol

[2] Transmission Control Protocol
No. 4, october 1980, pg. 52-132

[3] ISO IS 8073
Transport Protocol Specification

[4] ISO TC97/SC16/N 3453
Addendum to DIS 8473 covering Provision of the
Connectionless mode Subnetwork Service

[5] The Cyclades Computer Network
Louis Pouzin
North - Holland Publishing Company