

SUMÁRIO

Os protocolos serão introduzidos de acordo com as primeiras formas de que se revestiu a utilização de computadores à distância. Apresenta-se uma sequência de evolução que vai até às redes de computadores, no âmbito das

quais descrevem a forma como o software de comunicações está organizada, sendo feita uma análise e crítica à sua estrutura.

De seguida procuramos caracterizar os protocolos do ponto de vista dos objectivos a que se destinam e das funções que devem executar por forma a alcançarem as finalidades requeridas. Faz-se ainda antes de terminar uma breve análise ao desempenho dos protocolos do ponto de vista da sua estabilidade e eficiência de funcionamento.

1. - INTRODUÇÃO HISTÓRICA

Uma característica muito importante da situação da computação nos anos 60 era a centralização dos meios de processamento e armazenamento de informação. Razões económicas de peso obrigavam as empresas e demais entidades utilizadoras a adaptar-se a estrutura centralizada de processamento de forma a garantir a rentabilidade dos investimentos, através de elevados níveis de utilização dos equipamentos.

No entanto, este estado de coisas é contrário a forma como as actividades humanas se organizam o que originou todo um cortejo de problemas que tem sido largamente discutidos.

É neste ambiente que começam a surgir tentativas de utilização do computador à distância, como forma de mitigar os problemas criados pela centralização. Assim surgem os primeiros terminais e também as primeiras estações de trabalho que permitem o desejado acesso descentralizado aos serviços do sistema central.

Apesar de se ter introduzido um novo regime de utilização da máquina, nada se fez para a adaptar a esta situação. A nível do sistema central construiu-se um controlador cablado que geria os aparelhos remotos e que para o sistema central se comportava como um periférico conhecido (por exemplo uma unidade de banda). O programa de aplicação de Teleprocessamento corria num inalterado sistema de operação orientado para processamento por lotes.

Ao nível do sistema central não houve evolução significativa, até ao aparecimento nos anos 70 das arquitecturas de comunicação dos fabricantes, nomeadamente a SNA/IBM e DNA/DEC para referir apenas as mais conhecidas.

Por outro lado, ao nível da comunicação entre o controlador central e os aparelhos remotos, aí sim houve bastante evolução. Indicam-se como passos dessa evolução a separação do controle de aparelhos das funções de comunicação, a possibilidade de operar em linhas privadas ou também comutadas e ainda a possibilidade de operar linhas multi ponto. Este último melhoramento referido é introduzido apenas pelo conhecido BSC/IBN. Já nos anos 70 aparece uma nova família de protocolos de linha que superam as deficiências dos anteriores protocolos, incluindo o BSC, norma ISO NFZ66010 etc. de que são exemplos o HDLC/ISO, SDLC/IBM, DDCMP/DEC, etc.

Em termos de arquitectura de comunicações foi no entanto o empreendimento ARPAnet o marco mais significativo na história desta evolução. Utilizando a já bem estabelecida técnica de construção do software numa hierarquia de camadas constrói-se software de comunicações que vai permitir por máquinas heterogéneas a cooperar para realizar trabalho comum. Este empreendimento influenciou decisivamente tudo o que se seguiu incluindo as arquitecturas dos fabricantes, que apareceram alguns anos depois. Um outro aspecto relevante a referir é o da queda de custos dos equipamentos. Um mais baixo preço possibilitou a proliferação de sistemas de informação. Com a fase seguinte aparece a

necessidade de fazer traduzir em ligações directas entre esses sistemas de informação os laços existentes entre as organizações humanas em que estão inseridos. O empreendimento Arpanet já mostrou que isso é possível e as situações concretas de necessidade existem o que como corolário leva ao desenvolvimento de sistemas como a SNA/IBM, DNA/DEC, modelo provisório ISO em desenvolvimento ainda, etc.

2. - SITUAÇÃO ACTUAL

2.1. - Estruturação hierárquica do software de comunicações

Os sistemas de software de comunicação são bastante complexos e é nomeadamente muito importante garantir que sejam de fácil manutenção e operem com fiabilidade. Estas características, em particular a de manutenção, são Indispensáveis num meio em que a mudança é uma constante. É assim importante a possibilidade de substituir módulos sem interferir com o resto do sistema.

Razões deste tipo levaram ao desenvolvimento do software de comunicações como um conjunto de camadas, cada qual com funções específicas. Outro aspecto também conducente a uma maior fiabilidade e manutenibilidade é o da separação em diferentes camadas da responsabilidade pela gestão dos diferentes recursos (capacidade de canais, espaço de endereços, etc.), que esta abordagem torna possível.

Esta hierarquia de camadas é descritível como uma estrutura Imbricada de protocolos. Nesta fase poderíamos dizer que um protocolo é um conjunto de regras que permite que dois ou mais processos troquem ordenadamente informação entre si. De acordo com este esquema conceptual cada camada é traduzida pela implementação de dois ou mais correspondentes que obedecendo às regras do protocolo trocam informações. As camadas Interiores em regra são Implementações de outros protocolos que têm por finalidade prestar serviços à camada imediatamente superior. Se nos colocamos ao nível do utilizador podemos projectar protocolos para esse contexto de utilização, como sejam por exemplo protocolos de gráficos, de transferência de ficheiros, teleconferência, etc. Esses protocolos terão necessidades de serviços a assegurar pelas camadas inferiores. Por exemplo o protocolo de teleconferência precisa de transferir informação dentro de condicionalismos rígidos de atraso o qual deve ser praticamente constante e muito pequeno; já o protocolo de transferência de ficheiros tem necessidades diferentes, não fazendo exigências quanto ao atraso de transmissão mas antes necessitando de alta capacidade de transferência com taxa de erro muito baixa. Assim, de acordo com as necessidades do protocolo do nível utilizador projectaríamos aquele que deverá prestar esse serviço. Prosseguindo recursivamente num processo deste tipo acabaríamos sobre a linha física e as convenções a nível de sinais eléctricos que permitem o transporte físico da Informação.

De facto a situação não é assim tão simples, dado que muitas vezes uma certa camada deve ser projectada para servir diferentes protocolos na camada superior. Por exemplo os protocolos de gráficos e de transferência de ficheiros podem ambos recorrer a um protocolo de terminal virtual, numa implementação particular. Assim cada camada muitas vezes deve poder ser utilizada com múltiplas finalidades. Um outro aspecto é que um dado sistema pode hospedar num dado momento um elevado numero de correspondentes diferentes. Isto terá, nalguns níveis, como consequência, a necessidade de multiplexagem e de gestão dessas ligações diferentes dado que existe um numero limitado de linhas, pontos de entrada e saída para o sistema e ainda restrições relativamente ao espaço de endereços globais (a nível da rede).

2.2. - Modelo geral de hierarquia

Existem diferentes abordagens ao projecto do software de comunicações. Por exemplo a arquitectura SNA/IBM baseia-se em critérios que se podem dizer de tipo funcional para fazer a divisão em camadas e decidir sobre as funções de cada. De acordo com este critério evita-se ao máximo a duplicação de funções e sempre que aparecem funções comuns em duas camadas adjacentes decide-se a criação de mais uma camada que ficará entre as duas e que as assegurará aquelas funções que deixarão de ser executadas por qualquer uma das chamadas Iniciais.

Um outro critério, utilizado pelo DNA/DEC, é o de criar uma camada da sempre que haja "quebras" de endereçamento. De acordo com este critério haverá uma camada para gerir cada nível numa hierarquia de endereços. Na DNA/DEC há três níveis de endereçamento: nível linha (estações numa linha multiponto), nível rede (nos individuais da rede) e o nível "host" (processos individuais alojados nos sistemas hospedeiros).

Ainda um outro critério, independente dos dois anteriores, reparte as camadas de acordo com as interfaces existentes, como seja por exemplo a x25, CCITT

Face ao que se disse poderíamos ser levados a crer que existe uma grande variedade de arquitecturas, mas isso é mais aparente do que possa parecer à primeira vista. Efectivamente apesar de, uns dividirem mais do que outros tem-se chegado quase sempre ao mesmo tipo de estrutura.

Um modelo mais ou menos geral de arquitectura será o que compreende as seguintes cinco camadas: nível físico, nível linha, nível transporte, nível "end-to-end" e nível aplicação. O nível físico cria um caminho físico para a transmissão de informação; trata de problemas como sejam as características eléctricas e de controle de equipamentos de interface, como por exemplo modems. Este nível usa linhas, feixes hertzianos, satélites, etc. e cria uma interface através da qual os aparelhos periféricos do computador se ligam a estes meios trocando com eles blocos de informação. Algumas normas aplicáveis a este nível são a X21/CCITT, RS 232/EIA, RS 422/EIA, etc.

O nível linha cria um canal sequencial de dados com uma taxa de erros muito baixa apesar de estar baseado num suporte com ruído significativo. Deve resolver os problemas de detecção e correcção de erros binários, sincronização de blocos transmitidos e recebidos e deve ainda controlar a duplicação ou falta de blocos de dados. No caso de canais multiponto descreve o controle de acesso ao canal. Exemplos de protocolos do nível linha: ADCCP/ANSI.

HDL/ISO, SDLC/IBM, DDCMP/DEC, etc.... O nível de transporte encaminha mensagens entre nós terminais em termos da rede de transportes de dados. Este nível pode ou não deixar o controle de fiabilidade para o nível superior. Por exemplo no caso da rede cyclades há possibilidade de perda ou de duplicação de informação. A razão é simples: Dado que o nível do transporte não tem sido possível garantir 100% de fiabilidade, esse tipo de controles acabam por se traduzir em duplicação de funções o que poderá ser evitado deixando-as exclusivamente para o nível superior. O nível end-to-end gere os circuitos virtuais utilizador a utilizador encaminhando a informação dentro dos nós aos processos de aplicação e proporcionando nomeadamente o controle sobre a duplicação ou perda de informação.

Por último o nível de aplicação é o que gere as trocas entre os utilizadores directamente: Como exemplos de protocolos a este nível referimos o Telnet (terminal virtual), FTP (file transfer protocol), NGP (network graphics protocol), NVP (network voice protocol) todos da Arpanet, DAP da Decnet, etc.

2.3. - Critica

As críticas que se podem fazer ao modelo de arquitectura atrás descrita baseiam-se essencialmente nos seguintes aspectos: insuficiências, complexidade e duplicação de funções.

A insuficiência tem que ver com os tipos de serviços proporcionados por uma arquitectura particular. Existem polémicas várias relacionadas com este aspecto, como por exemplo, circuitos virtuais versus datagramas. Nas quais não vamos entrar. Para exemplificar este aspecto da insuficiência consideremos dois protocolos: de teleconferencia e de transferencia de ficheiros.

Já referimos que têm necessidades de transporte de tal diversidade que será difícil uma arquitectura de comunicações compatibilizar objectivos tão contraditórios. Desta forma seria de concluir que para tornar o problema o melhor seria projectar uma arquitectura com fim específico e que seria obviamente

mais simples.

A enorme complexidade do software de comunicações pode ter como efeito uma sobrecarga a nível de processamento e de transporte de dados. De facto o cruzamento das numerosas interfaces entre as varias camadas têm como resultado uma sobrecarga de processamento. Por outro lado, dada a grande adição de cabeçalhos as mensagens e de mensagens de controle necessárias ao funcionamento dos protocolos, ha também uma sobrecarga do suporte de transmissão de dados. A redução do numero de camadas teria como resultado a perda das vantagens de fiabilidade, manutenibilidade e adaptação ao ambiente de mudança. Pelo contrario o aumento agrava as sobrecargas referidas. Um aspecto que resulta particularmente negativo será que os pequenos sistemas dificilmente poderão suportar a sobrecarga de processamento que implica a sua ligação a uma rede com este tipo de arquitectura. Se imaginarmos duas pequenas empresas que desejem trocar em linha vulgares ordens de pagamento e encomendas, utilizando para isso uma arquitectura de comunicações do tipo do descrito, vemos que isso não pode ser por provocar uma exagerada carga de processamento nesses pequenos sistemas em geral dada a sua natureza de sistemas de monoprogramação nem necessitariam de grande número das facilidades oferecidas, como seja por exemplo a multiplexagem de varias ligações num endereço único. É claro que os tais pequenos utilizadores poderão através de cartões, diskettes, bandas, etc, transportar os dados debaixo do braço...

Finalmente relativamente à duplicação de funções refira-se que ela tem que ver com aquilo que e designado como pureza dos protocolos, ou seja com o tacto (e o protocolo não necessitar (se for puro) de analisar informação referente a cabeçalhos que não sejam os seus. Cada protocolo existe apenas ao nível dos correspondentes que regula, considerados como pares colocados ao mesmo nível da hierarquia. Outro aspecto que denota este tipo de característica e o de transparência. De acordo com este conceito o protocolo não necessita analisar ou conhecer a estrutura daqueles em que se apoia, bem como deve funcionar sem ter de conhecer o estado da maquinaria hard/soft em que se apoia. Toda a informação necessária a cada camada é fornecida através da interface com a camada inferior. Desta forma existe uma duplicação que muitas vezes não seria necessária se as funções das duas camadas fossem combinadas numa só.

Digamos em conclusão que este tipo de arquitectura tem muita generalidade, que resultou de anos de experiência, e que, é a base dos sistemas actualmente existentes e de alguns que se projectam para o futuro, como por exemplo as normas ISO em estudo. Mais do que críticas, os aspectos atrás apontados devem ser considerados como discussão sobre os limites de utilização e características inerentes do modelo de arquitectura.

3. CARACTERÍSTICAS GERAIS DOS PROTOCOLOS

3.1. - O que são protocolos

Os protocolos são regras que permitem a elementos, correspondentes entre si e colocado no mesmo nível da hierarquia de camadas software de comunicações, trocar informação. Estes elementos estão distribuídos geograficamente em locais distantes entre si e devem ser capazes de comunicar apesar de não conhecerem o estado actual do seu correspondente, mas apenas o seu estado de algum tempo. Conceptualmente podemos dizer que estes elementos constituem uma maquina distribuída cujos componentes devem ser mantidos sincronizados. As mensagens que os elementos trocam entre si são produzidas e reconhecidas de acordo com as regras do protocolo. Podemos assim dizer que cada autómato correspondente para dialogar com o seu par necessita de um par de sub-autómatos, um gerador e um reconhecedor de mensagens. Note-se a analogia que é possível estabelecer com as linguagens de programação (as mensagens são produções do gerador que o reconhecedor aceita ou rejeita). No caso de rejeição e em regra necessário um procedimento, descrito pelas regras do protocolo, que permita aos correspondentes colocarem-se num contexto de execução mutuamente aceite.

Para terminar este aspecto da caracterização retira-se que para conseguir a transferencia e necessário que cada par de correspondentes não necessite (nem deva) conhecer o estado, estrutura ou suporte tecnológico da camada inferior de protocolo que se encontra atrás (ou por baixo, como se queira) da interface através da qual comunica com essa camada.

3.2. - Necessidade dos Protocolos

os protocolos destinam-se a estabelecer (a) elementos de dados normalizados, (b) convenções de comunicação e finalmente (c) a definição da via de comunicação.

(a) - Elementos de dados normalizados: englobamos aqui todas as regras que tem por fim descrever um tipo de dados virtual em termos do qual os correspondentes vão falar. Este aspecto e particularmente importante podendo dizer-se que para muitos casos o problema do projecto do protocolo se resume ao desenho desta estrutura de dados. Em que é que se caracteriza esta estrutura ? Damos dois exemplos: A forma canónica intermédia de ficheiro num FTP e o ficheiro de desenho (virtual display file) num protocolo de gráficos. Este problema tem complexidade muito variável, desde o trivial no nível linha, ao que ainda não esta desenvolvido satisfatoriamente como por exemplo a forma canónica num FTP.

(b) - Convenções de comunicação: Englobamos aqui todas as convenções que têm que ver com condições de temporização, códigos, métodos de multiplexar dados, controlo e de um modo geral a estruturação de mensagens a trocar dum ponto de vista de codificação. Este ultimo aspecto particularmente importante tem que ver com a forma pela qual são implementados os elementos de dados normalizados referidos em (a). Este processo é análogo ao que nas linguagens de programação se utiliza para fazer a implementação duma estrutura de dados através da atribuição dum significado específico a bites ,ou grupos de bites ou palavras da memória central da máquina.

(c) - Definição da via de comunicação: Este objectivo especifica o interesse de, através de um certo número de procedimentos de controle, criar uma via de comunicação virtual com certas características desejáveis que o suporte em que se apoia não possui. Consoante o tipo de protocolo assim teremos um tipo específico de melhoria. Por exemplo o nível linha reduz a taxa de erro e adapta as velocidades de emissão e de recepção enquanto que um transporte apoiado num serviço de datagramas poderia procurar garantir a sequencialidade na via virtual de comunicação.

I

3.3. - Funções dos Protocolos

Para assegurar os objectivos atrás referidos os protocolos devem genericamente assegurar um certo numero de funções. Nalguns protocolos algumas das funções, aqui descritas, poderão ser praticamente inexistentes. Indicamos de seguida as funções que consideramos mais importantes.

(a) - Controle de erros

O controle de erros tem por fim a detecção e correcção de erros introduzidos pelas camadas inferiores em termos da estrutura imbricada. O tipo de erros a considerar depende do nível do protocolo na hierarquia. Genericamente ha erros de conteúdo, de entrega e de sequência. O erro de conteúdo refere-se a entrega de dados alterados no receptor. O erro na entrega refere-se à entrega de dados em duplicado, como se fossem dados novos, ou perda de dados no suporte de transmissão. Finalmente o erro na sequência refere-se à entrega de dados na recepção por ordem diferente daquela pela qual foram emitidos. Depende do contexto de utilização a importância de cada um dos tipos de erros referidos.

Genericamente os erros de conteúdo são detectados com o auxilio de CRC's (cyclic redundancy check) e de "checksums". Os erros na entrega e de sequencialidade são detectados através da identificação, com números em sequência das mensagens.

Embora haja variantes e muito se possa dizer a respeito da correcção (e também a respeito dos tipos de erros e sua detecção) indicamos apenas e muito sucintamente que os erros de conteúdo e perda na entrega são resolvidos através de pedidos de retransmissão da informação concernente. Quanto à duplicação na entrega a correcção é inerente à detecção.

Finalmente quanto à sequencialidade obriga-se á retransmissão em sequência de tudo o que já tenha sido transmitido desde uma detecção de quebra de sequência. Note-se que por exemplo uma rede de datagramas não transporta sequencialidade e mesmo nas redes de comutação de pacote; e necessário uma sobrecarga de processamento e transmissão para que se possa recorrer a circuitos virtuais que gozam da propriedade da sequencialidade.

(b) - Controle de Fluxo

Deve haver um controle de fluxo que regule o tráfego de mensagens de acordo com as capacidades de absorção dos receptores. Mas mais globalmente o controle de fluxo, em conexão com o encaminhamento, deve evitar a congestão da rede. No subsistema de comunicações da rede de computadores é muito fácil ocorrer a congestão de totalidade ou de parte da rede havendo apenas controle de fluxo nó a nó. Mesmo sem chegar ao máximo de mensagens que a rede pode ter em transporte em simultâneo poderá haver interbloqueio entre todos ou alguns dos nós da rede.

O controle de fluxo pode ser uniforme, se for indistintamente aplicado a dados de controle, ou selectivo se for aplicado somente a certas categorias de mensagens, normalmente mensagens de dados.

Como técnicas de controle de fluxo referimos a transmissão alternada e a autorização de transmissão antecipada.

A transmissão alternada é muito simples de implementar. De acordo com este esquema a numeração binária das mensagens é suficiente e só pode haver uma mensagem em trânsito de cada vez.

A transmissão antecipada utiliza um mecanismo de janela que dá a identidade das mensagens autorizadas a seguir. O receptor indica explicitamente a extensão da janela (crédito) ao emissor que só pode transmitir as mensagens com números dentro do intervalo descrito pela janela. A janela é movida de acordo com as indicações do receptor. Ao fornecer um crédito o receptor pede ter ou não reservado efectivamente o espaço de memória correspondente (abordagens segura e optimista, respectivamente).

As escolhas que se façam a este nível é um importante impacto no desempenho da maquinaria de comunicações. Na verdade as escolhas que se fazem relativamente aos mecanismos de controle de fluxo são motivadas fundamentalmente por questões de desempenho nomeadamente aproveitamento das capacidades de transmissão, dimensão de memórias tampão, nível de retransmissões, tempos de resposta etc...

(c) - Endereçamento

Os protocolos destinam-se a permitir que processos hospedados em sistemas heterogéneos possam falar, recorrendo nomeadamente a um tipo de dados virtual independente da estrutura desses sistemas. Assim não seria de esperar que os correspondentes ao iniciar uma ligação conheçam a estrutura de endereços dentro da máquina situada remotamente. Estamos pois a evidenciar a necessidade de um processo normalizado de endereçar processos a nível da rede. Os nomes dos processos a nível da rede (network-wide names) devem pois ter uma estrutura fixa que permita a qualquer processo identificar outro.

Existem várias abordagens à formação destes nomes bem como de estabelecer as ligações no âmbito das quais os protocolos vão operar. Vamos resumidamente apresentar as duas de maior interesse: atribuição dinâmica e atribuição fixa.

No âmbito do primeiro esquema referido os processos dirigem-se em cada sistema, a um processo específico (um gestor de endereços e ligações) o qual acorda ou cria o processo objecto, dá-lhe um nome de rede e fornece esse nome ao iniciador da ligação. Note-se que de acordo com esta abordagem o processo iniciador deve apesar de tudo ter um conhecimento não trivial do sistema hospedeiro remoto. Esta abordagem foi seguida na Arpenet.

Na rede cyclades por exemplo seguiu-se uma outra abordagem, a atribuição fixa, a qual procura superar o problema referido no caso anterior. Os nomes têm uma estrutura fixa independente das entidades locais tal como na atribuição dinâmica. Mas aqui estes nomes estão associados fixamente a processos que são acedidos a nível da rede. O objectivo deste tipo de abordagem é a obtenção duma rede de recursos, esbatendo as fronteiras entre sistemas. A atribuição dinâmica ao contrário dá ênfase ao conceito de rede de computadores, ainda que haja sempre subjacente a ideia de partilha de recursos.

(d) - Encaminhamento

Existe normalmente uma camada específica que tem por função básica o encaminhamento (routing) das mensagens, na direcção do seu destino final. É claro que este problema só tem sentido em redes de uma certa dimensão.

Tem havido grande proliferação de técnicas de encaminhamento. Uma hipótese seria haver um esquema fixo de encaminhamento, o qual obviamente por não se adaptar às mudanças de topologia e padrão de tráfego na rede é inadequado. De facto é indispensável que o algoritmo de encaminhamento preveja a adaptação tanto quanto possível sincronizadamente, às mudanças que ao longo do tempo se vão operando na rede.

Esquemas como este último, conhecidos como adaptáveis, são os predominantemente utilizados e dele têm proliferado inúmeras variantes.

É em especial nas situações anormais que é necessário que o mecanismo de encaminhamento funcione adequadamente.

Referiríamos a terminar que três abordagens frequentes nos esquemas de encaminhamento adaptáveis são o isolado, centralizado e o distribuído.

Cada um daqueles esquemas tem níveis de desempenho próprios, robustez própria etc. A escolha do método de encaminhamento deve pois obedecer a critérios de rendimento e de estabilidade.

(e) - Segurança

Será desejável, em especial, nos protocolos de alto nível, poder garantir que as mensagens trocadas não poderão ser conhecidas por entidades não autorizadas. Trata-se de uma função indispensável em certas actividades, como por exemplo a actividade bancária. O problema é que a rede de comunicação é um meio inseguro, onde não é possível garantir que as mensagens não podem ser interceptadas ou alteradas com intenção criminosa. Tem sido feitas várias abordagens e propostas inúmeras soluções na imensa quantidade de material que sobre isto tem sido publicado. Citamos duas soluções, a criptografia simétrica e a não simétrica. A primeira abordagem baseia-se na utilização da mesma chave em ambos os terminais ao passo que a segunda recorre a duas chaves diferentes, uma para cifrar e outra para decifrar. Esta última abordagem tem sido a melhor aceite dado que se confere melhores níveis de segurança, a par da possibilidade de autenticação das mensagens. A autenticação refere-se à garantia de que uma dada mensagem foi transmitida efectivamente por uma determinada entidade.

(f) - Gestão de Recursos

Desejamos aqui referir-mos à função que todos os protocolos tem de assegurar a gestão de recursos próprios ou da rede, com o fim de assegurar o seu próprio funcionamento.

Devem ser geridos os recursos que são atribuídos ou de que é verificada a possibilidade de partilha no acto de autorização de acesso de um utilizador. Por utilizador desejamos referir uma certa camada que utiliza as interiores em termos de estrutura imbricada. Cada protocolo deve assegurar a administração do suporte de transmissão em que se apoia e dos recursos próprios do seu nível como sejam por exemplo, memórias tampão, ciclo de nomes de mensagens etc...

Muito haveria a dizer sobre as técnicas de gestão desses vários recursos, o que apesar de tudo, sai fora do escopo do presente texto.

(g) - Codificação

Todos os protocolos asseguram uma certa forma de codificação das estruturas de dados virtuais com que operam. Cada tipo de protocolo fala numa linguagem diferente (bites, bytes, caracteres, blocos, mensagens, ficheiros, 'display files', "jobs" etc). De acordo com a definição da estrutura é necessário encontrar um processo de codificação dessa estrutura de forma a que fique numa forma adequada ao transporte pelo respectivo suporte. Um problema a ter em conta em particular no nível linha e o da transparência de dados.

(h) - Negociação

Alguns protocolos, em especial de nível utilizador, tem previstos mecanismos de negociação através dos quais os processos correspondentes podem adquirir um melhor conhecimento das características em que o par se suporta, permitindo assim que o dialogo e desenrole da melhor forma entre ambos.

O mecanismo de negociação varia desde a quase inexistência ate formas bastante complexas. Assim nalguns casos alem de características hardware a negociação pode envolver a descrição e forma de executar certos procedimentos, bem como a indicação de quem os executa.

Os mecanismos de negociação historicamente apareceram nos protocolos de terminais onde continuam a ser utilizados, mas, foram estendidos a outros protocolos em particular os protocolos assimétricos, Isto é aqueles em que existe uma relação mestre-escravo ou dito noutros termos utilizador-servidor.

3.4. - Desempenho dos Protocolos

O desempenho de um protocolo depende de factores Intrínsecos como por exemplo, valores porque se opta, disciplinas de procedimento etc. os quais têm implicações sobre a forma como o protocolo opera. O desempenho depende ainda de factores exteriores a definição do protocolo como por exemplo a natureza do suporte de transmissão em que se apoia.

A qualidade do desempenho do protocolo deve ser avaliada a luz do fim a que se destina. De acordo com essa finalidade as opções tomadas num caso serão completamente inadequadas noutros casos.

É vulgar atender aos seguintes aspectos ,a avaliação da qualidade de desempenho: Atraso de transmissão (delay), fluxo de mensagens (throughput) e qualidade de serviço no que diz respeito a aspectos como maior disponibilidade, menor taxa de erro, em conteúdo e na entrega, segurança etc.

É inerentemente difícil satisfazer cada um desses aspectos sem descuar os outros. Assim um bom tempo de resposta leva a uma diminuição do fluxo e do nível do serviço, dado que exige mensagens curtas e baixo nível de utilização do suporte. Por outro lado um fluxo elevado exige filas de espera associadas a grande utilização dos suportes , mensagens grandes para diminuir a sobrecarga de controle etc. Finalmente a melhor qualidade de serviço implica uma sobrecarga de controle, tanto nos cabeçalhos das mensagens como numa adição de mensagens de controle o que obviamente prejudica simultaneamente o tempo de resposta e o volume de trafego útil transportável. Através do aumento da capacidade de transmissão do suporte melhoram-se estes três aspectos simultaneamente. No entanto nem sempre é possível tecnicamente. Dum ponto de vista económico é mesmo geralmente indesejável. Assim o estudo do desempenho apresenta um particular interesse por poder levar à escolha duma opção de compromisso que sem aumentar custos leve a um nível de qualidade aceitável para os vários aspectos referidos. Quais são os factores intrínsecos que afectam a eficiência do desempenho ? São vários, mas vamos referimos brevemente aos seguintes que consideramos relativamente relevantes. Assim por exemplo a dimensão das mensagens é um desses factores. A sua dimensão tem que ver com a memória tampão disponível, nível de retransmissões etc. Assim mensagens grandes podem pela necessidade de retransmissões conduzir a uma situação de penúria de memória bloqueando a taxa útil de transferencia de dados. Mas mensagens pequenas provocar, um adicional de carga de controle o que tem como efeito baixar o referido fluxo útil. Há obviamente um valor de compromisso. A dimensão das memórias tampão também tem efeitos evidentes no desempenho já que penúria de memória, exige que a mesma seja

utilizada com um ciclo menor o que provocara eventualmente descontinuidades no fluir da informação. Por outro lado os valores do atraso de retransmissão (retransmission time-out) e do atraso de confirmação (round trip delay) em conexão com as políticas de confirmação (ACK's) podem ditar uma utilização óptima (fluxo útil máximo) do suporte ou pura e simplesmente através - de escolhas inadequadas, sobrecarregar indevidamente o suporte com numerosas mensagens de controle. Os tempos têm que ver nomeadamente com o tempo de vida máximo das mensagens no sistema de transporte e também estão relacionados com a dimensão das memórias tampão.

Um último aspecto que convém referir relativamente ao desempenho e o da estabilidade. É desejável que os protocolos sejam insensíveis às variações das características do meio em que estão inseridos, mesmo se essas variações sejam apreciáveis. Por exemplo a taxa de erro é um parâmetro que expectavelmente variará. É desejável que o desempenho sofra variações mínimas num largo espectro de taxas de erros possíveis. É claro que apesar de tudo há limites a partir dos quais a degradação é drástica.

Face a este ambiente o protocolo deve incluir regras para sintonizar os atrasos de retransmissão e de confirmação, o crédito a conceder etc.

Um outro aspecto onde é desejável garantir estabilidade é relativamente a previsível variação no padrão de tráfego, isto é, tráfego com diferentes tamanhos de mensagens, diferentes níveis de fluxo, tráfego com corrente etc...

4 - CONCLUSÃO

O problema da comunicação entre muitas máquinas diferentes cria com uma acuidade renovada a necessidade de uma linguagem que reduza o esforço de implementação e manutenção de sistemas de software que lhes permita comunicarem entre si. Se tivermos N sistemas diferentes que necessitem comunicar entre si, teremos de fazer a implementação de N² sistemas de comunicação, ao passo que a utilização dum protocolo que normalize as regras de comunicação levará a que haja apenas N implementações. Esta diminuição de trabalho, bem como outras vantagens que tem sido apontadas conferem importância especial ao problema dos protocolos.

Os protocolos vieram evidenciar novos problemas. De facto no ambiente dum rede existe assincronia, ruído, paralelismo não controlado e mútua suspeita entre os vários correspondentes o que os coloca numa situação ímpar. Para enfrentar um ambiente de tal forma adverso os protocolos têm sido sucessivamente melhorados. Nos últimos dez anos, em especial, a evolução, esforço de, investigação e desenvolvimento neste campo tem sido impressionante conforme o demonstra o enorme volume de material editado sobre este assunto.