10. TCP/IP

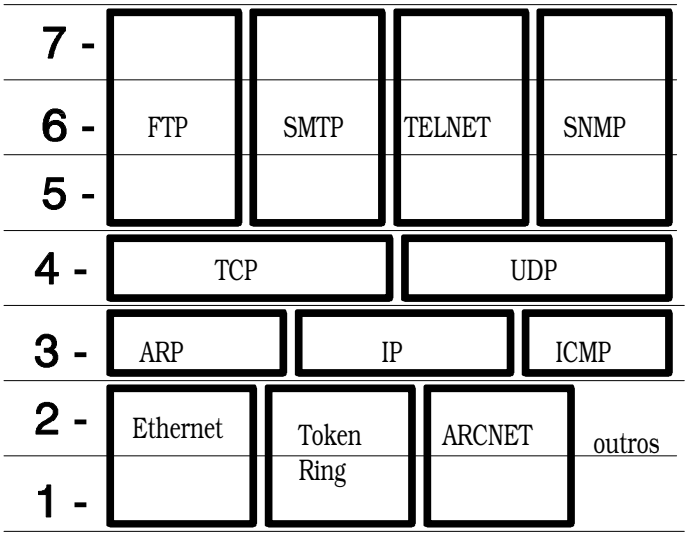


Figura 10.1 - Protocolos TCP/IP

Fruto direto do desenvolvimento de protocolos para redes de longa distância na década de 70, os protocolos TCP/IP se tornaram sinônimo de interoperabilidade. Sob a tutela do Departamento de Defesa Americano, as universidades e organizações governamentais se ligaram através da rede ARPANET. O mecanismo de definição de protocolos é a publicação de documentos pedindo comentários sobre a proposta (RFCs - Request For Comments). Os documentos mais importantes estão indicados a seguir (onde OBR indica obrigatório, REC é recomendado e OPC é opcional):

Protocolo	**Nome**	**Status**	**RFC**
----------	Protocolos Oficiais IAB	OBR	1360
----------	Números Definidos	OBR	1340
----------	Requisitos dos Nós - Comunicação	OBR	1122
----------	Requisitos dos Nós - Aplicações	OBR	1123
----------	Requisitos dos Roteadores	OBR	1009
IP	Protocolo Interredes	OBR	791
ICMP	Protocolo de Mensagens de Controle Interrede	OBR	792
IGMP	Protocolo de Transmissão de Grupo	REC	1112
UDP	Protocolo de Datagrama de Usuário	REC	768
TCP	Protocolo de Controle de Transmissão	REC	793
TELNET	Protocolo TELNET	REC	854,855
FTP	Protocolo de Transferência de Arquivos	REC	959
SMTP	Protocolo Símples de Transferência de Correio	REC	821
MAIL	Formato de Mensagens Eletrônicas	REC	822
CONTENT	Campo de Tipo de Conteúdo	REC	1049
NTP	Protocolo de Horário de Rede	REC	1119
DOMAIN	Sistema de Nomes por Domínios	REC	1034,1035
DNS-MX	Roteamento de Correio com Domínios	REC	974

Protocolo	Nome	Status	RFC
SNMP	Protocolo Símples de Gerenciamento de Redes	REC	1157
SMI	Estrutura de Gerenciamento de Informação	REC	1155
MIB-II	Base de Informações de Gerenciamento-II	REC	1213
EGP	Protocolo de Roteadores Externos	REC	904
NetBIOS	Protocolos de Serviços NetBIOS	OPC	1001,1002
ECHO	Protocolo de Eco	REC	862
DISCARD	Protocolo de Discarte	OPC	863
CHARGEN	Protocolo de Geração de Caracteres	OPC	864
QUOTE	Protocolo de Citação do Dia	OPC	865
USERS	Protocolo de Usuários Ativos	OPC	866
DAYTIME	Protocolo de Hora e Data	OPC	867
TIME	Protocolo do Servidor de Hora	OPC	868
TFTP	Protocolo Trivial de Transferência de Arquivos	OPC	1350
RIP	Protocolo de Informações de Roteamento	OPC	1058

10.1. Address Resolution Protocol (ARP)

Muitos outros protocolos de mais baixo nível estão definidos, além de outros que ainda estão em estudo ou são de carater experimental. Um desses protocolos de baixo nível é o de resolução de endereços (ARP - RFC 826) que está indicado no nível rede na figura 10.1. Cada nó de um conjunto de redes interligadas recebe um endereço IP diferente de 32 bits. Dado um endereço IP de uma estação na mesma rede que quem quer enviar a mensagem, é o ARP quem descobre o endereço físico a ser usado. Cada nó mantém uma pequena tabela com a correspondencia entre endereços físicos e IP. Se o endereço desejado estiver na tabela, já está resolvido o problema. Caso não esteja, é transmitido um pacote de "broadcast" no formato ARP e o pacote original é descartado (para simplificar, já que os níveis mais alto devem ter lógica de retransmissão). A tabela é atualizada quando voltar alguma resposta, de modo que a próxima tentativa de transmitir para aquela estação terá sucesso. Assim, uma estação só mantém informações sobre os nós com que efetivamente se comunica, o que reduz o espaço de memória ocupado. É claro que existem mecanismos para eliminar informações que fiquem obsoletas.

10.2. Internet Protocol (IP)

O protocolo fundamental deste tipo de rede é o IP. Os endereços IP são de 32 bits e normalmente são representados como quatro números decimais separados por pontos. Por exemplo, o endereço hexadecimal 8F6B036Dh é indicado por 143.107.3.109. Este endereço é dividido em duas partes: o número da rede e o número do nó. Quantos bits são dedicados a cada parte depende do tipo do endereço:

```
                    1         2         3
          01234567890123456789012345678901
Tipo A    0 rede  |           nó
Tipo B    10      rede      |     nó
Tipo C    110           rede        |   nó
```

A lógica deste tipo de organização é que existem poucas redes com um número muito grande de nós, mas muitas redes pequenas. Reservamos endereços com 0s para indicar valores desconhecidos. Enquanto 0.0.0.26 indica a estação 26 de uma rede ignorada, 143.107.0.0 indica uma máquina

desconhecida na rede 143.107 (também é uma maneira conveniente para designar a rede como um todo). Um endereço com todos os bit em 1 indica “broadcast”. Assim, para enviar para todos os nós da rede 143.107 podemos usar o endereço 143.107.255.255. Na realidade, como o “broadcast” é limitado a um única rede, o endereço 255.255.255.255 poderia ser usado por um nó da rede 143.107 com exatamente o mesmo resultado.

Os seguintes campos fazem parte do cabeçalho do PDU do protocolo IP, conforme mostrado na figura 10.2. **Versão** (4 bits) - confirma que o transmissor e receptor estão usando a mesma versão do software. A versão atual é a 4. **IHL** (4 bits) - comprimento do cabeçalho em palavras de 32 bits. **Tipo de Serviço** (8 bits) - indica a qualidade de serviço exigida pelo pacote. **Comprimento Total** (16 bits) - indica o comprimento em bytes do datagrama. I**ndentificação** (16 bits) - usada na montagem de pacotes fragmentados. **Flags** (3 bits) - o primeiro é sempre zero, o segundo indica se a fragmentação é permitida ou não e o último indica se existem mais fragmentos depois deste ou não. **Deslocamento do Fragmento** (13 bits) - indica onde este fragmento se encaixa no pacote original em múltiplos de 64 bits. **Tempo de Vida** (TTL - 8 bits) - indica quantos segundos de vida este pacote tem antes de ser discartado. Isto impede que um pacote fique preso indefinidamente em algum "loop". **Protocolo** (8 bits) - qual protocolo de mais alto nível deve receber este pacote. **Checagem do Cabeçalho** (16 bits) - confirma a integridade do pacote. **Endereço de Origem** (32 Bits) - quem está enviando este pacote. **Endereço de Destino** (32 bits) - onde este pacote está tentando chegar. Existe, também, um campo de opções de segurança e roteamento. É interessante comparar este formato com o usado no ISO 8473 (CLNS). Podemos verificar que a única diferença significativa está no fato deste separar nos endereços o número da rede e do nó que são contidos em campos de comprimento variável. O “enchimento” serve para garantir que o tamanho do cabeçalho IP seja um múltiplo de 4 bytes.

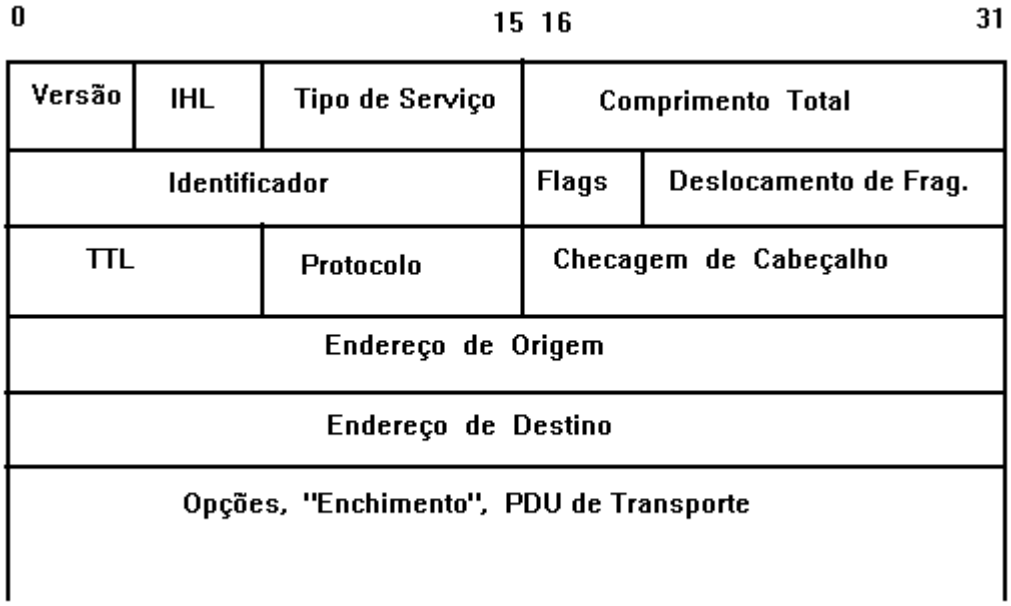


Figura 10.2 - Formato do PDU do IP

Os campos referentes à fragmentação de pacotes operam exatamente como no CLNS: o comprimento total, identificador e o deslocamento de fragmento. Já para o roteamento, o campo que interessa é o endereço de destino. A figura 10.3 mostra uma configuração típica de interligação de redes. Cada roteador

é ligado a duas, ou mais, redes diferentes e recebe um endereço IP próprio para cada uma das redes. Quando chega um pacote, a parte da rede de seu endereço de destino é comparada com o número de cada rede ao qual está ligado o roteador. Caso o destino seja um "vizinho" deste nó, a mensagem é enviada diretamente pela interface apropriada. Senão, é consultada uma tabela que lista para cada rede qual é o próximo roteador no caminho.

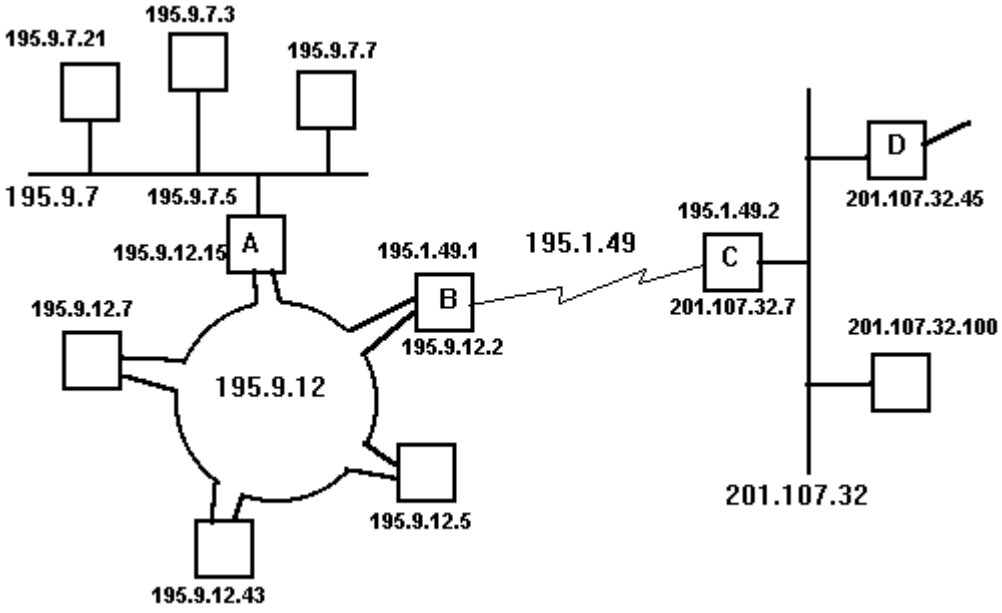


Figura 10.3 - Típica interligação de redes

Um problema com este esquema é que cada nó teria que ter uma tabela com uma entrada para cada rede existente no sistema (que pode ser o mundo inteiro, no caso da Internet). Observando a figura 10.3 vemos que todas as entradas da tabela no nó 195.9.7.21 indicam o roteador A (195.9.7.5 quando visto do primeiro nó) como o próximo passo da rota. Podemos compactar significativamente a tabela ao criar uma rota "default" que é usada sempre que a rede do destino não estiver explicitamente indicada na tabela. Neste caso, a tabela do nó 195.9.7.21 só precisa conter a rota default, já que só é consultada quando o nó destino não for da rede 195.9.7. A tabela do roteador B é, também, bastante compacta:

Rede Destino	**Interface**	**Próximo Nó**
195.9.7	Token Ring	195.9.12.15 (A)
0.0.0.0 (default)	Serial	195.1.49.2 (C)

Isto indica que o roteador B pode enviar pacotes diretamente para as redes 195.9.12 e 195.1.49 sem consultar a tabela, pode enviar para a rede 195.9.7 por intermédio de A e "para o resto do mundo" através de C. Uma tabela de apenas duas entradas é suficiente para o nó B rotear pacotes para todos os destinos possíveis. Isto é típico de roteadores na "periferia" do sistema, que é o caso mais comum.

10.3. Internet Control Message Protocol (ICMP)

O protocolo de mensagens de controle (ICMP) permite que os nós sejam avisados sobre problemas

na transmissão de pacotes. Note que a geração destes avisos é opcional - um pacote pode ser sumariamente descartado sem nenhuma explicação. Apesar de ser, na prática, uma extensão do IP, o ICMP é implementado como se fosse um protocolo de nível de transporte com seus PDUs encapsulados dentro dos PDUs do IP.

Dos 13 pacotes definidos neste protocolo, o número 3 (destino inatingível) é o mais comum. Ele indica detalhes do problema para que o usuário possa saber se a máquina está fora do ar, se há um problema na rede pelo caminho, se faltou permissão para fragmentar o pacote, ou se o processo especificado não está operando na máquina destino.

Os pacotes 8 (Pedido de Eco) e 0 (Resposta de Eco) são utilizados pelo programa "ping" para testar o funcionamento e desempenho da rede de comunicação entre dois nós.

Já o pacote 5 (Redirecionamento) é usado pelo roteador para indicar para o nó de origem a existência de um caminho mais curto. Suponha que o nó 195.9.12.43 tente enviar uma mensagem para o nó 195.9.7.21 da figura 10.3 através do roteador B. Consultando sua tabela, B decide enviar o pacote para o roteador A pela interface Token Ring. Reparando que está enviando o pacote exatamente pela mesma rede em que chegou, o ICMP de B envia também um pacote de redirecionamento para a origem para avisá-la para usar A da próxima vez. Uma estratégia bastante popular é iniciar a tabela dos nós não roteadores com apenas uma entrada default indicando algum roteador. Ao receber pacotes ICMP de redirecionamento, o nó vai atualizando sua tabela dinamicamente. Assim, apenas as tabelas dos roteadores precisam ser alteradas quando há uma mudança na topologia da rede. Mesmo esta administração pode ser automatizada pela utilização de protocolos como o RIP (descrito mais adiante).

10.4. User Datagram Protocol (UDP)

Os dois protocolos de nível de transporte trabalham com o conceito de "portas" na interface com o nível de sessão. Estas portas são indicadas por números de 16 bits, de modo que qualquer porta do mundo (no caso do Internet) pode ser localizada por um endereço de 48 bits: número da porta mais endereço IP.

O UDP não acrescenta nenhuma funcionalidade ao IP além desta multiplexação. É usado por aplicações que desejam trocar pacotes isolados e que possam tolerar a possibilidade de perdas. A grande vatágem do UDP é evitar a complexidade de se estabelecer uma conexão.

Enquanto é comum aplicações escolherem um número de porta aleatório que não esteja em uso para se comunicarem, certos servidores trabalham com números de portas pré-estabelecidos para que outras aplicações possam "encontrá-los". Assim, ao se enviar uma mensagem para a porta 13 de outra máquina, recebemos um pacote com a hora e data daquele nó como resposta. A porta 11 serve para listar os usuários de um nó, e a 7 é um "eco" que simplesmente devolve o que foi enviado (para testes). O servidor de nomes usa sempre a porta 53 e o RIP a 520.

10.5. Transmission Control Protocol (TCP)

O UDP trabalha com o conceito de datagramas enquanto que o TCP implementa circuitos virtuais. Desta forma, o TCP é equivalente ao TP4 da ISO ao adaptar uma rede de datagramas (IP) para aplicações que operam com conexões lógicas. A figura 10.4 mostra a PDU usada neste protocolo. As portas de origem e destino funcionam como no UDP (inclusive com serviços, como ECO, que usam exatamente os mesmo números). Os campos de número de seqüência e número de confirmação implementam o protocolo de janelas. A diferença em relação aos protocolos vistos até aqui é que estes números se referem a bytes enviados, e não a pacotes. Assim, depois do pacote 100 com 600 bytes deve vir o pacote 700. O

campo janela indica quantos bytes (após o indicado pelo número de confirmação) o receptor pode aceitar. Isto serve para impor um controle de fluxo.

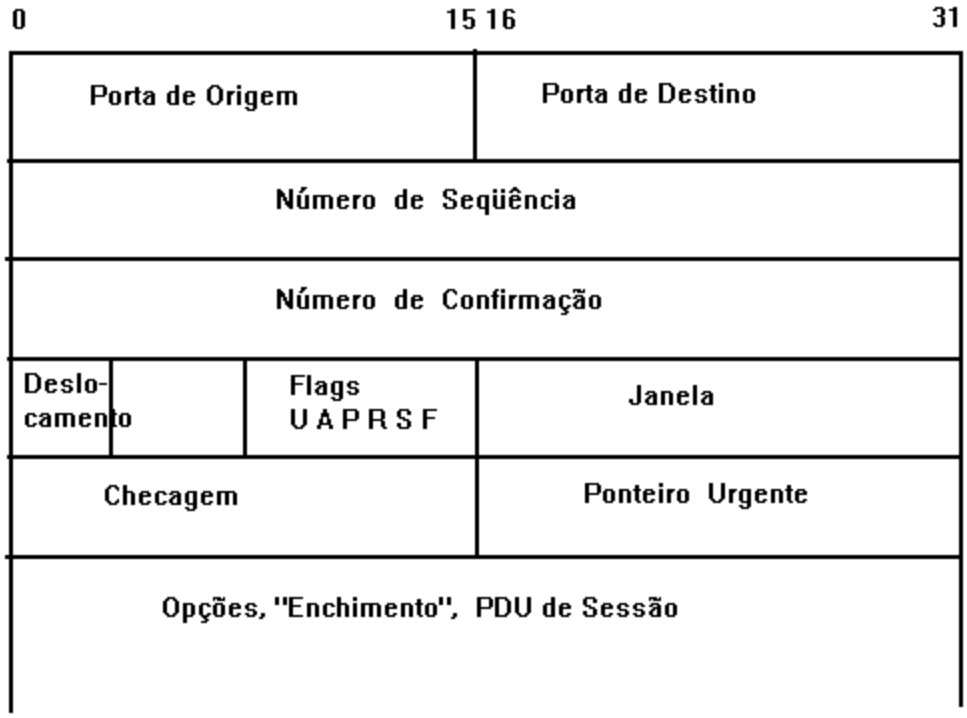


Figura 10.4 - Formato da PDU do TCP

O campo de 4 bits Deslocamento indica onde começa a PDU de Sessão no pacote (ou seja, qual é o tamanho do cabeçalho TCP incluindo as opções e "enchimento"). Os flags têm os seguinte significados: URG - indica que o campo ponteiro urgente tem um valor válido, ACK - o campo de confirmação tem um valor válido, PSH - exige a entrega imediata deste pacote, RST - reinicializa a conexão, SYN - o receptor deve sincronizar o número de seqüência, FIN - fim dos dados. O campo de checagem permite verificar se houve algum erro no processamento ou transmissão do pacote.

Normalmente os bytes surgem do outro lado exatamente na seqüência em que foram enviados. Em certos casos, porém, é necessário passar uma mensagem à frente das outras. Neste caso é indicado um byte pelo ponteiro urgente e todos os byte anteriores do pacote são imediatamente entregues ao destino. Isto equivale aproximadamente aos dados expeditos da ISO. Quando informações são enviadas pelo TCP, elas podem ficar armazenadas por um certo tempo em buffers na origem e no destino. O bit PUSH pode exigir que estes buffer sejam esvaziados para garantir a entrega imediata de todos os dados (como a função FLUSH dos sistemas de arquivos). Note que o PUSH não altera a ordem dos dados.

Quando é estabelecida uma nova conexão, o transmissor escolhe uma número aleatório de 32 bits como seu número de seqüência inicial e liga o bit SYN para que o receptor sincronize seu contador. A figura 10.5 mostra uma "conversa" típica do protocolo TCP. Normalmente a aplicação de um dos lados fica escutando em uma porta à espera que outra faça uma chamada. No caso do TCP existe apenas um único comando (ao contrário do NetBIOS que tem LISTEN e CALL) onde o nó remoto pode ser especificado ou não. Apesar da complexidade maior, o protocolo TCP tem como vantagem o fato de que consegue estabelecer uma conexão mesmo quando duas aplicações fazem chamadas simultâneas uma

para outra.

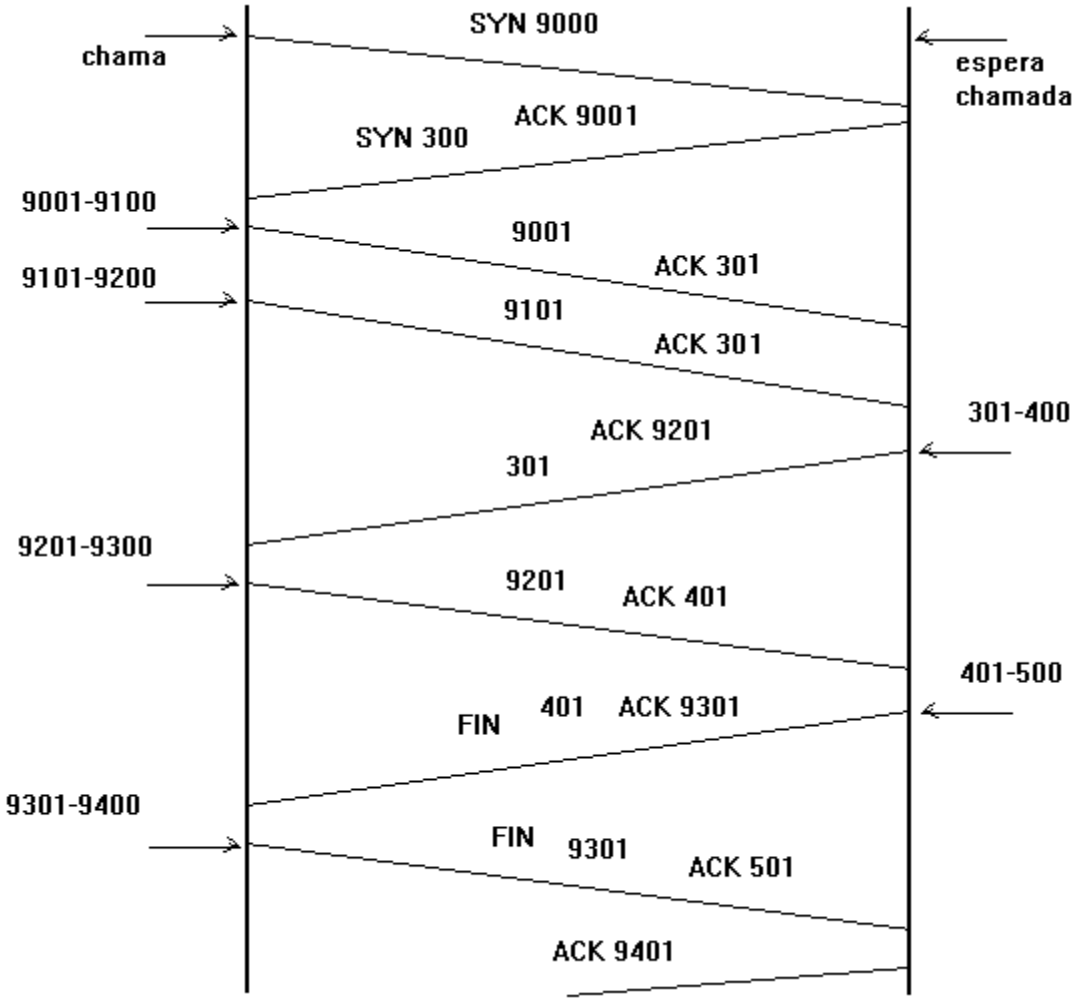


Figura 10.5 - Típica operação do protocolo TCP

Vemos uma máquina chamando outra que está a espera da conexão. Uma vez sincronizados os números de seqüência, o primeiro nó envia dois pacotes de 100 bytes cada (este não é um tamanho típico; foi usado no exemplo para simplificar os números). Não está sendo mostrado o valor do campo janela, mas podemos supor que é maior ou igual a 200 no exemplo. Após uma série de pacotes de dados o segundo nó envia um pacote com o bit FIN ligado. Isto significa que metade da conexão full-duplex está encerrada. O primeiro nó pode continuar enviado dados que serão confirmados. O seu último pacote terá o bit FIN ligado e, quando for confirmado, encerrará de vez a conexão.

Como o número de seqüência inicial é aleatório, pode ser escolhido um número que corresponda a um pacote de uma conexão anterior que esteja perdido pela rede (apesar disto ser de uma probabilidade ridiculamente pequena). Em função disto, após encerrada uma conexão só pode ser estabelecida outra com a mesma porta após um certo tempo que deve ser suficiente para quaisquer pacotes perdidos

morrerem de velhice (campo TTL - Time To Live). O flag RST (Reset) é usado em respostas a pacotes que chegam sem ser esperados (número de seqüência invalido, bit SYN ligado no meio da conversa, etc.). A reinicialização de uma conexão pode implicar na perda de dados já enviados.

Podemos dividir o "espaço" de bytes em três regiões: até o último ACK recebido estão aqueles que já chegaram do outro lado, depois destes até o meu número de seqüência estão os enviados que ainda não foram confirmados, e destes até o último ACK + a última janela são aqueles que posso enviar sem esperar que chegue algum pacote do outro lado. O byte indicado por ACK + janela só pode avançar, nunca voltar atrás. O valor de janela pode diminuir, é claro, a medida em que o outro lado vai recebendo dados. Se o valor deste compo chega a zero, dizemos que o outro lado "fechou a janela". Quando isto acontece devemos enviar periodicamente pacotes vazios para descobrir pela resposta se a janela continua fechada. Em implementações simplistas do TCP, ocorria o problema da "janela maluca" onde, após um lado fechar a janela, tão logo surgiam alguns bytes de buffer ele anunciava isto para o outro lado que enviava o número certo de bytes fechando a janela novamente. O desempenho cai dramaticamente pois a informação passa a ser enviada em pequenos pacotes menores que os cabeçalhos do TCP e IP. A solução para este problema é só avisar o outro lado que a janela abriu quando houver um buffer de tamanho razoável.

10.6. Aplicações

Os protocolos de mais alto nível não fazem distinção entre os níveis de sessão, apresentação e aplicação. Cada protocolo é obrigado a cuidar de todos estes aspectos. Os mais populares são: TELNET para terminais virtuais, o FTP para transferência de arquivos e o SMPT para correio eletrônico. Os protocolos relacionados com o SNMP estão ficando cada vez mais importantes e serão estudados na parte de gerenciamento de redes.

10.7. Routing Information Protocol (RIP)

Alguns dos protocolos que ainda não foram mencionados são o RARP, BOOTP e TFTP que viabilizam a implementações de estações sem disco. Os protocolos RIP e EGP cuidam da manutenção automática das tabelas de roteamento usadas pelo IP (inicialmente estas tabelas eram criadas e administradas manualmente).

No RIP as estações são divididas em passivas e ativas (roteadores). Todas autalizam sua tabela IP baseado nos pacotes RIP recebidos, mas apenas as estações ativas enviam estes pacotes. O pacote RIP contém, essencialmente, uma cópia da tabela IP do nó. Além das informações já mostradas, cada entrada tem um "custo" (que normalmente indica quantos nós intermediários o pacote deve percorrer até atingir o destino). O nó que recebe o pacote verifica se sua tabela já continha esta rota com um custo menor ou igual. Caso contrário, a informação é inserida na tabela com o custo incrementado de 1 e próximo nó igual à origem da mensagem RIP. Quando o custo é maior que o indicado na tabela, mas a origem da mensagem é igual ao próximo nó, o custo é atualizado pois indica uma alteração na topologia da rede. Normalmente cada roteador manda uma mensagem RIP para todas as redes ao qual está ligado a cada 30 segundos. É fácil ver que a informação de que o nó A pode atingir a rede 195.9.7 com um custo 0 se propaga para B, que eventualmente informa C de que pode atingir aquela rede com custo 1. Eventualmente o nó C dirá para D que pode atingir a rede 195.9.7 com um custo 4 (vamos atribuir um custo 3 à interface serial em função de seu baixo desempenho) e se este não tiver nenhuma alternativa melhor colocará a seguinte rota em sua tabela: rede = 195.9.7, custo = 5, interface = Ethernet, próximo nó 201.107.32.7 (C). Depois de um certo tempo, todos os roteadores ficam conhecendo a topologia completa da rede.

Um fenômeno indesejável ocorrerá se pifar a linha serial: o nó C verificará que não mais pode atingir

a rede 197.9.7, mas logo receberá um pacote RIP de D dizendo que este tem uma rota com custo 5. C instala, então, em sua tabela uma rota com custo 6 indicando D como próximo nó. A próxima vez que esta informação for enviada para D, este aumentará seu custo para 7, o que eventalmente fará C aumentar o seu custo para 8 e assim por diante. O RIP define 16 como sendo custo infinito para limitar o tempo que os nós levam para perceber que uma rede ficou desconectada (uns 6 minutos, neste caso). Isto limita o RIP a redes com extensão máxima de 14 roteadores - o EGP é usado para sistemas maiores. Este exemplo, a propósito, mostra como surgem "loops" de roteamento temporários que são contornados com o campo TTL (tempo de vida).

Uma solução para o problema acima é não transmitir para uma rede rotas que passam por aquela rede. Não tem sentido o nó D avisar para a rede 201.107.32 que tem uma rota para 197.9.7 já que esta passa por esta mesma rede (para o nó C). Assim, qualquer um da rede 201.107.32 que queira enviar para 197.9.7 vai sempre encontrar um caminho mais eficiente (através de C) do que tentar D. Com a complicação de ter que gerar um pacote RIP diferente para cada rede, evitamos o "loop" descrito acima. Esta técnica se chama "split horizon". Para apressar ainda mais a decoberta de falhas de comunicação podemos usar o "poison reverse" em que a rota é transmitida para a rede de onde veio, mas sempre com um custo infinito. Existem outras sofisticações que melhoram o desempenho. É interessante notar que o RIP é típico dos algorítmos distribuídos: fazer funcionar é uma coisa, fazer funcionar razoavelmente bem é outra bem mais complexa. A dificuldade é que cada nó tem uma visão parcial do estado do sistema e precisa decidir com base nesta informação incompleta.

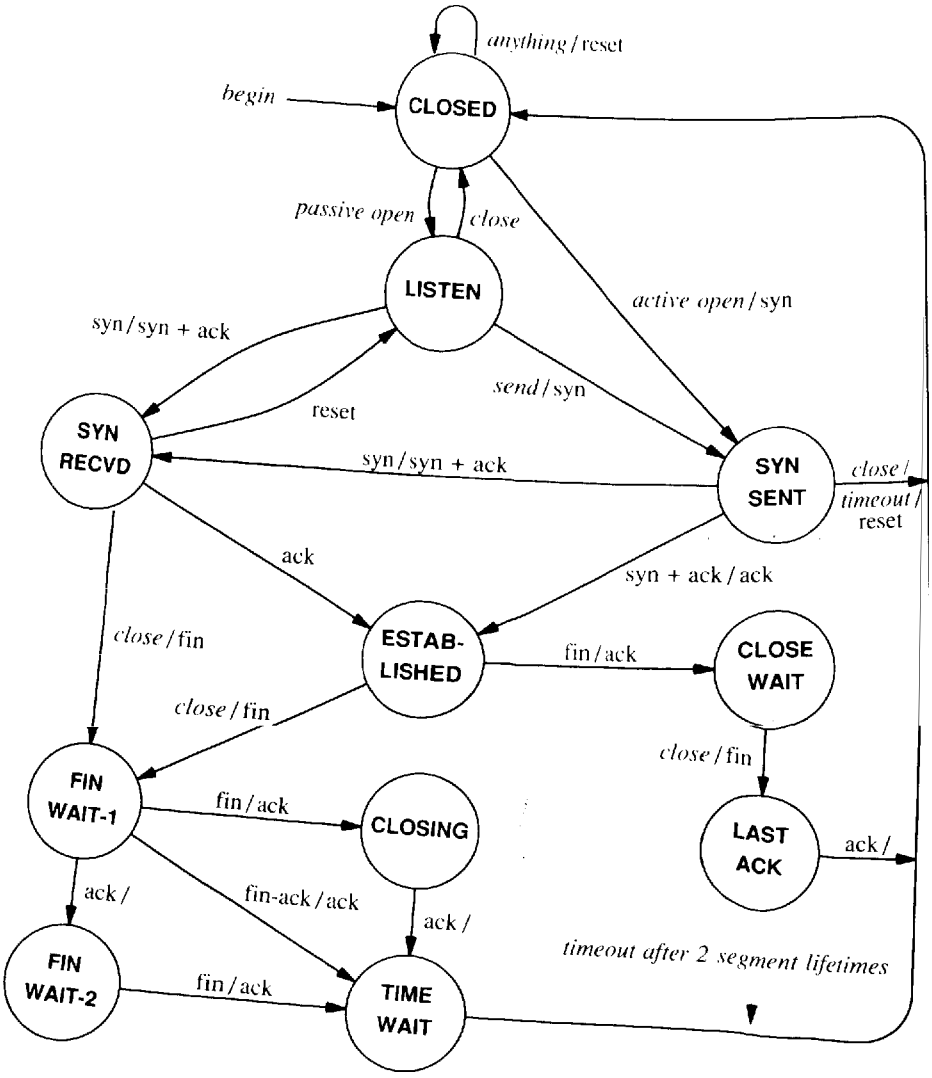
10.8. Sistema de Nomes por Domínios

O serviço de nomes por domínios permite o uso de nomes simbólicos para se referir a nós, no lugar de endereços IP. Isto dá uma flexibilidade maior ao permitir que o endereço mude e programas e pessoas continuem a usar um serviço em sua nova localização. O nome de um subdomínio é o nome local seguido de um ponto e do nome do domínio onde ele se encontra. Assim, fpsp.fapesp.br é o nó (ou subdomínio) fpsp dentro do domínio fapesp.br. Alguns domínios de mais alto nível são: com (comercial), edu (educacional), mil (militar), br (Brasil), de (Alemanha) e assim por diante.

Exercícios

1. Os padrões relacionados com o TCP/IP definem o ICMP (RFC 792 - protocolo de mensagens de controle interrede) que avisa quando pacotes não podem ser entregues explicando a origem do problema. Um dos usos mais populares deste protocolo é "ecoar" mensagens do utilitário PING para verificar se uma máquina "está no ar" a ligação está em ordem. Por que, então, foi definido o protocolo ECHO (RFC 862 - protocolo de eco de mensagens) que usa a porta 7 do TCP ou do UDP?

2. Por que os roteadores possuem um endereço IP diferente para cada rede a qual estão ligados?



3. A figura ao lado é a máquina de estados finitos de um dos protocolos estudados no curso. Qual protocolo é este?

4. Como podemos ver **pela figura** se o protocolo implementa circuitos virtuais ou datagramas?

5. Suponha que uma máquina está no estado "established" e receba um comando "close" da camada superior. Quantos pacotes irá gerar até atingir o estado "closed"?

6. Faça um esboço do conjunto de redes que contém um roteador com a seguinte tabela:

Rede Destino	**Interface**	**Próximo Nó**	**Custo**
195.12.3	Ethernet	195.7.1.1	1
195.25.48	Token Ring	195.3.50.23	1
0.0.0.0 (default)	Token Ring	195.3.50.9	

7. O bit URG do protocolo TCP permite que dados importantes passem à frente do fluxo normal de informações. Como isto é feito no TP4 da ISO?

8. A classe I do IEEE 802.2 (LLC) implementa um serviço de datagramas, enquanto que a classe II trabalha com circuitos virtuais. Qual é a mais adequada para uso com o TCP/IP? Por que?

9. O protocolo de informações de roteamento (RIP) usa o UDP como seu nível de transporte. O UDP **não** garante a entrega dos pacotes. Por que isto não é um problema para o RIP?

10. Suponha que você irá implementar o protocolo ARP no sistema operacional de uma nova máquina que está sendo desenvolvida. O que deve determinar o tamanho máximo da tabela de endereços usada neste protocolo?

11. O programa PING envia o pacote ICMP 8 (pedido de eco) e fica aguardando um pacote 0 (resposta de eco) desta. Como isto pode ser usado para testar o desempenho da rede?

12. Mostre (com um diagrama no estilo da figura 10.5 da apostila) o que acontece se dois processos tentarem chamar um ao outro simultaneamente. Indique apenas o estabelecimento da conexão - não é necessário mostrar a “conversa” completa.