Unidade II

3 AS CAMADAS DE APRESENTAÇÃO, SESSÃO E TRANSPORTE

3.1 A camada 6: apresentação

A camada de apresentação não tem uma preocupação declarada com os princípios dos níveis de dados em bits, mas sim com sua sintaxe, ou seja, sua representação. Nela são definidas a sintaxe abstrata, a forma como os tipos e os valores dos dados serão definidos, independentemente do sistema computacional usado em sua sintaxe de transferência, ou seja, a maneira como se realiza essa qualificação. Um bom exemplo através da sintaxe de abstração é definir a forma como um caractere deve ser transmitido, aceitar o protocolo de transferência específico e então negociar o formato de codificação do dado, que poderá ser ASCII ou EBCDIC, o resultado do dado então será entregue à camada sessão.

A principal função da camada de apresentação é representar os dados para que sejam legíveis para a camada de apresentação do dispositivo de destino. Nesse nível, a camada de apresentação precisa conhecer a sintaxe de seu sistema local e também a do seu sistema de transferência.

Os serviços oferecidos nesse nível são a representação dos dados, a formatação dos dados, a seleção das sintaxes e o estabelecimento e manutenção das conexões da apresentação.

Existe uma correspondência atuante entre os endereços da apresentação e da sessão, e nesse caso não há existência da multiplexação no nível do protocolo.

Aliada às funções de representação de dados, a camada de apresentação também é responsável pela realização da compactação e da criptografia.

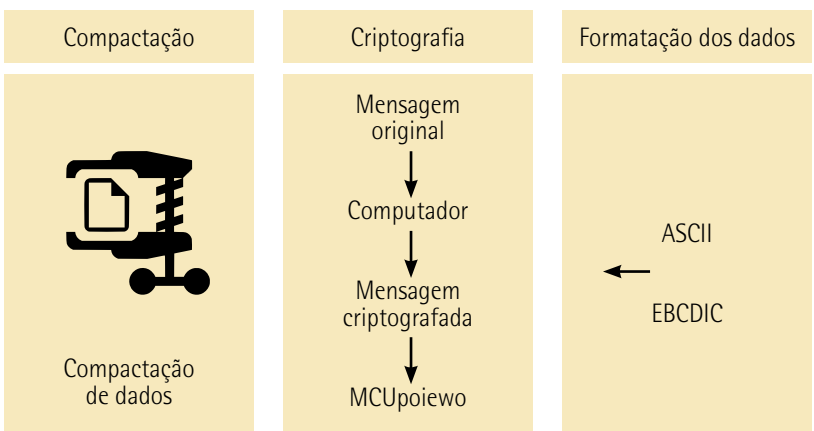


Figura 4 – Funções da camada de apresentação

JPEG e GIF são bons exemplos de padrões de formatação que são definidos na camada de apresentação. O padrão MPEG compõe o grupo definido pela ISO para padronização e compressão de transmissão de áudio e de vídeo. Já o padrão JPEG é usado para compressão de dados geralmente utilizados na composição de imagens fotográficas. O padrão GIF é utilizado na formatação de imagens de baixa resolução, como o uso dos ícones.

3.2 A camada 5: sessão

A camada de sessão oferece mecanismos que permitem a estruturação dos circuitos que são oferecidos pelo nível de transporte. Os principais serviços fornecidos nesse nível são o gerenciamento do token, o controle do diálogo e o gerenciamento das atividades.

Embora um circuito que permita transmissões nos dois sentidos seja necessário para o intercâmbio das informações, em algumas aplicações essa troca de informações é do tipo half-duplex em vez de ser full-duplex. Com a intenção de fornecer o serviço de intercâmbio de informações half-duplex em um circuito full-duplex, o serviço da sessão usa conceitos de token em uma comunicação half-duplex.

O proprietário do token dos dados pode transmitir os seus dados. O serviço da sessão, então, fornece os mecanismos de gerenciamento, a posse e a passagem deste token entre as entidades da aplicação que estão utilizando este serviço.

No momento que ocorre um volume muito grande dos dados, por exemplo, um arquivo muito extenso é transmitido em redes não muito confiáveis. Essa rede pode basicamente deixar de funcionar, então, resta ao nível de transporte indicar qual tipo de falha e deixar a aplicação decidir o que pode ser feito. Eventualmente, a rede pode voltar a funcionar, podendo a conexão ser restabelecida. No caso desse restabelecimento, o ideal seria que a transferência dos dados pudesse ser retomada do ponto exatamente ou imediatamente anterior ao da interrupção. Com o objetivo de fornecer esse tipo de serviço, o nível da sessão usa o conceito de ponto de sincronização.

O ponto de sincronização é uma marca lógica que é posicionada na extensão do diálogo entre os dois usuários do serviço dessa sessão. A qualquer tempo, toda vez que se recebe um ponto de sincronização, o usuário do serviço da sessão deve então responder ao aviso do recebimento ao usuário com quem está se dialogando. Se, por qualquer motivo, uma conexão foi interrompida e depois restabelecida, os usuários podem retomar o diálogo a partir do último ponto de sincronização confirmado.

O conceito da atividade torna possível aos usuários dos serviços da sessão a distinção das partes do intercâmbio nos dados, normalmente denominada atividade. Cada atividade pode então consistir em uma ou mais unidades/partes desse diálogo. Em uma conexão da sessão só é permitida a execução de uma atividade por vez, porém, em algumas circunstâncias, podem existir várias atividades consecutivas durante a concepção da conexão.

Uma atividade pode ser interrompida e depois recomeçada nessa mesma sessão ou em conexões de sessão subsequentes. Para um bom exemplo do uso do conceito de uma atividade, vamos considerar o envio de uma mensagem através de um sistema de correio eletrônico como uma atividade específica, vamos supor que esta mensagem é grande e de baixa prioridade. Durante o método de transmissão, a entidade do nível da sessão que está enviando essa mensagem recebe uma solicitação para enviar uma outra mensagem de prioridade maior, essa entidade então pode suspender a atividade corrente e transferir a mensagem com a prioridade alta e começar, nesse caso, uma nova ou uma outra atividade que, posteriormente, poderá retomar a atividade inicial, que é a transmissão da mensagem com prioridade baixa, sempre usando o conceito de atividade.

O nível da sessão sempre permite que os dois usuários suspendam um diálogo, por exemplo, o fim do expediente, naturalmente desfazendo a conexão da sessão e retomando a posterior, no início do próximo expediente, usando uma nova conexão da sessão.

3.3 A camada 4: transporte

Para conhecer os detalhes da camada de transporte, é preciso entender que na camada de rede, ou seja, na camada antecessora, não há garantia de que os dados e pacotes cheguem ao seu destino. Estes podem ser perdidos ou, ainda, chegar fora da sequência original da transmissão.

Para fornecer uma estratégia de comunicação fim a fim que seja confiável de verdade, é necessário um nível de protocolo, que é este oferecido pela camada transporte. Esse nível tem a intenção de isolar os níveis superiores da transmissão da rede.

No nível da camada de transporte, a comunicação é do tipo fim a fim. A entidade do nível de transporte da máquina que origina a comunicação se comunica com a entidade do nível de transporte da máquina a que se destina a comunicação. Isso não necessariamente acontece em níveis físicos do enlace ou da rede onde esta comunicação se dá entre máquinas adjacentes ou máquinas vizinhas na sua rede.

As funções mais importantes nesse nível do modelo OSI são a multiplexação, em que várias conexões de transporte compartilham a mesma conexão de rede, e o spliting, que são as conexões de transporte ligadas a várias conexões de rede. O spliting é usado para superdimensionar a vazão de uma conexão do transporte usando várias conexões de rede simultaneamente. Já a multiplexação é usada quando uma conexão de transporte não tem geração de tráfico suficiente para ocupar toda a capacidade da conexão da rede por ela usada.

Outra função não menos importante no nível de transporte é o controle de fluxo. Assim, nenhuma implementação de um espaço de armazenamento, seja infinito ou mesmo algum mecanismo que deva ser utilizado no módulo, faz evitar que o transporte envie mensagens a uma taxa muito maior do que a capacidade de receber.

Além de todas as funções mencionadas, ainda podemos lembrar de funções nesse nível de controle, como a sequência de informe de dados fim-a-fim, a detecção e a recuperação de erros do tipo fim-a-fim e também a segmentação e blocagem das mensagens, entre outras.

Recapitulando as atribuições da camada transporte, temos as funcionalidades:

- Serviço orientado à conexão.
- Entrega ordenada.
- Entrega confiável.
- Controle de fluxo.
- Identificação das diferentes aplicações.

3.3.1 Serviço orientado à conexão

A camada transporte faz uso do serviço orientado à conexão para garantir confiabilidade.

O fato de ser um protocolo orientado à conexão indica que uma sessão precisa ser estabelecida entre destino e origem antes de transmitir dados. Após essa sessão ser estabelecida, os dados poderão ser transmitidos, e após o término da transmissão dos dados, a sessão será encerrada na camada de transporte da comunicação por meio de um handshake triplo. O handshake triplo é, na verdade, a sincronização iniciada pelo cliente ao servidor.

Fase 1

A entidade que está iniciando a comunicação transmite o segmento contendo o número de sequência para inicialização, indicando o início da comunicação ==> SYN inicial.

Fase 2

A entidade receptora responde com um ==> SYN/ACK, confirmando o estabelecimento da comunicação.

Fase 3

A entidade que iniciou a comunicação responde esta confirmação completando a fase de estabelecimento e a sincronização da comunicação.

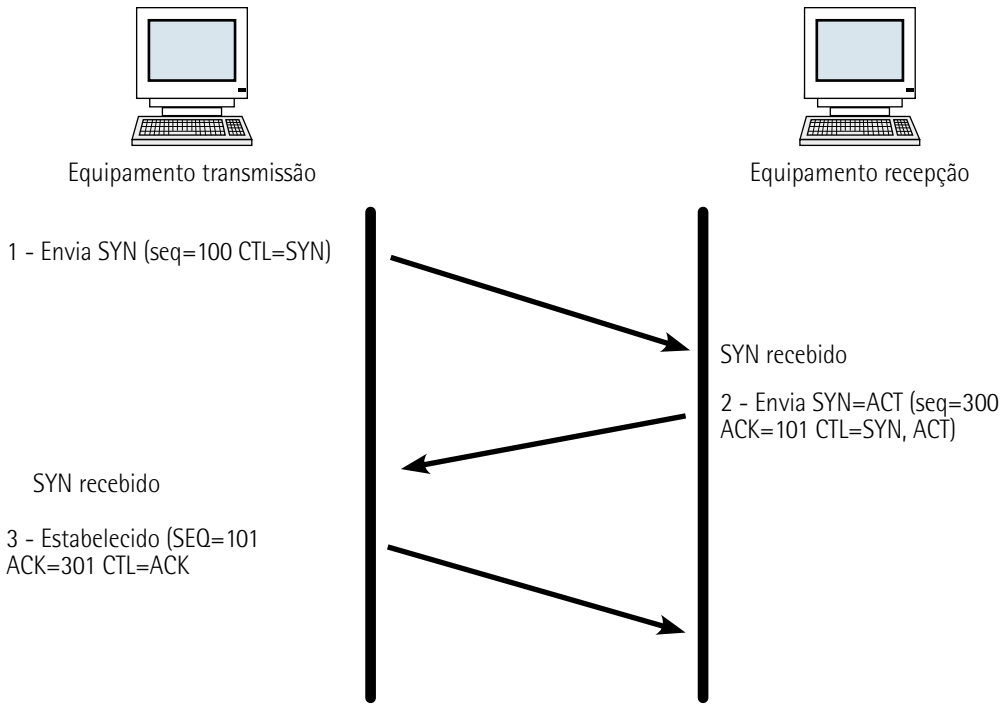


Figura 5 – Sincronização do handshake triplo

No momento em que a comunicação é estabelecida, esta fase já se encontra concluída e os dados podem ser transmitidos. Somente depois do handshake triplo os dados serão enviados pela entidade de origem e, após eles serem transmitidos, a sessão precisa ser encerrada.

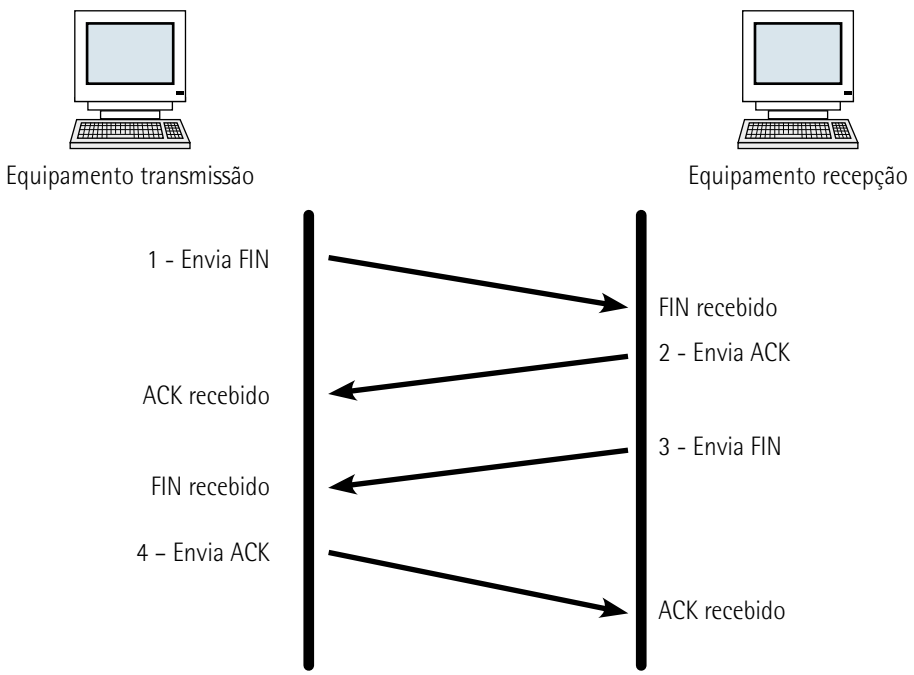


Figura 6 – Finalização de uma conexão

Saiba mais

Para saber mais sobre a camada transporte, leia os capítulos 12 e 13 do livro:

COMER, D. E. *Internetworking with TCP/IP*. 4. ed. New Jersey: Prentice Hall, 2000. v. 1.

3.3.2 Entrega ordenada

Em uma comunicação, quando diversos datagramas são enviados entre a entidade de origem e a entidade de destino, a chegada dos datagramas ao seu destino pode ser encarada de forma desordenada, justamente pelas diversas possibilidades de rota que estão disponíveis em uma comunicação em rede. Para que eles possam ser organizados e ordenados ao seu destino, cada datagrama recebe um número de sequência. Quando esses datagramas chegam fora da sua ordem original, eles são colocados em um buffer para que depois de organizados e ordenados possam ser entregue às camadas superiores.

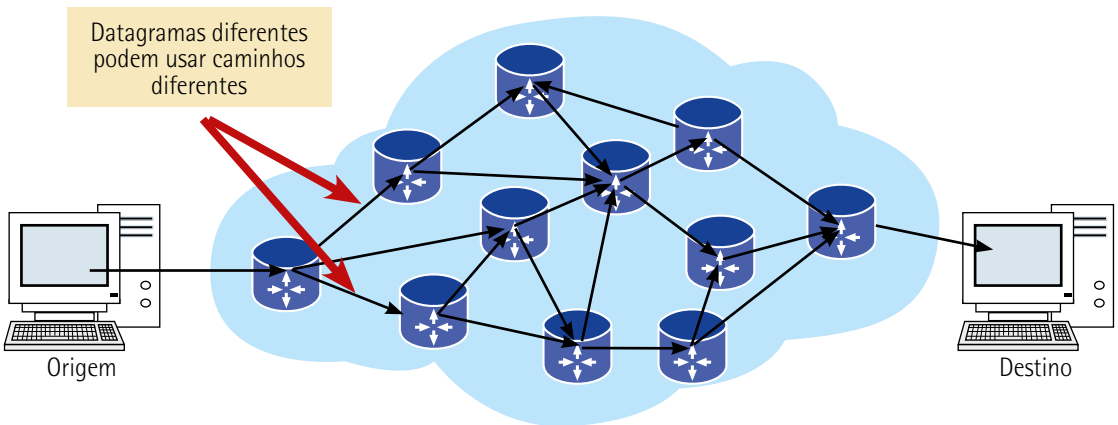


Figura 7 – Possibilidades de caminho em uma rede de pacotes massiva

Nesta maneira de organizar os datagramas na camada de transporte, existe a possibilidade da retransmissão dos datagramas faltantes.

3.3.3 Entrega confiável

Para garantir confiabilidade em uma comunicação, a camada de transporte utiliza o conceito de confirmação positiva ou confirmação esperada. Nesse caso, são usados números sequenciais juntamente com os números de confirmações (ACK). Ao receber esses datagramas que foram enviados pela entidade de origem, a entidade de destino confirma o recebimento desses datagramas, pedindo o próximo na fila, ou seja, o próximo datagrama é solicitado e, desta forma, a entidade de origem entende que a entidade de destino recebeu todos os datagramas anteriores.

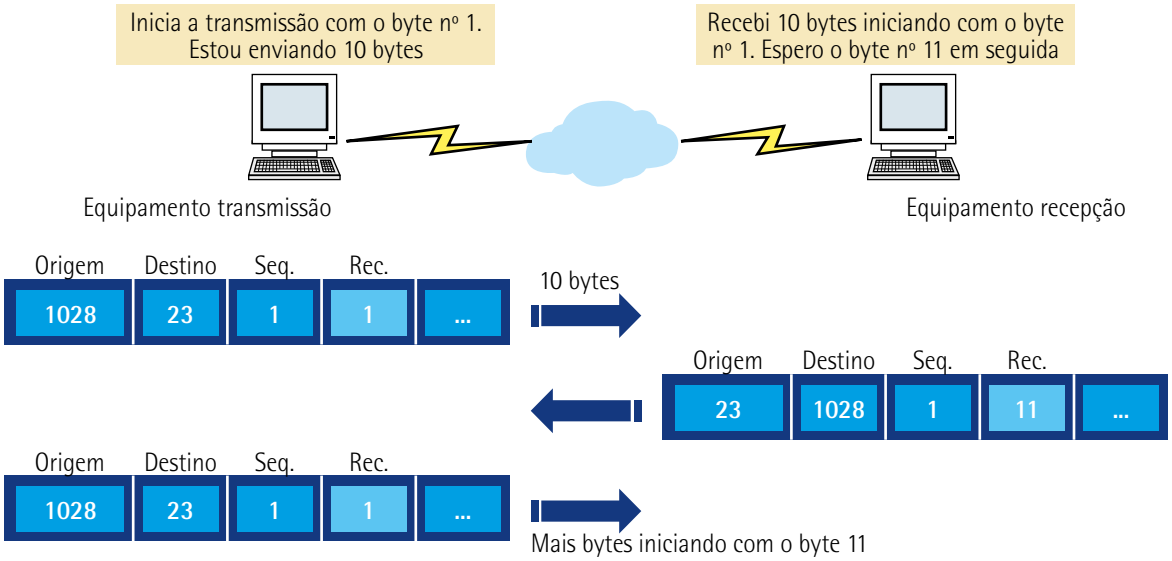


Figura 8 – Confirmação positiva

3.3.4 Controle de fluxo

Gerenciamento e controle de fluxo das informações é uma atribuição da camada de transporte e indica a quantidade de informação que poderá ser transferida antes de aguardar uma confirmação do recebimento ao seu destino. A camada de transporte então faz uso do janelamento para essa função.

O janelamento é considerado uma janela móvel, também é conhecida como janela deslizante, ou seja, o valor do tamanho da janela não é fixo, os valores vão sendo alterados durante a transmissão. Assim, o fluxo das informações vai sendo gerenciado quando ocorre então o controle de fluxo.

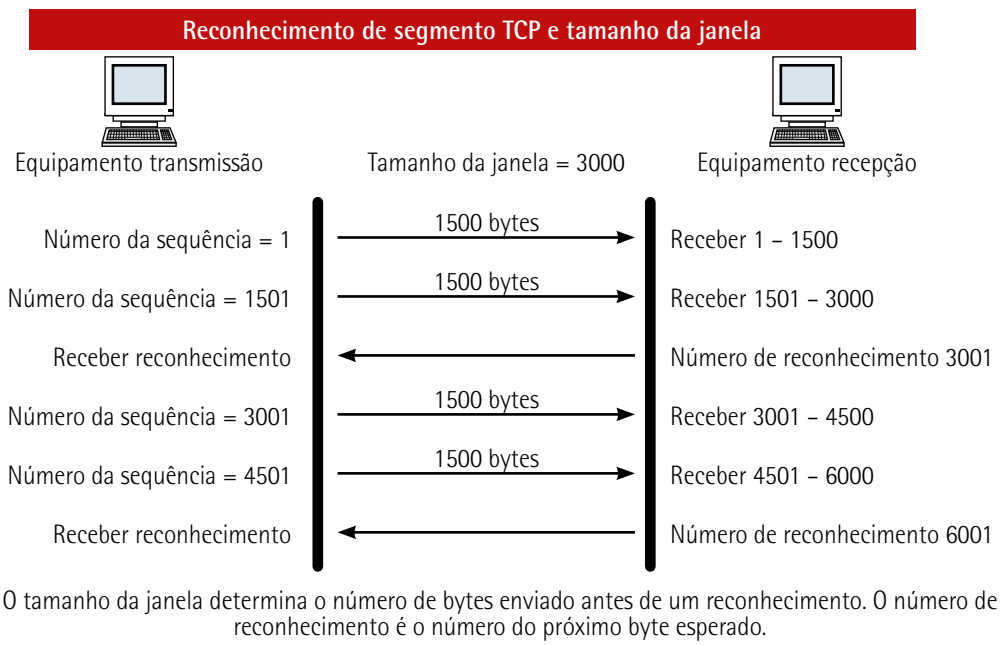


Figura 9 – Controle de fluxo

3.3.5 Como a camada transporte identifica as diferentes aplicações

A forma como a camada transporte identifica diversas comunicações simultâneas, quando essas ocorrem entre as entidades de origem e de destino, parece complexa, mas na verdade é muito simples, como seria um dispositivo que opera diferentes aplicações de rede simultaneamente, por exemplo, a navegação na internet e o envio de um e-mail. Como esse dispositivo vai identificar qual aplicação precisa receber o dado que chegou através da rede? A resposta está na atribuição de portas.

Números de portas são usados para identificação dessas comunicações pelas diversas aplicações do usuário. Quando o dispositivo inicia uma comunicação, ele atribui um número de porta de origem e outro número de porta para o destino, essa porta de origem identifica a comunicação na sua origem enquanto a porta do destino vai identificar a aplicação que vai receber a informação ao seu destino. No retorno da sua comunicação, esses números são trocados sistematicamente.

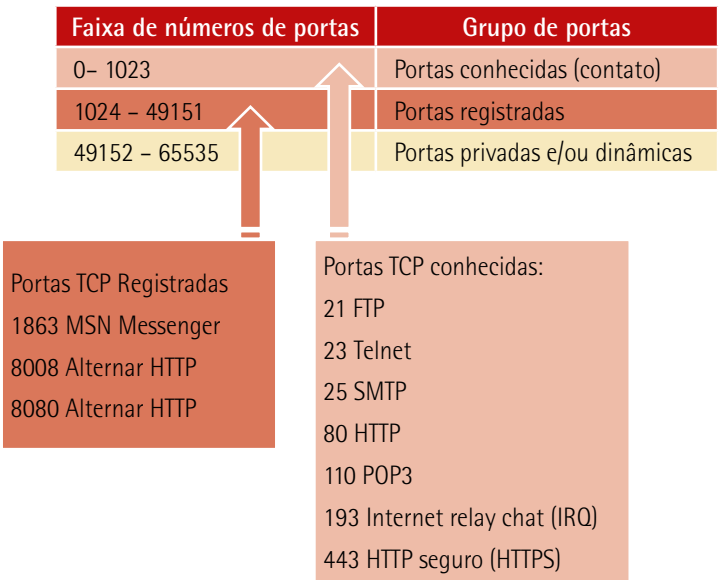


Figura 10 – Número de portas

A primeira faixa, de 0 a 1023, identifica as portas conhecidas, ou seja, números de portas para aplicações previamente estabelecidas. Veja as principais aplicações e seus números de portas:

Tabela 1 – Endereços de porta TCP das principais aplicações

Número da porta	Protocolo
20 e 21	FTP
22	SSH
23	Telnet
25	SMTP
53	DNS
69	TFTP

80	HTTP
110	POP3
143	IMAP
443	HTTPS

Adaptado de: Red Hat Enterprise (2005).

A segunda faixa de números de portas, de 1024 a 49151, identifica as portas registradas. Estas identificam processos ou aplicações do usuário, ou seja, aplicações individuais do usuário final. As portas registradas também podem ser usadas dinamicamente, como uma porta de origem do dispositivo que inicia a comunicação. Um exemplo comum de uma porta registrada é a porta 1863, do MSN.

A terceira faixa de números de portas, de 49152 até 65535, identifica as portas privadas ou dinâmicas. Esses números de portas são geralmente usados dinamicamente por aplicações do dispositivo que inicia a transmissão, apesar de que geralmente esses dispositivos podem usar portas registradas.

3.3.6 Protocolo orientado à conexão

Como sabemos, a camada transporte fornece um serviço orientado à conexão, e o protocolo da camada de transporte que fornece esse serviço é o TCP (Transmission Control Protocol). Assim, o TCP aplica todas as funções de entrega de forma ordenada, confiável e com controle de fluxo.

Para usar os recursos de entrega ordenada, confiável e com controle de fluxo, o TCP precisa usar uma estrutura de datagrama que comporta todas essas funções.

Bit 0 — Bit 15	Bit 16 — Bit 31
Porta de origem (16)	Portal de destino (16)
Número de sequência (32)	
Número de reconhecimento (32)	
Comprimento do cabeçalho (4) Reservado (6) bits de código (6)	Janela (16)
Checksum (16)	Urgente (16)
Opções (0 ou 32, se houver)	
Dados da camada de aplicativos (tamanho varia)	

Figura 11 – Estrutura do datagrama TCP

Os campos que constam na figura são:

- Porta de origem: campo de 16 bits que contém o número da porta origem.
- Porta de destino: campo de 16 bits que contém o número da porta de destino.

- Número de sequência: campo de 32 bits utilizado para ordenar os datagramas.
- Número de reconhecimento: campo de 32 bits com o número de confirmação que indica o próximo segmento TCP esperado.
- Comprimento do cabeçalho: campo de 4 bits que indica o tamanho do cabeçalho do datagrama.
- Janela: campo de 16 bits com o número de segmentos que poderão ser transmitidos antes de aguardar uma confirmação.
- Checksum1: campo de 16 bits para o cálculo de verificação de erros.
- Dados: campo com os dados das camadas superiores.

3.3.7 Protocolo não orientado à conexão

A camada transporte nem sempre precisa oferecer um serviço confiável, no qual é preciso estabelecer uma comunicação entre origem e destino antes de enviar os dados, mas sim oferecer uma entrega ordenada com controle de fluxo. Em alguns casos, em que a confiabilidade da comunicação não é necessária, um protocolo não orientado à conexão pode ser usado.

O protocolo de camada de transporte que pode fornecer o serviço não orientado à conexão é o UDP (User Datagram Protocol). A seguir, temos detalhes desse datagrama.

Bit 0 — Bit 15	Bit 16 — Bit 31
Porta de origem (16)	Portal de destino (16)
Comprimento (16)	Checksum (16)
Dados da camada de aplicativos (tamanho varia)	

Figura 12 – Estrutura do datagrama UDP

Os campos que constam na figura são:

- Porta de origem: campo de 16 bits que contém o número da porta origem.
- Porta de destino: campo de 16 bits que contém o número da porta de destino.
- Comprimento: campo de 16 bits que indica o tamanho do datagrama, incluindo os dados.
- Checksum: campo de 16 bits para o cálculo de verificação de erros.
- Dados: campo com os dados das camadas superiores.

As diferenças entre o protocolo TCP e UDP

Podemos observar que os protocolos TCP e UDP possuem semelhanças e diferenças. Em primeiro lugar, vale lembrar que a função deles é basicamente a mesma, ou seja, o transporte de dados das camadas superiores entre os dispositivos finais e a diferenciação das diversas conversações em formato simultâneo por meio de números de portas. Os dois protocolos possuem campos de números de portas e de checksum, e também campos de dados com funções equivalentes.

Porém as semelhanças param por aí. Podemos observar que o protocolo TCP possui mais campos do que o UDP, exatamente pelo fato do TCP oferecer serviços orientados à conexão com confiabilidade.

Além do TCP possuir um cabeçalho muito maior que o UDP, são 20 bytes para o TCP e 8 bytes para o UDP, o overhead que o protocolo TCP impõe é bem maior, ou seja, o protocolo UDP é bem mais leve. Assim sendo, o protocolo UDP poderá ser usado em princípios de comunicação nos quais não seja necessário existir a confiabilidade, embora isso não seja recomendado.

4 A CAMADA DE REDE, OS PROTOCOLOS IPV4 E IPV6

4.1 A Camada 3: rede

A camada de rede é responsável pela atribuição de endereçamento lógico e também permite a transferência de dados da origem até o destino em uma rede de comunicação. Outro atributo dessa camada é permitir que dispositivos possam se comunicar através de diversas redes interconectadas.

Sempre que precisamos de uma aplicação que depende de comunicação remota a um determinado equipamento ou que precisamos interligar outros equipamentos formamos uma rede de comunicação.

A camada de rede fornece serviços que permitem a transferência dos dados da origem até o destino em uma comunicação de dados. Essa camada possui quatro atributos básicos:

- Endereçamento: é o processo que define os endereços para os dispositivos existentes em uma rede e que permite a comunicação dos dados. Existem padrões de endereçamento de acordo com o protocolo de rede escolhido.

- Encapsulamento: é o processo de empacotar, segmentar, mudar o fluxo de dados que deve ser transmitido pela rede dentro do protocolo da camada usado por esse processo. São criados pacotes com as informações que serão entregues ao destino pela rede de comunicação.

- Roteamento: é o processo que tem a tarefa de direcionar os pacotes que serão montados no processo de encapsulamento através da rede de dados. O roteamento é usualmente realizado pelos dispositivos que trabalham na camada 3. É a intenção de escolher o melhor caminho para entrega ou a entrega mais eficiente de cada pacote ao seu destino. Usualmente, essa função é realizada por equipamentos chamados de roteadores.

- Desencapsulamento: é o processo de desempacotar e retirar o conteúdo de dados de cada pacote recebido e entregá-los à camada superior do modelo de referência OSI, que nesse caso é a camada de transporte.

Vários outros protocolos foram desenvolvidos para atender às funcionalidades básicas da camada de rede. Esses protocolos foram criados para atender funcionalidades específicas de cada fabricante, como:

- O IPv4, internet protocol versão 4.
- O IPv6, internet protocol versão 6.
- O IPX, Novell Network Packet Exchange.
- O AppleTalk.

4.2 O protocolo IPv4

O protocolo IPv4 ainda é um protocolo bastante difundido pelo mundo. Um bom exemplo de sua aplicação é a rede de comunicação internet, que permite todas as facilidades de roteamento e endereçamento necessários. Em breve esse protocolo estará sendo substituído pelo IPv6, que será a versão dominante da internet.

Esse protocolo foi especificado e alterado nas RFCs 791, 950, 919, 922, 1349 e 2474. O grande mérito desse protocolo é sua utilização ser permitida em qualquer tipo de rede física com interoperabilidade no nível da perfeição, entre as diversas tecnologias de rede existentes.

Cada pacote criado pelo protocolo IPv4 em uma comunicação tem tratamento isolado durante toda a sua vida ao longo do percurso na rede. Esse é o motivo pelo qual o IPv4 é um protocolo em que inexiste conexão, em que os pacotes são tratados e avaliados a cada nó, ou seja, a cada equipamento por onde eles trafegam.

Uma das características do tratamento desses datagramas na rede de comunicação é que os datagramas IPv4 podem ser entregues a seu destino não obedecendo a ordem de saída. O grande mérito dessa característica é graças às camadas superiores, como a de transporte, que podem reagrupar esses datagramas em sua ordem original e entregá-las às camadas superiores em ordem cronológica adequada.

Lembrete

Para que eles possam ser organizados e ordenados ao seu destino, cada datagrama recebe um número de sequência. Quando esses datagramas chegam fora da sua ordem original, eles são colocados em um buffer para que depois de organizados e ordenados possam ser entregue às camadas superiores.

O datagrama ou pacote do IPv4 é muito simples de ser compreendido. Basicamente, temos um campo cabeçalho e um campo de dados. O campo cabeçalho do IPv4 é definido por diversos campos que são utilizados para permitir o endereçamento e o roteamento correto dos pacotes pela rede.

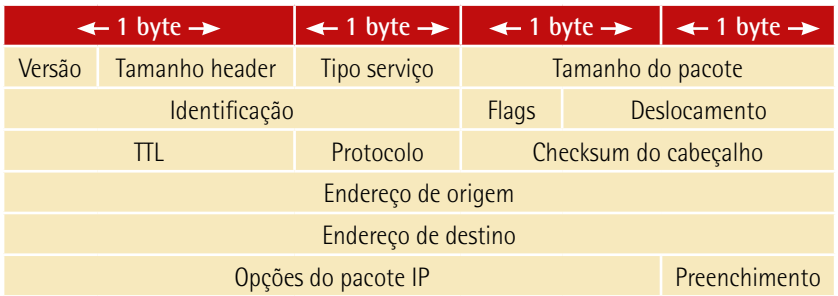


Figura 13 – O cabeçalho do datagrama IPv4

A definição dos campos do cabeçalho IPv4 é a seguinte:

- Versão: versão do protocolo, no caso 4.
- Tamheader: corresponde ao tamanho do cabeçalho contado em números de palavras de 32 bits (4 bytes).
- Tipo serviço: é o campo que contém a indicação de qualidade do serviço desejado para o encaminhamento do pacote. Esse campo possui 8 bits.
- Tampacote: campo que contém o tamanho do pacote em quantidade de octetos (bytes). O valor máximo é 65.535 bits.
- Identificação: é o campo preenchido pela origem do pacote que o identifica. É usado na montagem da sequência dos pacotes no destino. Um pacote que precisa ser fragmentado por outro equipamento no caminho até o seu destino utiliza, neste campo, o mesmo valor para todos os fragmentos resultantes.
- Flags: campo de 3 bits que identifica se o pacote pode ser fragmentado no caminho até o destino e também se já ocorreu fragmentação. O primeiro bit é sempre 0, o segundo bit indica se pode ou não fragmentar (0 = pode fragmentar, 1 = não pode fragmentar), e o terceiro bit indica se este pacote é (1) ou não é (0) o último fragmento.
- Deslocamento: caso tenha ocorrido fragmentação, este campo indica o deslocamento dos dados do pacote em relação ao campo de dados do pacote original (antes da fragmentação). Este campo é primordial para a remontagem do pacote e considera como unidade um octeto (1 byte).
- TTL (Tempo de Vida): representa a quantidade de saltos por onde um pacote pode trafegar. Cada ativo de rede que roteia este pacote diminui o TTL de 1, sendo descartado quando este valor chega a zero.

- Protocolo: campo preenchido com um valor numérico que identifica para qual protocolo da camada superior a camada de rede deve entregar o conteúdo deste pacote no momento em que ele chegar ao destino. Exemplo: 6 – TCP, 17 – UDP, 1 – ICMP, 89 – OSPF etc.

- Checksum do cabeçalho: é o campo calculado e checado para cada salto que o pacote passa na rede, a fim de verificar a integridade do cabeçalho.

- Endereço de origem: é o endereço de origem do pacote, composto por 32 bits.

- Endereço de destino: é o endereço de destino do pacote, composto por 32 bits.

- Opções do pacote IP: este campo é opcional, mas requerido para algumas implementações. A origem do pacote colocará nesse campo as opções selecionadas. Esse campo é variável em seu tamanho e vai depender das opções definidas pela origem.

- Preenchimento: é o campo para preencher o cabeçalho, mantendo sempre o alinhamento em 32 bits.

4.3 O endereçamento

As redes da atualidade encontram-se quase todas interligadas e são compostas de uma quantidade considerável de equipamentos e hosts integrados. O melhor exemplo dessa integração é a existência da internet, onde temos calculado milhões de hosts interligados através de uma malha complexa de conexões de dados, trocando informações e pacotes.

Essa integração é mérito, principalmente, da estrutura do endereçamento IPv4, ele foi idealizado e implementado com alguns requisitos importantes:

- Cada host é único em relação a seu endereço na rede, não podem existir dois endereços iguais no mesmo segmento.

- As redes podem ser divididas em sub-redes para garantir um gerenciamento eficiente de sua interligação com redes diferentes.

- Possibilidade de envio de informações para diversos hosts a partir de um único pacote.

Uma característica não menos importante no endereçamento do IPv4 é o fato de ele ser hierárquico, ou seja, em uma rede, é possível identificar cada host de uma maneira única, e, com isso, ao juntarmos as redes, estas conseguem se identificar em parte ou em sua totalidade a cada equipamento nó conectado, a partir dos gateways e roteadores, e ainda entregar os pacotes ao seu destino corretamente.

4.4 Classes do protocolo IPv4

O endereçamento IPv4 é classificado da seguinte forma:

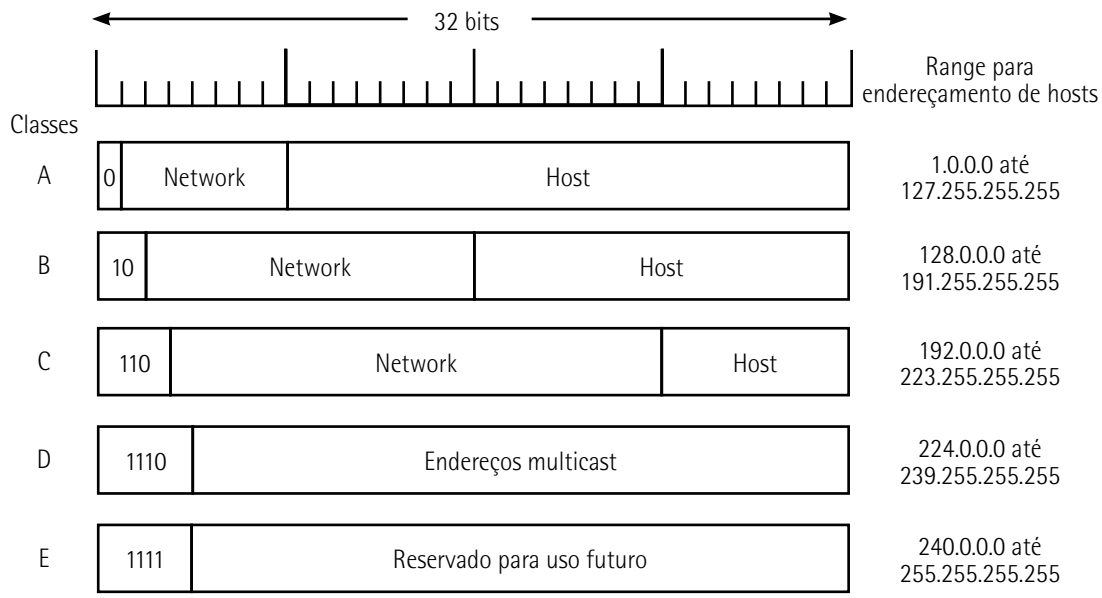


Figura 14 – Classes de endereçamento IPv4

4.5 O cálculo do protocolo IPv4

O endereço IPv4 é representado por uma **palavra** de 32 bits divididos em quatro octetos, assim temos o exemplo:

1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	1	1	0	0	1	0	0	.	0	0	0	0	0	0	0	1

Esses bits podem ser representados em seu formato binário (notação binária) ou em formato decimal (notação decimal) separados por pontos.

Sabendo que cada octeto possui 1 byte de tamanho e 1 byte possui 8 bits, a conversão decimal binária pode ser calculada pelo valor/referência:

- Primeiro bit da esquerda para direita tem o valor decimal = 128.
- Segundo bit da esquerda para direita tem o valor decimal = 64.
- Terceiro bit da esquerda para direita tem o valor decimal = 32.
- Quarto bit da esquerda para direita tem o valor decimal = 16.
- Quinto bit da esquerda para direita tem o valor decimal = 8.

- Sexto bit da esquerda para direita tem o valor decimal = 4.
- Sétimo bit da esquerda para direita tem o valor decimal = 2.
- Oitavo bit da esquerda para direita tem o valor decimal = 1.

O primeiro octeto do exemplo tem o valor binário 11000000 e, somando os valores decimais dos bits **ligados** (com sinalização = 1) e desprezando os bits **desligados** (com sinalização = 0), temos o resultado da conta: 128 + 64 = 192, correspondente ao valor decimal desse octeto.

O segundo octeto tem o valor binário 10101000 e, somando os valores decimais dos bits **ligados** (com sinalização = 1) e desprezando os bits **desligados** (com sinalização = 0), temos o resultado da conta: 128 + 32 + 8 = 168, correspondente ao valor decimal desse octeto.

O terceiro octeto tem o valor binário 01100100 e, somando os valores decimais dos bits **ligados** (com sinalização = 1) e desprezando os bits **desligados** (com sinalização = 0), temos o resultado da conta: 64 + 32 + 4 = 100, correspondente ao valor decimal desse octeto.

O quarto octeto do exemplo tem o valor binário 00000001 e, somando os valores decimais dos bits **ligados** (com sinalização = 1) e desprezando os bits **desligados** (com sinalização = 0), temos o resultado da conta: 1, correspondente ao valor decimal desse octeto.

A representação decimal do exemplo resulta no endereço com notação decimal:

192.168.100.1

O endereço não estará completo se não for calculada a ocorrência de sua máscara de rede, que é a inserção de uma máscara de correspondência binária/decimal para a alocação do segmento lógico da rede, quantidade de hosts possíveis no segmento e identificação do endereço broadcast:

Voltando ao exemplo dado, agora com uma máscara classe C associada:

192.168.100.1/24

A representação do número 24 após uma barra indica que este endereço faz parte de uma classe C, em que os 24 bits mais relevantes (24 bits ligados) correspondem diretamente ao endereçamento da rede e os bits restantes (preenchidos com zeros) representam os hosts pertencentes a esta rede, então teremos uma máscara em notação binária:

1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	0	0	0	0	0	0	0	0

Usando a conversão binária/decimal teremos a notação decimal:

255.255.255.0

Estabelecidas as notações binárias/decimais do endereço, podemos executar o cálculo para descobrir o endereço de rede (que não pode ser usado para hosts e nomina diretamente a qual rede pertence o host) e o endereço broadcast (que o endereço máximo desta classe, por onde acontece o broadcast desta rede e não pode ser usado para nominar hosts).

Primeiro, para calcular o endereço da rede faça o AND BOOLEANO entre a máscara e o endereço IP do exemplo:

1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	1	1	0	0	1	0	0	.	0	0	0	0	0	0	0	1

Represente os 32 bits do endereço IP proposto no exemplo.

1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	X	X	X	X	X	X	X	X

Execute o AND BOOLEANO entre o endereço e a máscara (até o limite de 24 bits do exemplo).

1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	1	1	0	0	1	0	0	.	0	0	0	0	0	0	0	0

Após atingir o limite da máscara, preencha os demais octetos com **zeros**.

1	9	2	.	1	6	8	.	1	0	0	.	0

Represente o endereço de rede em decimal.

Agora, para calcular o broadcast:

1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	1	1	0	0	1	0	0	.	0	0	0	0	0	0	0	1

Represente os 32 bits do endereço IP proposto no exemplo.

1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	X	X	X	X	X	X	X	X

Execute o AND BOOLEANO entre o endereço e a máscara (até o limite de 24 bits do exemplo).

1	1	0	0	0	0	0	0	.	1	0	1	0	1	0	0	0	.	0	1	1	0	0	1	0	0	.	1	1	1	1	1	1	1	1

Após atingir o limite da máscara, preencha os demais octetos com **um**.

1 9 2 . 1 6 8 . 1 0 0 . 2 5 5

Represente o broadcast de rede em decimal.

Já calculamos os endereços de rede e broadcast deste exemplo, agora vamos calcular a quantidade de hosts possíveis, para isto vamos usar uma fórmula simples e muito eficaz:

$$2^n - 2 = \text{hosts}$$

Em que **n** é a quantidade de zeros da máscara, então teremos:

$$2^8 - 2 = 254 \text{ hosts}$$

Então podemos afirmar que neste exemplo é possível distribuir 254 endereços de hosts que são do endereço 192.168.100.1 até o endereço 192.168.100.254, lembrando sempre de excluir desta distribuição o endereço de rede, 192.168.100.0, e o endereço broadcast, 192.168.100.255.

4.6 O protocolo IPv6

Devido ao ritmo acelerado de evolução das redes de computadores, ao ingresso de novos dispositivos móveis, ao crescimento da população com acesso à internet em todas as localidades do planeta, surgiu a necessidade de mais endereços no padrão IP e, com o fim prematuro do protocolo IPv4, tornou-se necessária a evolução desse protocolo.

O endereçamento IPv4, ainda em uso atualmente, não suportou esse crescimento de dispositivos e a demanda de acesso à internet, extinguindo rapidamente os seus recursos de endereçamento. Certamente esse é o principal motivo para a idealização de um novo protocolo de endereçamento que fosse suportado pelos próximos anos, o que levou à criação do protocolo IPv6.

O IPv6 foi projetado para ser o sucessor do IPv4. Ele tem maior espaço de endereços, que desta vez possuem 128 bits, fornecendo 340 undecilhões de endereços. Esse valor é escrito com o número 340 seguido de 36 zeros. Entretanto o IPv6 é muito mais do que números em quantidades maiores. Quando o comitê IETF (Internet Engineering Task Force) iniciou seu desenvolvimento, aproveitou para corrigir muitas das limitações do IPv4 e ainda incluir novos aprimoramentos. Um bom exemplo é o ICMP versão 6, que inclui a resolução de endereço com uma configuração automática, que não é encontrada nos ICMP da versão 4.

Algumas características desse novo protocolo são:

- Maior espaço de endereçamento.
- Mobilidade.

- Segurança.
- Autoconfiguração.
- Compatibilidade com o IPv4.

A redução das reservas de endereços IPv4 certamente foi o principal fator para a criação e migração de um novo protocolo. Conforme continentes como África, Ásia e algumas outras partes do mundo forem se conectando à internet, não haverá endereços IPv4 suficientes para absorver todo esse crescimento.

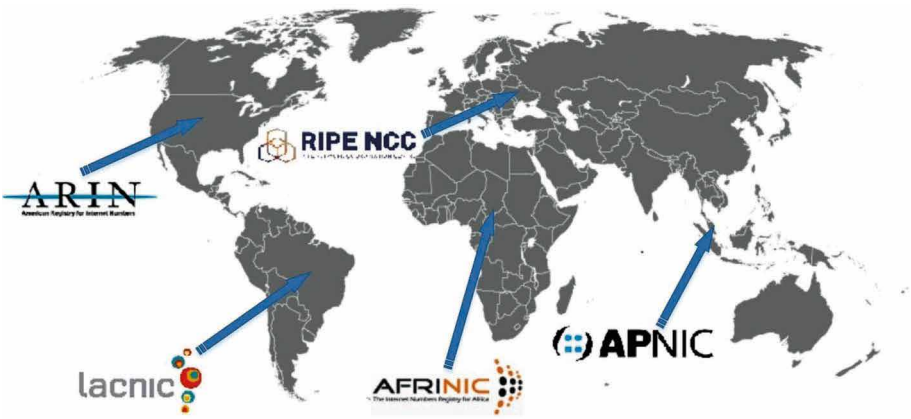


Figura 15 – Mapa da distribuição dos endereços IPv4 pelos continentes

O IPv4, como sabemos, tem um máximo teórico de 4,3 bilhões de endereços possíveis combinados ao NAT (tradução de endereços de rede). Os endereços privados foram imprescindíveis para retardar e conter a redução do espaço dos endereçamentos IPv4, entretanto o NAT danifica o funcionamento de muitos aplicativos e tem determinadas limitações que impedem, principalmente, comunicações ponto a ponto.

4.6.1 A Internet das Coisas (IoT – Internet of Things)

A internet da atualidade é, de longe, muito diferente da internet do passado. Ela é mais do que apenas e-mails, páginas de navegação web, aplicativos e suporte à transferência de arquivos entre os computadores. Ela evolui para uma internet chamada Internet das Coisas (Internet of Things), em que computadores, tablets e smartphones não serão os únicos dispositivos com acesso à internet. Isso, inclusive, já é uma realidade, e os dispositivos que virão no futuro serão equipados com sistemas de sensoriamento já habilitados para acesso à internet, abraçando todo este universo, desde dispositivos biomédicos, automóveis, eletrodomésticos e até sistemas naturais.

Com a alta taxa de crescimento demográfico, naturalmente a internet recebe dia a dia o maior contingente de usuários. Sabendo que os espaços de endereçamento IPv4 chegaram ao ponto limite, e que os problemas com o NAT interferem diretamente nos dispositivos da Internet das Coisas, chegou o momento da transição para o IPv6.

4.6.2 O datagrama do IPv6

Assim como o IPv4, o datagrama IPv6 é composto de duas partes: cabeçalho e dados.

Entre as grandes diferenças das duas versões está justamente o cabeçalho do pacote IPv6, que é mais simples e que foi pensado em otimizar e agilizar o encaminhamento das informações através das redes.

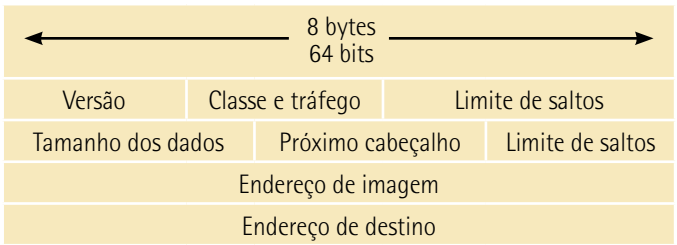


Figura 16 – O cabeçalho do protocolo IPv6

A definição dos campos do cabeçalho é a seguinte:

- Versão: é a versão do protocolo, no caso, 6.
- Classe e tráfego: indica a prioridade do pacote.
- Identificador de fluxo: QoS (Qualidade do Serviço).
- Tamanho dos dados: informa o tamanho da parte de dados do pacote IPv6.
- Próximo header (cabeçalho): é o campo que aponta para o próximo header do IPv6. Essa característica de possuir mais de um header foi criada para simplificar o cabeçalho padrão, e, caso sejam necessárias funções especiais, cabeçalhos extras são alocados e inseridos na parte de dados do pacote IP.
- Limite de saltos: oficializando o que já acontecia com o campo TTL (Tempo de Vida) do IPv4, este campo limita a quantidade de dispositivos que roteiam os pacotes por onde este pacote pode passar. Caso esse número chegue a zero, o pacote é descartado.
- Endereço de origem: é o endereço do dispositivo de origem representado por um campo de 128 bits.
- Endereço de destino: é o endereço do dispositivo de destino representado por um campo de 128 bits.

4.6.3 O endereçamento IPv6

O protocolo IPv6 usa como endereçamento uma **palavra** com 128 bits, capaz de gerar um total de **3.4 * 10^{38}** de endereços possíveis, garantindo uma longevidade considerável.

Da mesma maneira que no protocolo IPv4, a forma de representação do endereçamento do IPv6 não é realizada no formato binário, pois, pelo tamanho, seria muito difícil a sua representação. Então, no IPv6, a representação do endereço é feita pelo agrupamento de 16 em 16 bits separados pelo sinal de dois-pontos (:).

Como demostrado a seguir, o formato preferencial para se escrever um endereço do padrão IPv6 é X:X:X:X:X:X:X:X, com cada X consistindo de quatro valores hexadecimais. Ao falarmos de endereçamento IPv4 nos referenciamos a 8 bits com o termo **octeto**. Entretanto, no IPv6 o termo usado é o **hexteto**, um termo ainda informal e que é empregado basicamente para fazer referência a um segmento de 16 bits, ou 4 valores hexadecimais, sabendo que cada X equivale a um único hexteto, ou 16 bits, ou ainda a 4 dígitos hexadecimais.

Basicamente, o formato preferencial significa que os endereçamentos IPv6 são gravados usando todos os 32 dígitos hexadecimais, entretanto isso não significa que seja o método ideal para representar os endereçamentos em IPv6. A seguir veremos as regras que nos ajudam a reproduzir os números e os dígitos que são necessários e imprescindíveis para a representação de endereço IP versão 6.

Esses grupos de 16 bits são representados usando uma notação hexadecimal, sendo que cada dígito hexadecimal representa 4 bits separados, assim teremos:

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

Onde X é um dígito hexadecimal representado pelos valores (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F).

Exemplos:

FADA:FADA:0000:FFFF:FFFF:4AFD:5EAA1:0000

Ou:

FFFF:0000:0000:0000:0000:0000:0000:0001

Caso existam valores **0** à esquerda do número nos grupos de 16 bits, no momento da representação, esses zeros podem ser suprimidos, exemplo: 001A pode ser representado apenas por 1A.

2017:0000:1F3A:0000:0000:**1A**:2345:5678

Se existirem grupamentos de 4 dígitos zero (**0000**), estes podem ser suprimidos e representados desta forma:

2017:0000:1F3A:::FF1A:2345:5678

Ou:

2017::1F3A:0000:0000:FF1A:2345:5678

Observação

Observe com atenção, pois estará errado representar este endereço como:

2017::1F3A:::FF1A:2345:5678

Porque deve-se observar a sequência de pares dos octetos, ou seja, não podemos resumir octetos quebrados em seus pares com **::**, desta forma a notação hexadecimal ficará incorreta.

O endereçamento IPv6 também especifica três tipos diferentes de endereçamento: o unicast, anycast e o multicast.

4.6.4 Unicast

Endereça apenas uma interface, ou seja, não há mais de uma interface **respondendo** ao mesmo endereço. O endereço IPv6 unicast identifica exclusivamente uma interface de um dispositivo que esteja habilitado para IPv6. Observe na figura a seguir um mecanismo de endereço IP versão 6 origem, que deve ser um endereço unicast.

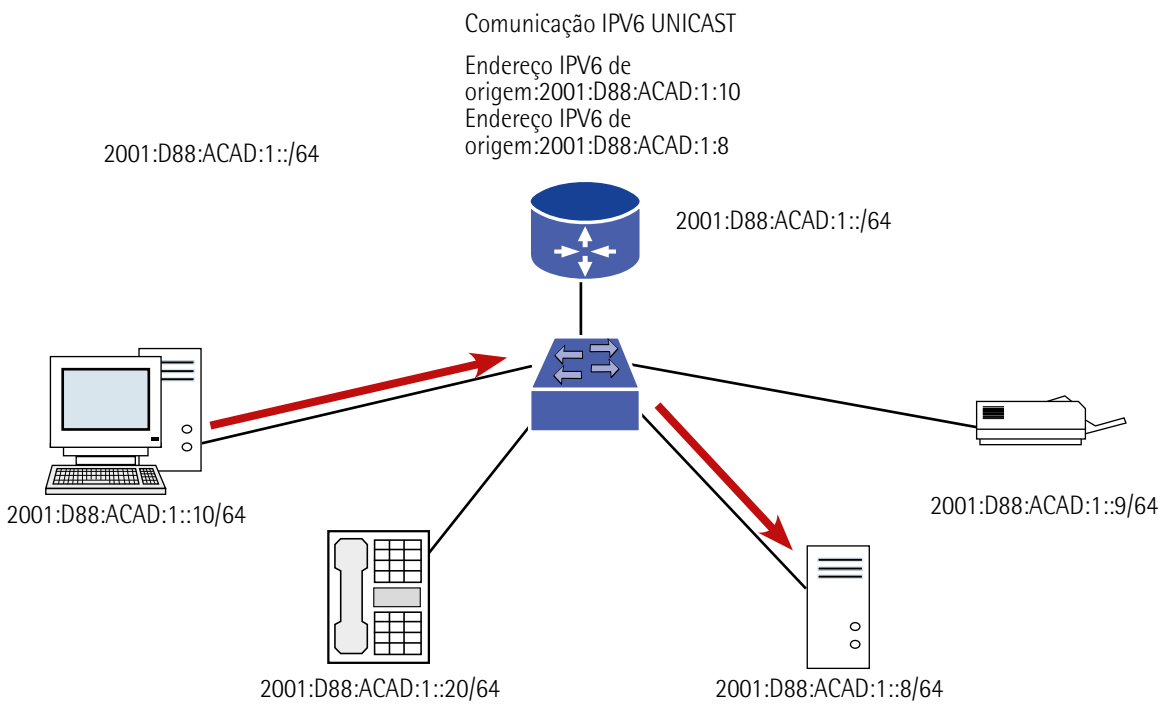


Figura 17 – Atuação do tráfego IPv6 unicast

Comprimento e prefixo do endereço IPv6

Devemos lembrar que o prefixo, que é a parte da rede de endereço padrão IPv4, deve ser identificado pelo comprimento, pela notação em sua barra ou por uma máscara de sub-rede em formato decimal com pontos, um exemplo é o endereço IPv4 192.168.1.10 com a máscara de sub-rede em formato decimal com pontos 255.255.255.0, que é equivalente à notação decimal 192.168.1.10/24.

O endereçamento IPv6 usa um comprimento de prefixo a fim de representar a parte de prefixo do endereço. O IPv6 não utiliza uma notação de máscara de sub-rede decimal com pontos, como acontece no IPv4. O comprimento desse prefixo indica a parte de rede de um endereço IPv6 no formato do endereço IPv6/comprimento do prefixo.

O comprimento do prefixo pode variar de 0 a 128. Um comprimento do prefixo IPv6 padrão para LANs e para a maioria dos outros tipos de redes é /64. Isso significa que o prefixo ou a parte de rede do endereço é de 64 bits, restando outros 64 bits para a ID da interface (parte de host) do endereço.

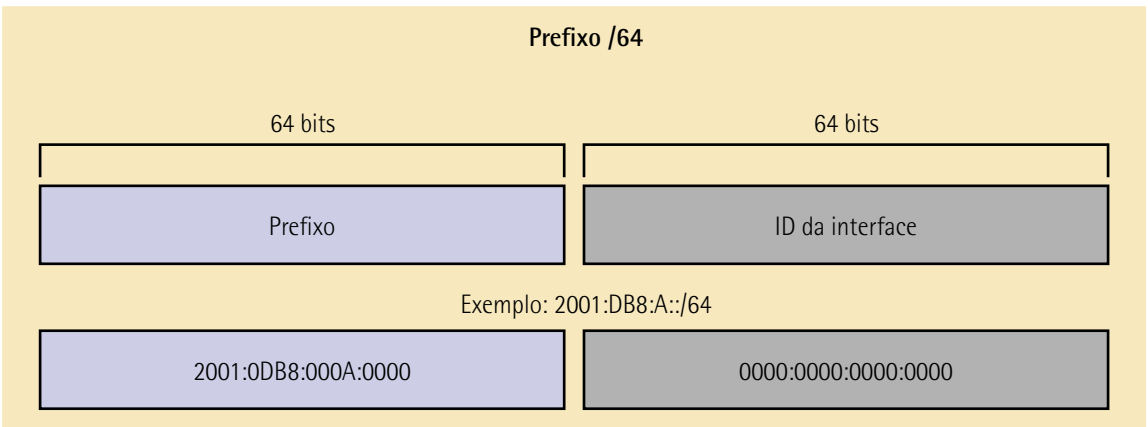


Figura 18 – Entendendo o prefixo dos endereços IPv6

Categorização dos endereços unicast

No momento de representar o endereço IPv6 unicast, este identifica, exclusivamente, uma interface em um tipo de dispositivo que esteja habilitado para IPv6. Um pacote que seja enviado a um endereço unicast é recebido por uma interface atribuída diretamente a esse endereço. Muito semelhante ao IPv4, os endereços IPv6 de origem devem ser um endereço unicast, mas o endereço IPv6 de destino ainda pode ser um endereço unicast ou multicast.

Os tipos mais comuns de endereços IP versão 6 unicast são endereços unicast globais, ou GUA, e os endereços unicast de link local.

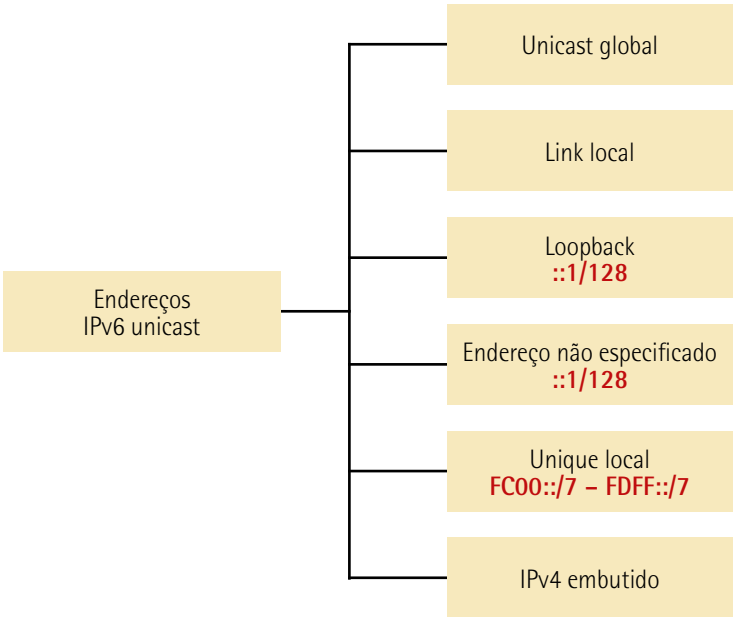


Figura 19 – Tipos de endereços unicast padrão IPv6

Unicast global

O endereço unicast global é bem parecido com o endereço IPv4 público. São endereços de internet basicamente roteáveis e globalmente exclusivos. Os endereços unicast globais podem ser configurados estaticamente ou serem atribuídos em formato dinâmico.

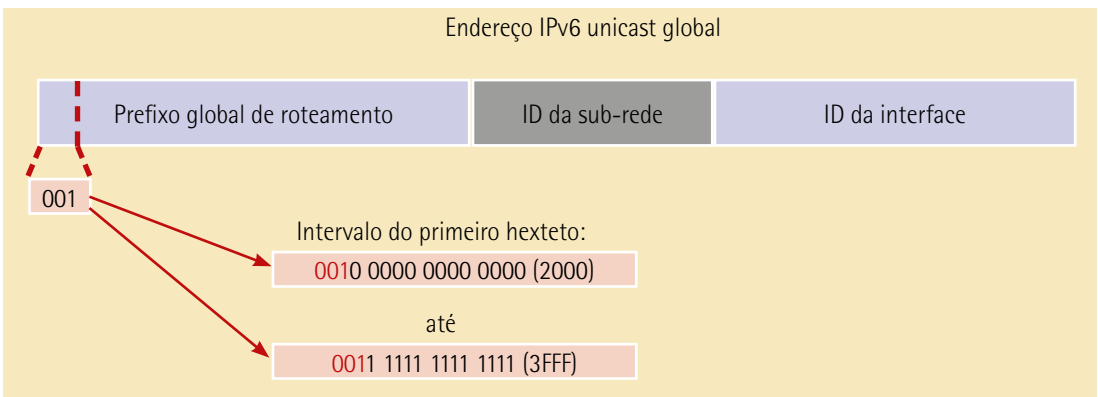


Figura 20 – Endereços unicast globais

Endereços do link local

Os endereços de link local são utilizados para estabelecer a comunicação com outros dispositivos que estejam presentes no mesmo segmento do link local. No caso do IPv6, o termo link refere-se a uma sub-rede e os endereços de link local são limitados a um único link. Essa exclusividade só deve ser

afirmada nesse link porque eles não são roteáveis além do link, ou seja, os roteadores não encaminham pacotes com endereços de link local origem ou destino.

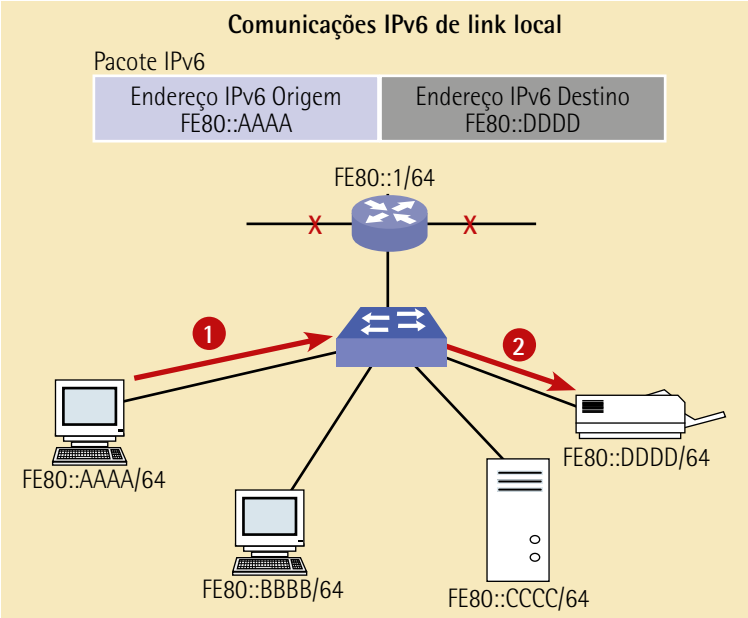


Figura 21 – O estabelecimento do link local em redes operando IPv6

Endereços unique local

Outra classe de endereçamento IPv6 unicast é conhecida como unique local. Os endereços IPv6 unique local possuem certas semelhanças com endereços privados da RFC 1918 para IPv4 (exemplo 127.0.0.1), porém essas semelhanças param por aí. Os endereços unique local são usados para o endereçamento local dentro de um site ou dentro de um número limitado de sites. Esses endereços não devem, em hipótese alguma, serem roteados pelo IPv6 global e nem passar por tradução (NAT) de endereços IPv6 global. Os endereços unique local estão no intervalo FC00::/7 a FDFF::/7.

No endereçamento IPv4, os endereços privados são combinados com mecanismos de tradução de rede ou tradução de porta. Endereços de vários para um, privados para públicos, por exemplo. Isso acontece em função da limitante disponibilidade do espaçamento de endereços IPv4. Muitos sites utilizam mecanismos de natureza privada para endereços RFC 1918 com a intenção clara de proteger sua rede contra potenciais vulnerabilidades à segurança ou até mesmo ocultá-la, entretanto essa técnica nunca foi originalmente definida para estas tecnologias. A IETF recomenda que sites tomem suas devidas precauções de segurança em seu roteador de borda da internet. Os endereços unique local podem ser aplicados tanto para dispositivos que nunca precisarem ou nunca precisaram ou que nunca terão acesso por qualquer outra rede.

Estrutura de um endereço IPv6 unicast global

O endereçamento IPv6 unicast global, como o nome diz, é exclusivo globalmente. Roteável na internet IPv6, estes endereços têm equivalência aos endereços públicos no IPv4. O Internet Commitee for Assigned Names and Numbers (ICANN), operador do Internet Assigned Numbers Authoriry (IANA) para a versao IPv6, designa

e aloca blocos de endereço IPv6 para cinco RIRs (Registro Regional de Internet, entidade que reúne as cinco organizações que regulamentam o uso dos endereços IP pelo mundo, formado por LACNic, ARIN, APNIC, RIPE NCC e AfriNIC). No entanto, atualmente estão sendo distribuídos apenas endereços unicast globais com os primeiros 3 bits iguais a 001 ou 2000::/ 3. Observe que isso reflete apenas um oitavo do espaço de endereços IPv6 disponíveis totais, excetuando uma parte muito pequena de outros tipos de endereços unicast e multicast.

Observação

Importante: o endereço 2001:0DB8::/32 foi reservado para fins de documentação, como a utilização em exemplos.

A figura a seguir mostra a estrutura e a faixa dos endereços unicast globais.

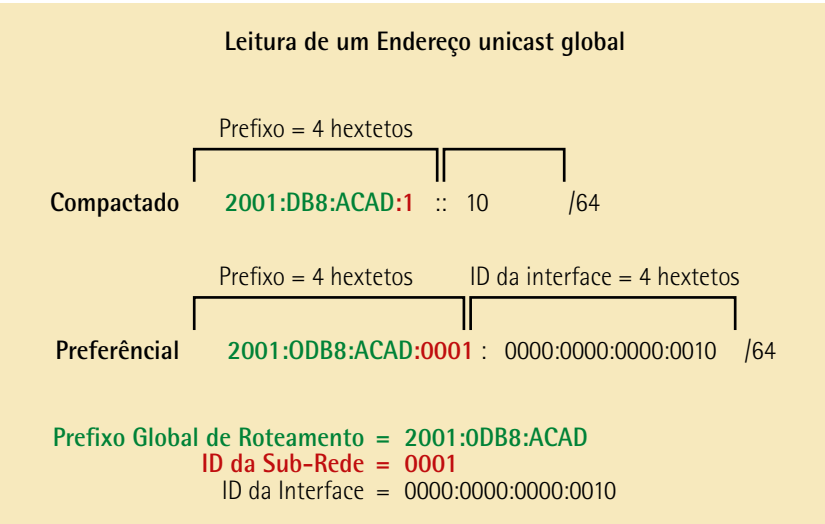


Figura 22 – Exemplo de denominação unicast global

Um endereço unicast global é formado por três partes:

- Prefixo global de roteamento.
- ID da sub-rede.
- ID da interface.

Prefixo global de roteamento

Prefixo de roteamento global é o prefixo parte de rede do endereço IPv6 que é atribuído pelos provedores como uma ISP (Internet Solution Provider), diretamente a um cliente ou a um site. No momento, os RIRs atribuem o prefixo global de roteamento /48 a seus clientes. Nessa visão se inclui todo mundo, partindo de residências até redes corporativas.

A figura a seguir mostra a estrutura de um endereço unicast global usando um prefixo global de roteamento /48. Os prefixos /48 são os prefixos de roteamento global mais comumente atribuídos.

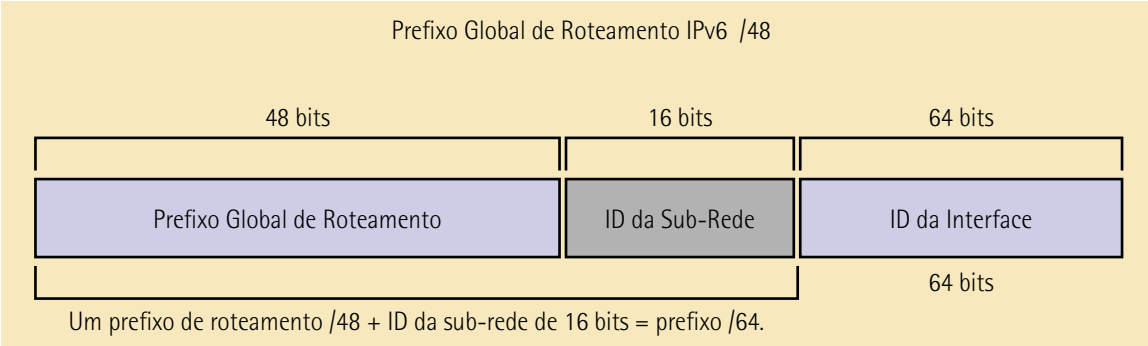


Figura 23 – Exemplo de endereço unicast global /48

Por exemplo, o endereço IPv6 2001:0DB8:ACAD::/48 tem um prefixo que indica que os primeiros 48 bits (3 hextetos: 2001:0DB8:ACAD) são o prefixo ou a parte de rede do endereço. Dois-pontos duplo (::) antes do comprimento de prefixo /48 significa que o restante do endereço contém apenas zeros.

O tamanho do prefixo de roteamento global determina o tamanho da ID da sub-rede.

ID da sub-rede

A ID da sub-rede é empregada por uma empresa para identificar sub-redes localmente. Quanto maior a ID da sub-rede, mais sub-redes disponíveis ela terá.

ID da interface

A ID da interface IPv6 é equivalente à parte de host do endereço IPv4. O termo ID de interface se usa por um único host e pode ter diversas interfaces, cada uma com um ou mais endereços IPv6. É bem provável e também recomendável que as sub-redes /64 sejam as usadas na maioria dos casos.

Observação

Ao contrário do IPv4, é no IPv6 que se encontram todos os endereços de host apenas com zeros ou apenas com 1s. Esses podem ser atribuídos diretamente a um dispositivo. O endereço que possui apenas 1s pode ser usado ainda em função dos endereços broadcast, porém não são usados em IPv6. O endereço apenas com zeros pode ser usado, mas é reservado como um endereço anycast a partir dos roteadores de sub-redes e só deve ser atribuído especificamente a roteadores.

A melhor maneira de ler a maioria dos endereços IPv6 é contando o número de hextetos. Observe na figura a seguir um endereço unicast global /64, os primeiros quatro hextetos são a parte de rede do

endereço, com o quarto hextetos indicando o ID da sub-rede e os quatro hextetos restantes são usados para o ID da interface.

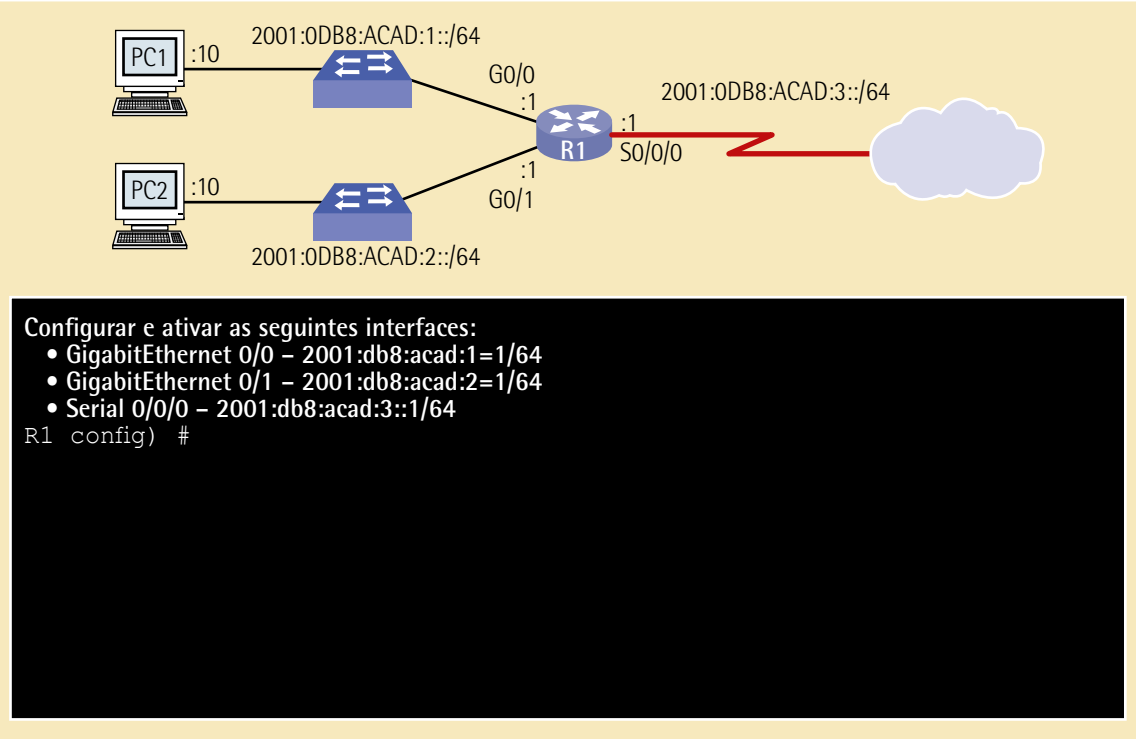


Figura 24 – Atribuição de endereço estático em um roteador

Configuração do roteador com endereçamento IPv6

Boa parte dos comandos de configuração e verificação do IPv6 no Cisco IOS são semelhantes aos seus equivalentes no IPv4. Em várias situações, a única diferença é o uso de IPv6 em vez de ip na linha de comandos.

O comando para configurar um endereço IPv6 unicast global em uma interface é IPv6 address endereço-ipv6 /comprimento-do-prefixo.

O exemplo de configuração usa a topologia mostrada na figura 23 e as seguintes sub-redes IPv6:

2001:0DB8:ACAD:0001:/64 (ou 2001:DB8:ACAD:1::/64)

2001:0DB8:ACAD:0002:/64 (ou 2001:DB8:ACAD:2::/64)

2001:0DB8:ACAD:0003:/64 (ou 2001:DB8:ACAD:3::/64)

Veja que não há espaço entre o endereço IPv6 e o comprimento do prefixo.

A figura mostra também os comandos necessários para configurar o endereço IPv6 unicast global nas interfaces GigabitEthernet 0/0, GigabitEthernet 0/1 e Serial 0/0/0 do roteador R1.

Configuração do host com endereçamento IPv6

Configurar manualmente o endereço IPv6 em um host se parece muito com configurar um endereço IPv4.

Como pôde ser visto na figura 22, o endereço de gateway padrão configurado para PC1 é 2001:DB8:ACAD:1::1. Esse é o endereço unicast global da interface GigabitEthernet de R1 na mesma rede. Como alternativa, o endereço de gateway padrão pode ser configurado para corresponder ao endereço de link local da interface GigabitEthernet. Com certeza qualquer uma das configurações funcionará perfeitamente.

Use o verificador de sintaxe para configurar o endereço IPv6 unicast global.

Do mesmo jeito que ocorre no IPv4, a configuração de endereços estáticos em clientes não favorece a implementação para ambientes maiores. Por esse motivo, a maioria dos administradores de redes IPv6 permite a atribuição dinâmica de endereços IPv6.

Existem duas maneiras de um dispositivo obter um endereço IPv6 unicast global automaticamente:

- Configuração automática de endereço stateless (Slaac).
- DHCPv6.

Observação

Quando DHCPv6 ou Slaac é atribuído, o endereço de link local do roteador local é especificado automaticamente como o endereço de gateway padrão.

Configuração dinâmica – Slaac

A melhor forma de implementar a configuração automática de endereço stateless (Slaac) é o método que indica que um dispositivo consiga registrar o prefixo, o comprimento do prefixo, o endereço do gateway padrão e outras informações sobre um roteador IPv6 sem usar um servidor DHCPv6. Com o método Slaac, os dispositivos esperam das mensagens ICMPv6 de RA (Anúncio de Roteador) do roteador local para obter essas informações indispensáveis.

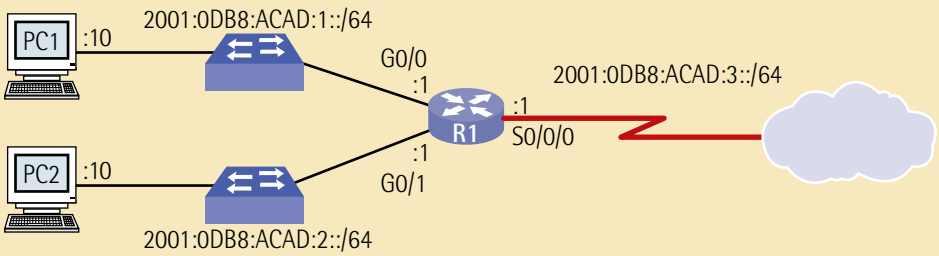
Os roteadores IPv6 transmitem mensagens ICMPv6 de RA a cada 200 segundos para todos os dispositivos habilitados no segmento para IPv6 na rede. Uma mensagem ou instrução de RA também é enviada em resposta a um host que envie uma mensagem ICMPv6 de RS (Solicitação de Roteador).

Sabemos que o roteamento IPv6 não é ativado e habilitado por padrão. Para habilitar um roteador como roteador IPv6, deve ser usado o comando de configuração global IPv6 unicast-routing.

Observação

É possível habilitar e disponibilizar endereços IPv6 em um roteador sem que ele seja um roteador IPv6.

Configurando IPv6 em um roteador




```
R1 (config) # interface gigabitethernet 0/0
R1 (config-if) # ipv6 address 2001:db8:acad:1::1/64
R1 (config-if) # no shutdown
R1 (config-if) # exit
R1 (config) # interface gigabitethernet 0/1
R1 (config-if) # ipv6 address 2001:db8:acad:2::1/64
R1 (config-if) # no shutdown
R1 (config-if) # exit
R1 (config-if) # interface serial 0/0/0
R1 (config-if) # ipv6 address 2001:db8:acad:3::1/64
R1 (config-if) # clock  rate 56000
R1 (config-if) # no shutdown
```

Figura 25 – Método de configuração de um roteador na habilitação do protocolo IPv6

A mensagem ICMPv6 de RA é uma dica para um dispositivo de como conseguir um endereço IPv6 unicast global. A resposta final é do sistema operacional do dispositivo. A mensagem ICMPv6 de RA inclui:

- Prefixo de rede e comprimento do prefixo: informa ao dispositivo a que rede ele pertence.
- Endereço do gateway padrão: é um endereço IPv6 de link local, o endereço IPv6 origem da mensagem de RA.
- Endereços DNS e nome de domínio; endereços de servidores DNS e um nome de domínio.

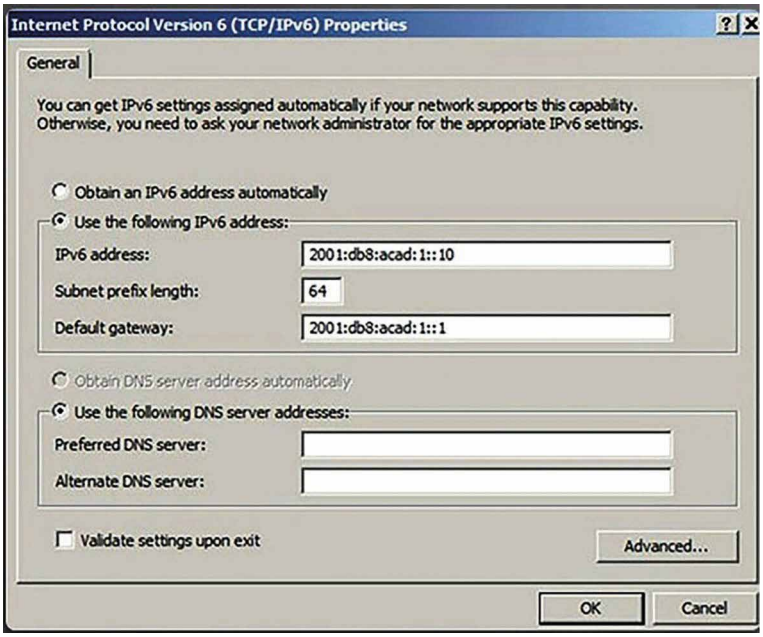


Figura 26 – Janela de configuração do protocolo IPv6 em sistema operacional Windows

Como no exemplo a seguir, existem três tipos de mensagens de RA:

- Opção 1: Slaac.
- Opção 2: Slaac com servidor DHCPv6 stateless.
- Opção 3: DHCPv6 stateful (sem Slaac).

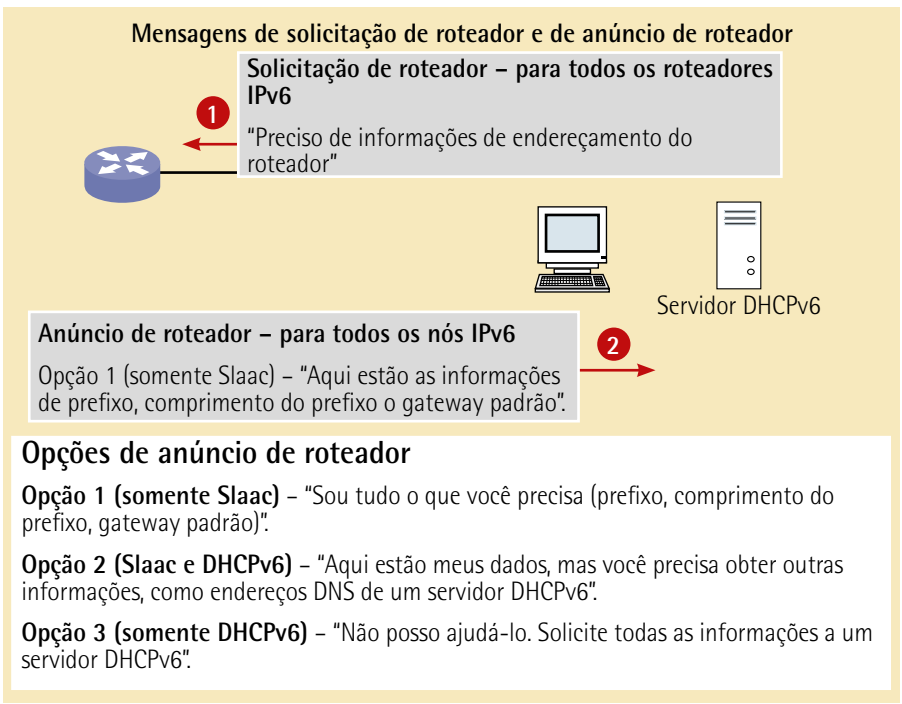


Figura 27 – Diagrama de envio de mensagem Slaac do protocolo IPv6

Opção 1 de RA: Slaac

Por padrão, todas as mensagens RA indicam que um dispositivo de recebimento use determinadas informações dessa mensagem para obter seu próprio endereço IPv6 unicast global. Com ela, todas as demais informações são enviadas junto, porém, em caso de servidores DHCPv6, estas não serão obrigatórias.

A atribuição Slaac é stateless, o que significa que não existe um servidor central (exemplo, um servidor DHCPv6 stateful) alocando e reservando endereços unicast globais e mantendo e registrando uma lista de dispositivos e seus endereços. Com Slaac, o dispositivo cliente usa informações da mensagem de RA para obter seu próprio endereço unicast global.

Opção 2 de RA: Slaac e DHCPv6 stateless

Nesta opção, a mensagem de RA indica que os dispositivos usem:

- O protocolo Slaac para criar seu próprio endereço IPv6 unicast global.
- O endereço de link local do roteador, ou seja, o endereço IPv6 origem da RA para o endereço de gateway padrão.
- Um servidor DHCPv6 stateless para obter outras informações, como o endereço de um servidor DNS e um nome de domínio.

Um servidor DHCPv6 stateless distribui endereços do servidor DNS e nomes de domínio. Ele não aloca endereços unicast globais.

Opção 3 de RA: DHCPv6 stateful

O DHCPv6 stateful é bem parecido com o funcionamento do DHCP para IPv4. Um dispositivo pode receber a atribuição automática de seu endereço, como endereço global unicast, comprimento do prefixo e endereços de servidores DNS, usando os serviços de um servidor DHCPv6.

Nesta opção, a mensagem de RA indica que os dispositivos usem:

- O endereço de link local do roteador, ou seja, o endereço IPv6 origem da RA para o endereço de gateway padrão.
- Um servidor DHCPv6 stateful para obter o endereço unicast global, o endereço do servidor DNS, o nome do domínio e todas as demais informações.

Um servidor DHCPv6 stateful aloca e mantém uma lista dos dispositivos que recebem endereços IPv6. O DHCP para IPv4 é stateful.

Observação

O endereço de gateway padrão só pode ser obtido dinamicamente da mensagem de RA. O servidor DHCPv6 stateless ou stateful não fornece o endereço de gateway padrão.

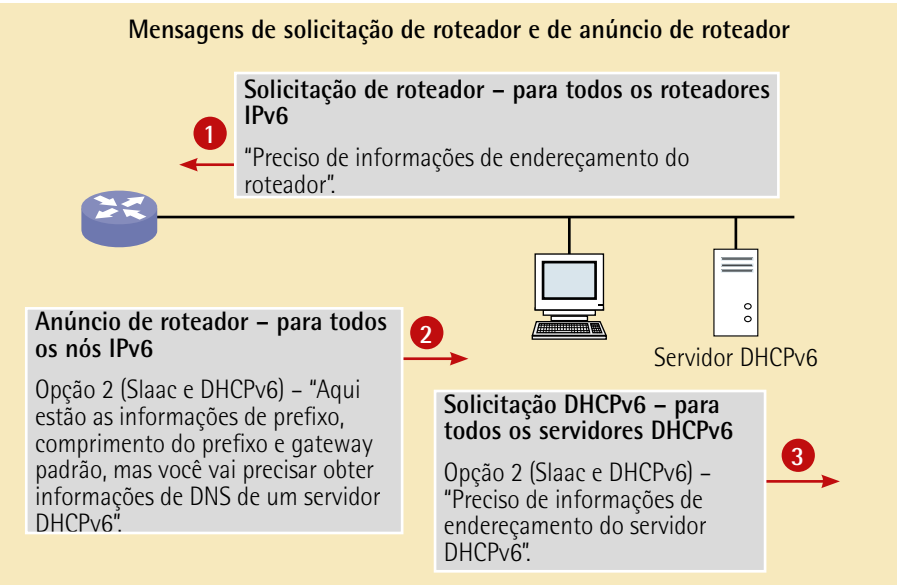


Figura 28 – Mensagens de solicitação de um roteador para opções Slaac do protocolo IPv6

- Prefixo: recebido na mensagem de RA.
- ID da interface: usa o processo de EUI-64 ou gera um número aleatório de 64 bits.

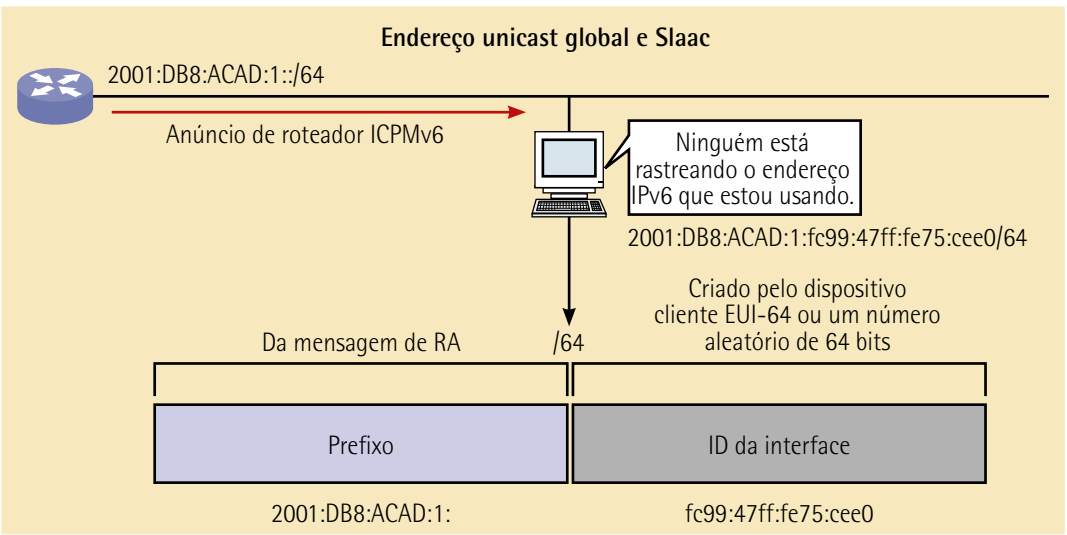


Figura 29 – Obtenção do endereço unicast global no método Slaac do protocolo IPv6

Configuração dinâmica – DHCPv6

No comportamento de um serviço DHCPv6, por padrão, uma mensagem de RA é sempre:

- Opção 1: somente Slaac. A interface do roteador pode ser configurada para enviar um anúncio de roteador usando Slaac e DHCPv6 stateless ou somente DHCPv6 stateful.

- Opção 2 de RA: Slaac e DHCPv6 stateless. Nesta opção, a mensagem de RA sugere que os dispositivos usem:

  – Slaac para criar seu próprio endereço IPv6 unicast global.

  – O endereço de link local do roteador, ou seja, o endereço IPv6 origem da RA para o endereço de gateway padrão.

  – Usar um servidor DHCPv6 stateless para obter outras informações, como o endereço de um servidor DNS e um nome de domínio.

  – O uso de um serviço DHCPv6 stateless distribui endereços do servidor DNS e nomes de domínio. Ele não registra endereços unicast globais.

- Opção 3 de RA: DHCPv6 stateful. O DHCPv6 stateful é bem parecido com o serviço DHCP usado em IPv4. Um dispositivo pode receber automaticamente suas informações de endereçamento, como endereço global unicast, comprimento do prefixo e endereços de servidores DNS, usando os serviços de um servidor DHCPv6. Nesta opção, a mensagem de RA passa a sugestão de que os dispositivos usem:

  – O endereço de link local do roteador, digamos, o endereço IPv6 origem da RA, para o endereço de gateway padrão.

  – Um servidor DHCPv6 stateful para conseguir o endereço unicast global, o endereço do servidor DNS, o nome do domínio e as demais informações.

  – Um servidor DHCPv6 stateful que registre e mantenha uma lista dos dispositivos que recebem endereços IPv6. O DHCP para IPv4 é stateful.

Observação

O endereço de gateway padrão só pode ser conseguido dinamicamente da mensagem de RA. O servidor DHCPv6 stateless ou stateful não oferece o endereço de gateway padrão.

Quando uma mensagem enviada de um RA é Slaac ou Slaac com DHCPv6 stateless, o cliente deve gerar sua própria ID da interface. O cliente conhece a parte de prefixo do endereço da mensagem de

RA, mas precisa criar sua própria ID da interface. A ID da interface pode ser criada por meio do processo EUI-64 ou de um número de 64 bits gerado aleatoriamente, como mostrado a seguir.

Processo EUI-64

A IEEE atribuiu o identificador exclusivo estendido (EUI) ou processo EUI-64 modificado. Esse processo usa o endereço MAC Ethernet de 48 bits de um cliente e insere outros 16 bits no meio do endereço MAC de 48 bits para criar uma ID da interface de 64 bits.

Geralmente representados em hexadecimal, os endereços MAC de Ethernet são compostos de duas partes:

- Identificador exclusivo da organização (OUI): o OUI é um código de 24 bits do fornecedor (6 dígitos hexadecimais) atribuído pela IEEE.
- Identificador de dispositivo: o identificador de dispositivo é um valor exclusivo de 24 bits (6 dígitos hexadecimais) com um OUI em comum.

Uma ID da interface EUI-64 é representada em binário e composta por três partes:

- OUI de 24 bits do endereço MAC do cliente, mas o sétimo bit (o bit universal/local (U/L)) é invertido. Isso significa que, se o sétimo bit for 0, ele se tornará 1, e vice-versa.
- O valor de 16 bits FFFE (em hexadecimal) inserido.
- Identificador de dispositivo de 24 bits do endereço MAC do cliente.

O processo EUI-64 está demonstrado na figura a seguir, usando o endereço MAC GigabitEthernet de R1 FC99:4775:CEE0.

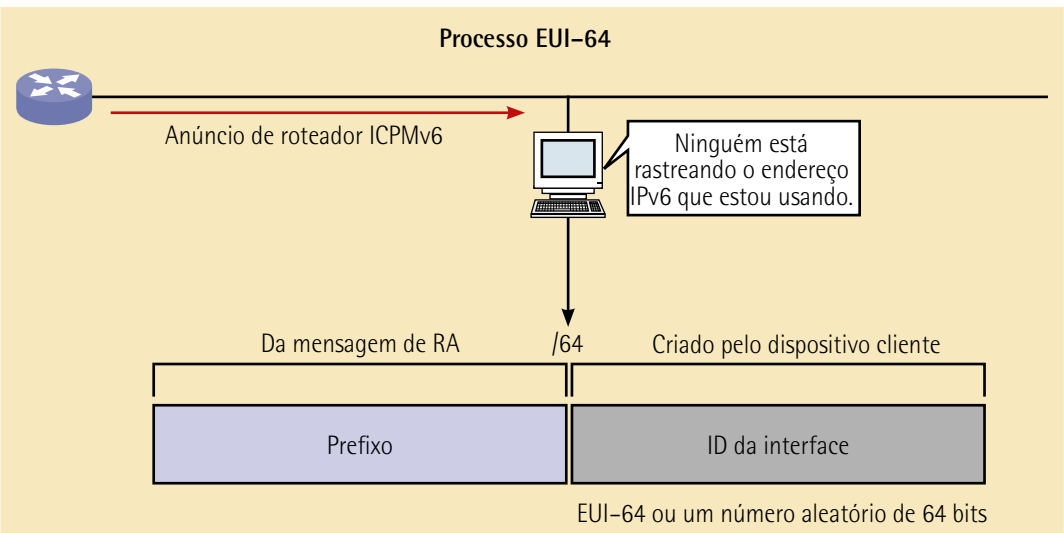


Figura 30 – Processo EUI-64 para protoloco IPv6

Os passos do processão são:

- Fase 1: divida o endereço MAC entre o OUI e o identificador de dispositivo.
- Fase 2: inserir o valor hexadecimal FFFE, em que o binário é 1111 1111 1111 1110.
- Fase 3: converter os dois primeiros valores hexadecimais do OUI em binário e ainda inverter o bit de U/L (bit 7). No exemplo, o 0 do bit 7 é alterado para 1.

O resultado é uma ID da interface FE99:47FF:FE75:CEE0 gerada pelo EUI-64.

Observação

O uso do bit U/L e os motivos para inverter seu valor são discutidos na RFC 5342.

A figura a seguir mostra um endereço IPv6 unicast global de PCA criado dinamicamente com Slaac e o processo EUI-64. O jeito mais fácil de identificar que um endereço foi criado com o uso do EUI-64 é o FFFE localizado no meio da ID da interface:

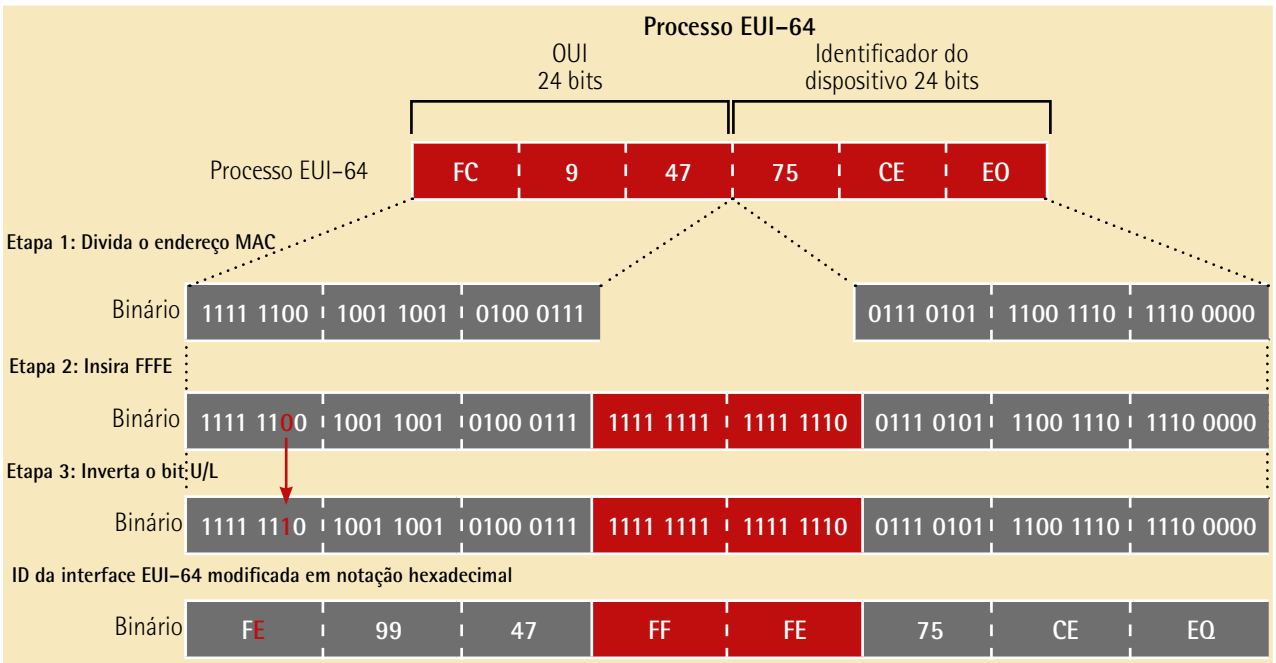


Figura 31 – Processo EUI-64 de atribuição de endereços IPv6

A grande vantagem do método EUI-64 é o endereçamento MAC Ethernet, que pode ser empregado para determinar o ID da interface. Pode também permitir que os administradores da rede tenham o rastreio fácil a um determinado endereço IPv6 para um dispositivo final, usando por exemplo o MAC

exclusivo. Porém isto gera grande preocupação com a privacidade dos usuários, a pergunta é: os pacotes poderão ser rastreados até o computador físico real? Em função dessas preocupações, podemos gerar uma ID de interface aleatória, descaracterizando o rastreio por obscuridade.

IDs da interface geradas aleatoriamente

Dependendo do tipo de sistema operacional, um dispositivo final pode usar uma ID de interface gerada aleatoriamente em vez de usar o endereço MAC address próprio. O processo usado é o EUI-64. Um exemplo é o caso do Windows Vista ou versões posteriores que usam uma ID de interface gerada de forma aleatória em vez de uma criada com o processo EUI-64, já Windows XP e sistemas operacionais Windows anteriores usam apenas um método EUI-64.

Depois que o ID da interface for estabelecido, seja pelo processo EUI-64 ou por geração aleatória, ele pode ser combinado a um prefixo IPv6, vindo da mensagem do RA para criar o endereço unicast global. Observe com atenção a figura a seguir.

Observação

Para garantir a exclusividade de qualquer endereçamento IPv6 unicast, o cliente faz uso de um processo conhecido como detecção de endereço em duplicidade DAD. Esse processo é semelhante a uma requisição ARP para o seu próprio endereço, caso não haja resposta, significa que o endereço é exclusivo.

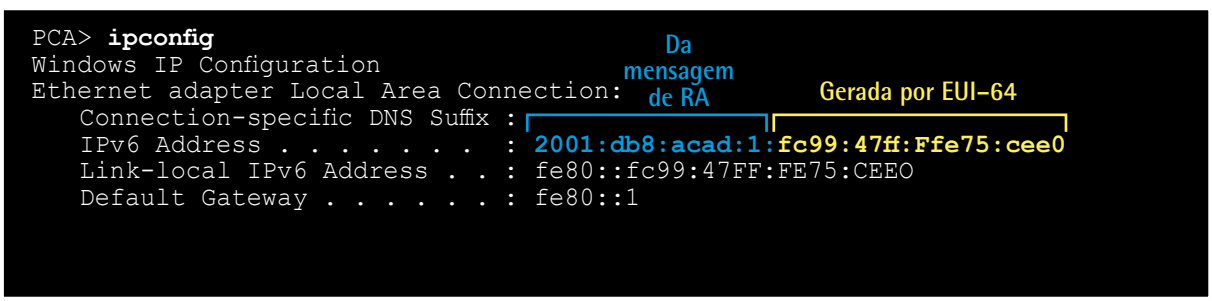


Figura 32 – Fase 1 para a detecção do método de verificação de duplicidade do endereço IPv6

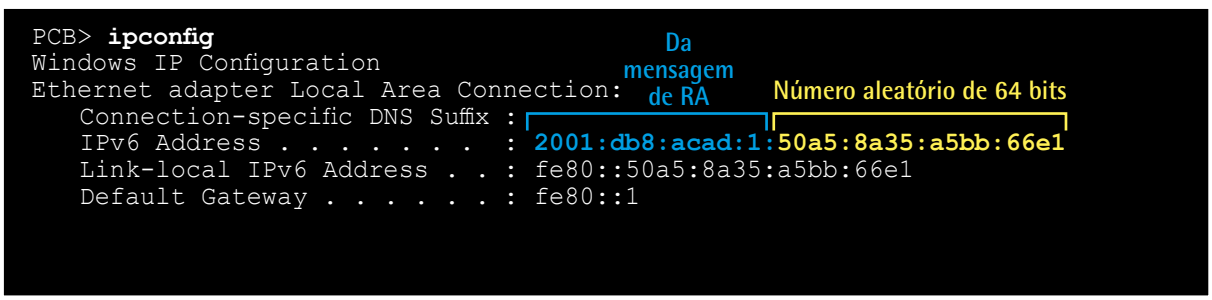


Figura 33 – Fase 2 para a detecção do método de verificação de duplicidade do endereço IPv6

Saiba mais

Segue a dica de uma leitura indispensável se você quiser saber mais sobre o protocolo IPv6:

TANEMBAUM, A. S. IPv6. *In*: ___. *Redes de computadores*. 4. ed. São Paulo: Campus, 2003. p. 357.

Endereços de link local dinâmicos

Todo e qualquer dispositivo com endereçamento IPv6 deve ter um endereço IPv6 de link local. Um endereço de link local é estabelecido dinamicamente ou configurado manualmente como um endereço de link local do tipo estático.

A figura a seguir exemplifica o endereço de link local que é criado dinamicamente com o prefixo FE80::/10 e cuja ID de interface foi criada pelo método EUI-64 ou por um número de 64 bits que foi gerado aleatoriamente. Usualmente, os sistemas operacionais usam um método parecido para estabelecer o endereço unicast global criado pelo processo Slaac e o endereço de link local atribuído em formato dinâmico, como mostra a figura.

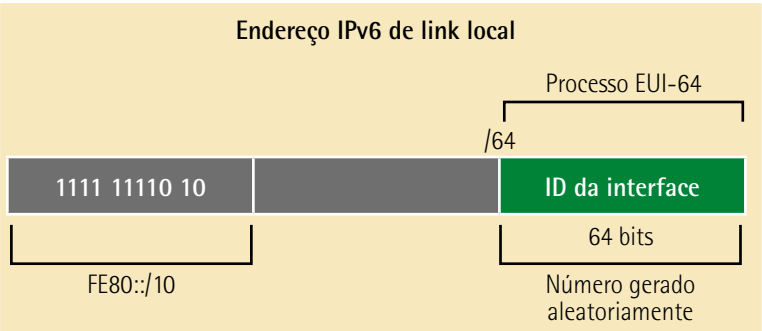


Figura 34 – Estabelecimento de um endereço de link local

A classe de roteadores cisco tem a habilidade de criar automaticamente um endereço de IPv6 de link local sempre que o endereço unicast global for atribuído a uma interface. Baseado nesse padrão, os roteadores cisco usam o método EUI-64 para gerar a ID de interface de todos os endereços de link local em interface do IPv6. No caso das interfaces seriais, o roteador usará o endereço MAC da interface Ethernet. Vamos lembrar que o endereço de link local precisa ser exclusivo, somente desse link ou na rede. Uma das desvantagens em usar um endereço do link local que seja atribuído dinamicamente é o seu tamanho ou comprimento, o que faz com que seja um desafio identificar e lembrar os endereços a ele atribuídos. A figura a seguir mostra um endereço MAC da interface GigabitEthernet 0/0 do roteador R1. Lembrando que esse endereço foi criado para ser usado dinamicamente, atribuindo o endereço do link local na própria interface.

Endereço de link local gerado com EUI-64 do roteador

```
R1# show interface gigabitethernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is fe99.4775.c3e0
(bia fc99.4775.c3e0)

R1# show ipv6 intercace brief
GigabitEthernet0/0     [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:1::1
GigabitEthernet0/1     [up/up]
    FE80::FE99:47FF:FE75:C3E1
    2001:DB8:ACAD:2::1
Serial0/0/0            [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:3::1
Serial0/0/1            [Administratively down/down]
    unassigned
R1#
```

Endereços de link local usando EUI-64

Figura 35 – Método de geração de endereços EUI-64 em interface IPv6

Para ficar fácil reconhecer esses endereços em roteadores e lembrar deles, é comum configurar estaticamente endereços IPv6 de link local nos roteadores.

Endereço de link local criado dinamicamente

ID da interface gerada com EUI-64

```
PCA> ipconfig
Windows IP configuration
Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix   :
   IPv6 Address . . . . . . . . . . : 2001:db8:acad:1:fc99:47ff:fe75:cee0
   Link-local IPv6 Address. . . . . . . . : fe80::fc99:47ff:fe75:cee0
   Default Gateway . . . . . . . . . : fe80::1
```

ID da interface gerada aleatoriamente com 64 bits

```
PCB> ipconfig
Windows IP configuration
Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix   :
   IPv6 Address . . . . . . . . . . : 2001:db8:acad:1:50a5:8a35:a5bb:66e1
   Link-local IPv6 Address . . . . . . . . : fe80::50a5:8a35:a5bb:66e1
   Default Gateway . . . . . . . . . : fe80::1
```

Figura 36 – Método dinâmico de criação de endereços IPv6

Endereços de link local estáticos

Configurar manualmente o endereço de link local permite criar um endereço reconhecível e fácil de lembrar.

O endereço de link local é possível ser atribuído manualmente com o mesmo tipo de comando de interface usado para criar endereços IPv6 unicast globais, mas com um parâmetro link-local adicional. Quando um endereço começa com esse hexteto dentro do intervalo FE80 a FEBF, o parâmetro de link local deve seguir o endereço sequencialmente.

A figura a seguir mostra a configuração de um endereço de link local ao usar o comando de interface IPv6 address. O endereço de link local FE80::1 é empregado para ser rapidamente reconhecido como pertencente ao roteador R1. O mesmo endereço IPv6 de link local é atribuído em todas as interfaces de R1 e ainda pode ser configurado em cada link porque só precisa ser exclusivo nesse link.

Configuração de endereços de link local em R1

```
Router (config-if)#
ipv6 address link-local-address link-local
```

```
R1# (config) # interface gigabitethernet 0/0
R1# (config) # ipv6 address fe80::1 ?
    link-local  Use link-local address

R1# (config) # ipv6 address fe80::1 link-local
R1# (config) # exit
R1# (config) # interface gigabitethernet 0/1
R1# (config) # ipv6 address fe80::1 link-local
R1# (config) # exit
R1# (config) # interface serial 0/0/0
R1# (config) # ipv6 address fe80::1 link-local
R1# (config) #
```

Figura 37 – Configurando endereços link local em roteadores Cisco

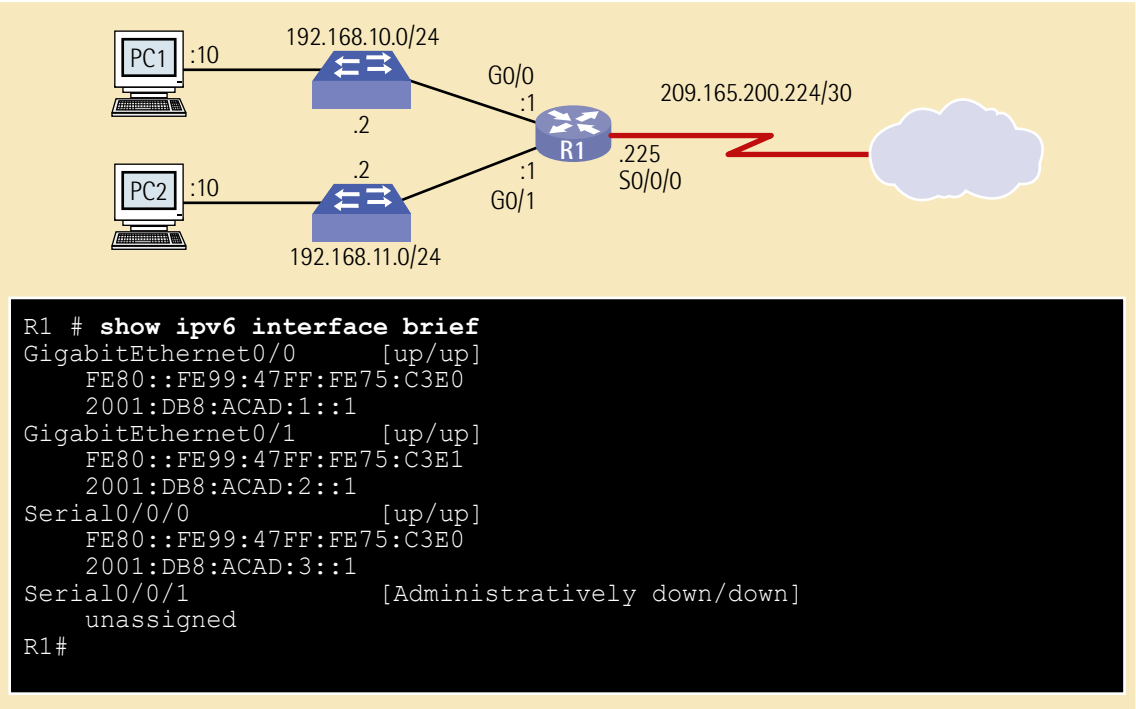
Bem parecido com o roteador R1, o roteador R2 deverá ser configurado com FE80::2 como endereço IPv6 de link local em todas as interfaces.

Verificando a Configuração de Endereço IPv6

Como demonstrado, o comando para verificar a configuração da interface IPv6 é parecido com o comando usado para IPv4.

O comando show interface mostra o endereço MAC das interfaces Ethernet. EUI-64 usa esse endereço MAC para gerar a ID da interface para o endereço de link local. Ainda, o comando show IPv6

interface brief, exibe a saída abreviada para cada uma das interfaces. A saída [up/up] na mesma linha que a interface indica o estado da Camada 1/Camada 2 da interface. O mesmo ocorre nas colunas Status e Protocolo no comando IPv4 em equivalência.




```
R1 # show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:1::1
GigabitEthernet0/1      [up/up]
    FE80::FE99:47FF:FE75:C3E1
    2001:DB8:ACAD:2::1
Serial0/0/0             [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:3::1
Serial0/0/1             [Administratively down/down]
    unassigned
R1#
```

Figura 38 – Exemplo do status da interface IPv6

Observe e analise nesse exemplo que cada interface possui dois endereços IPv6, em que o segundo endereço de cada interface é o endereço unicast global que foi atribuído, e o primeiro endereço, que começa com FF80, é o endereço do link unicast da interface. Vamos lembrar que o endereço de link local deve ser automaticamente adicionado à interface quando o endereço unicast global for designado.

Ainda observando que o endereço link local serial 0/0 do roteador R1 será o mesmo da sua interface GigabitEthernet 0/0, como interfaces seriais não possuem endereço MAC Ethernet, o roteador Cisco usa o endereço MAC da primeira interface Ethernet que está disponível. Isso só é possível porque as interfaces locais precisam apenas ser exclusivas neste link.

Já o endereço de link local da interface dos roteadores geralmente são o endereço de Gateway padrão para dispositivos do link ou da rede.

Observe na figura a seguir o comando show IPv6 route, que pode ser empregado para verificar se foram habilitadas redes IPv6 e endereços IPv6 específicos na tabela de roteamento usada no protocolo IPv6. O comando show IPv6 route passa a exibir apenas endereços IPv6 e não mais endereços IPv4.

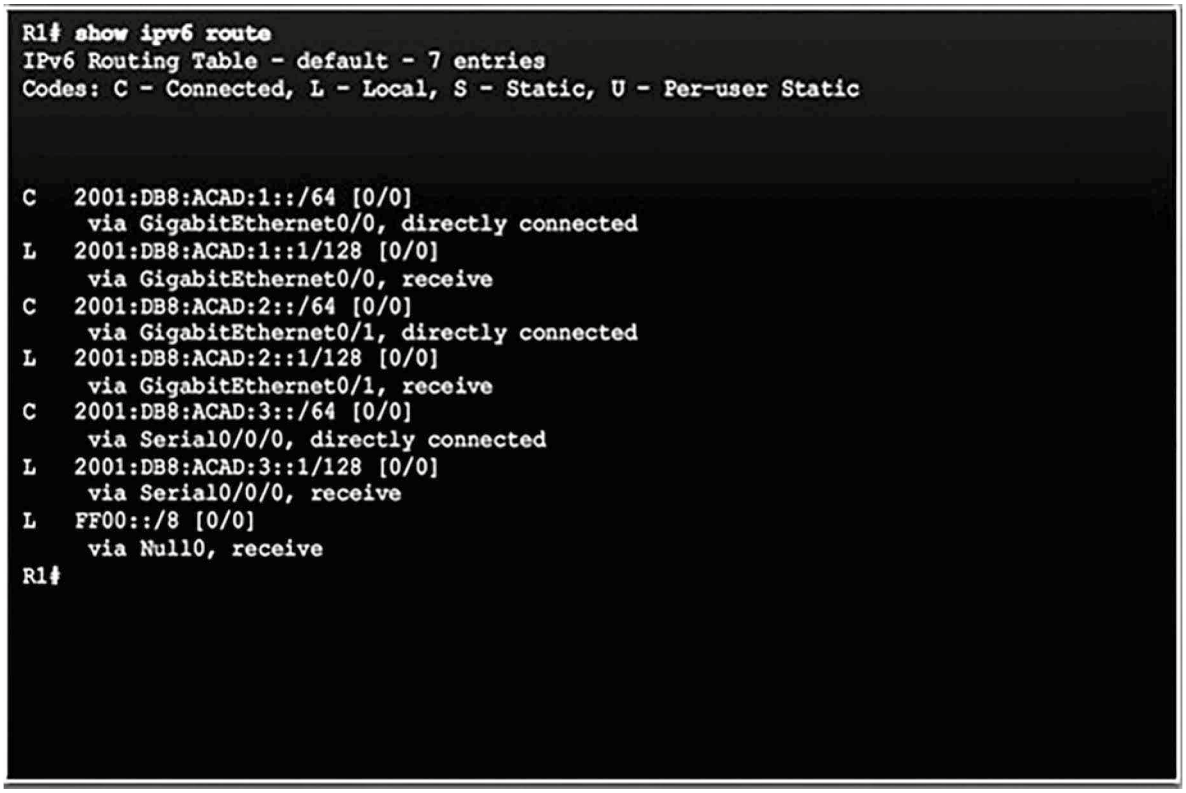


Figura 39 – Identificando métodos de roteamento e atribuição de endereçamento em roteadores classe Cisco

Veja na tabela de rotas, uma letra C ao lado de uma rota indica que se trata de uma rede diretamente conectada. Quando a interface de um roteador está configurada com um endereço unicast global e se encontra no estado **up/up**, o prefixo IPv6 e o comprimento do prefixo são mostrados na tabela de roteamento IPv6 como uma rota conectada.

Já o endereço IPv6 unicast global configurado na interface também é mostrado na tabela de roteamento como uma rota local. A rota local tem um prefixo /128 neste exemplo. As rotas locais são usadas pela tabela de roteamento para processar de forma eficiente pacotes com um endereço destino igual ao endereço da interface do roteador.

O comando ping IPv6 será idêntico ao comando usado em IPv4, exceto pelo fato de ser usado um endereço IPv6 em vez de IPv4. Veja na figura a seguir: o comando serve para confirmar a conectividade da Camada 3 entre o roteador R1 e o computador PC1. Ao executar o ping de um roteador para um endereço de link local, o sistema operacional Cisco vai solicitar que o usuário escolha a interface de saída. Como o endereço de link local de destino pode estar em um ou mais de seus links ou redes, o roteador precisa saber para qual interface enviar o ping.

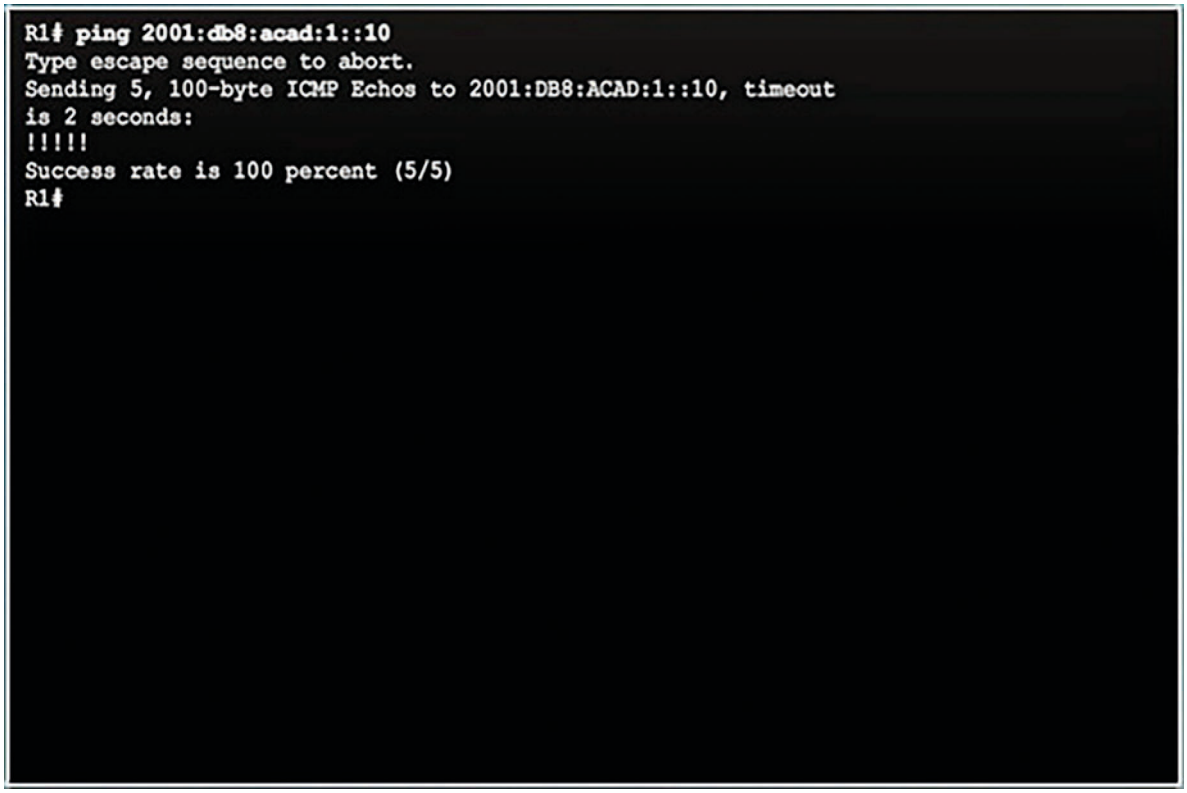


Figura 40 – Executando o ping em formato IPv6

Você pode usar o Verificador de Sintaxe demonstrado na figura a seguir para conferir a configuração do endereço IPv6.

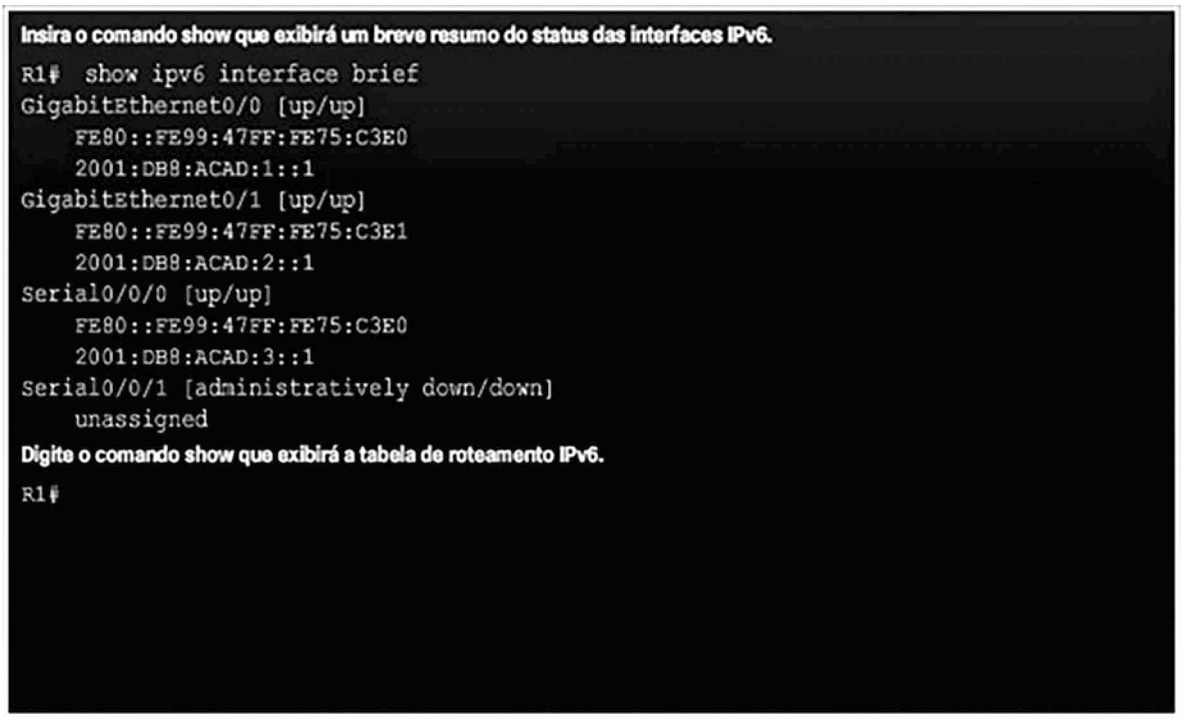


Figura 41 – Usando o verificador de sintaxe na interface IPv6, tela 1

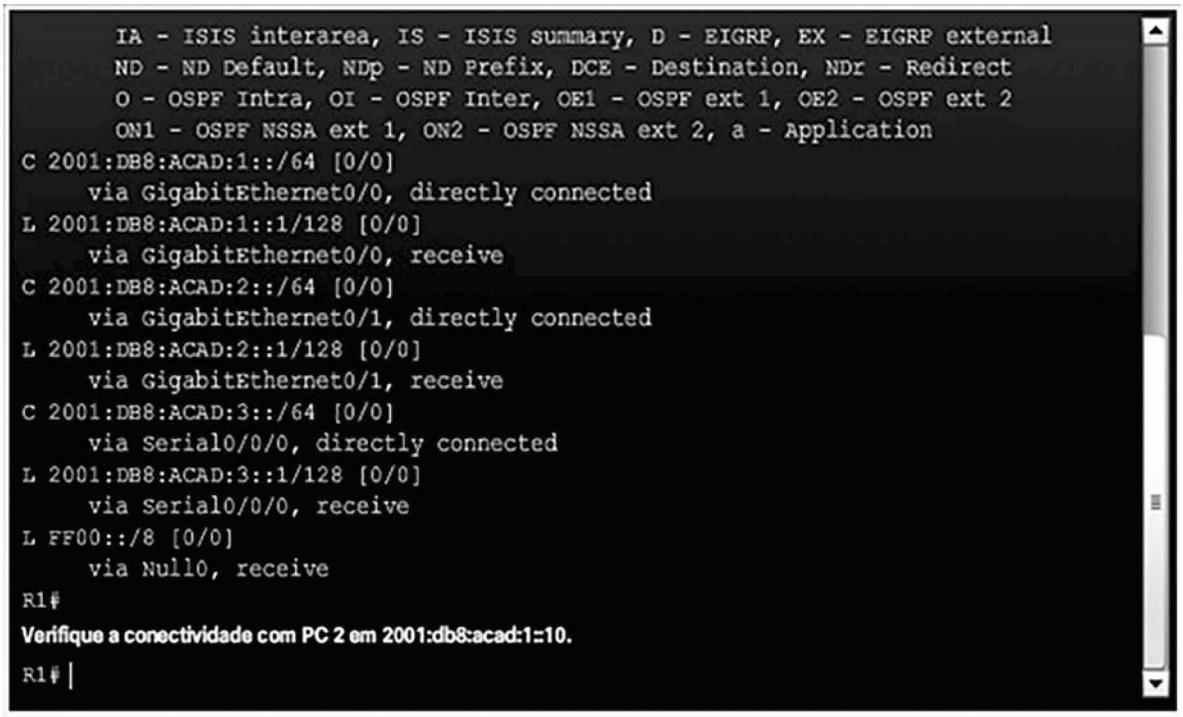


Figura 42 – Usando o verificador de sintaxe na interface IPv6, tela 2

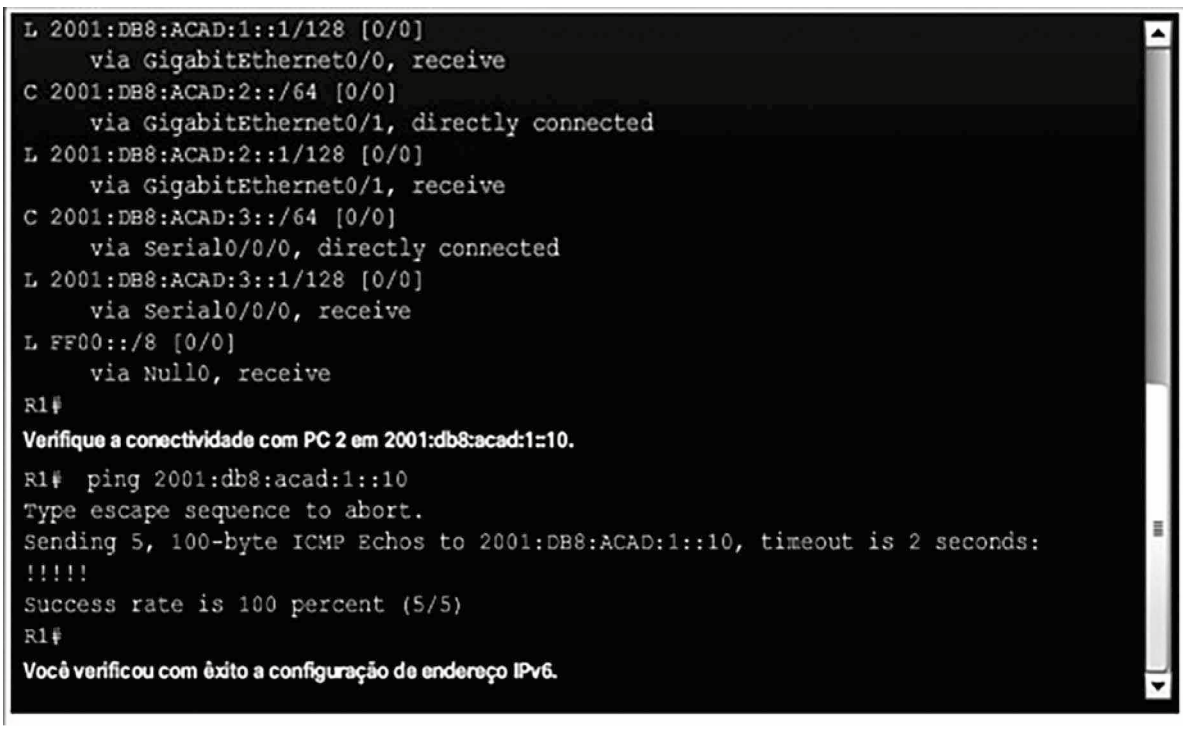


Figura 43 – Usando o verificador de sintaxe na interface IPv6, tela 3

4.6.5 Anycast

Endereça um conjunto de interfaces de múltiplos dispositivos, mas um pacote endereçado a um endereço anycast só será entregue para um dos elementos deste grupo. O elemento que receberá este pacote será o elemento com menor métrica para ser alcançado (mais curta distância). Existem diversas utilidades para esse tipo de endereço, como os servidores cluster, nos quais o servidor mais próximo da origem irá atender a esta solicitação.

4.6.6 Multicast

Do mesmo modo que o endereço anycast, o multicast endereça um conjunto de interfaces. A grande diferença é que o pacote endereçado para um endereço multicast é entregue para todas as interfaces de dispositivos. As funcionalidades de multicast são análogas às funcionalidades já existentes no IPv4.

No endereçamento IPv4, um endereço IP somente com zeros tem um significado especial. Ele se refere ao próprio host e é usado quando um dispositivo não souber seu próprio endereço. No endereçamento IPv6, esse conceito foi formalizado, e o endereço somente com zeros **0:0:0:0:0:0:0:0** recebe o nome de endereço **não especificado**.

Esse tipo de endereço é usado, normalmente, no campo de origem de um datagrama, que é enviado por um dispositivo que busca ter seu endereço IP configurado. É possível aplicar a compressão de endereços a esse endereço. Como somente contém zeros, este se tornará simplesmente **::**.

Os endereços do IPv6 usam identificadores de interface para identificar as interfaces em um link. Considere como uma **porção de host** de um endereço IPv6. Esses identificadores de interface devem ser exclusivos em um link específico. Os identificadores de interface são sempre de 64 bits e derivados dinamicamente de um endereço de Camada Enlace (endereço MAC).

Podemos atribuir uma ID de endereço IPv6 estática ou dinamicamente:

- Designação estática usando uma ID de interface manual.
- Designação estática usando uma ID de interface EUI-64.
- Configuração automática sem estado.
- DHCP para IPv6 (DHCPv6).

4.6.6.1 Endereços IPv6 multicast atribuídos

Os endereços IPv6 multicast são bem parecidos com os endereços IPv4 multicast. Vamos lembrar que um endereço multicast é usado para enviar um único pacote a um ou mais destinos (grupo multicast). Os endereços IPv6 multicast têm sempre o prefixo FF00:: /8.

Observação

Endereços multicast só podem ser endereços destino, e não endereços origem.

Existem dois tipos de endereços IPv6 multicast:

- Multicast atribuído.
- Multicast solicited-node.

Multicast atribuído

Os endereços multicast designados são endereços multicast reservados para grupos predefinidos de dispositivos. Um endereço multicast atribuído será um único endereço empregado para acessar um determinado grupamento de dispositivos que fazem uso de um serviço ou um protocolo comum. Os endereços multicast atribuídos são empregados no contexto com protocolos específicos, como o DHCPv6.

Existem dois grupos IPV6 de multicast atribuído comuns:

- Grupo multicast all-nodes (todos os nós) que começam com FF02::1. Será um grupo multicast no qual participam todos os dispositivos atribuídos para endereçamento IPv6. Qualquer pacote enviado para esse grupo é recebido e processado por todas as interfaces IPv6 no link ou na rede. A ação tem o mesmo efeito que um endereço de broadcast em IPv4. A figura a seguir demonstra esse exemplo de comunicação empregando o endereço multicast all-nodes. Um roteador IPv6 envia mensagens ICMPv6 (Internet Control Message Protocol versão 6) de RA para o grupamento multicast all-nodes. A mensagem de RA leva informações de endereçamento (como prefixo, comprimento do prefixo e gateway padrão) a todos os dispositivos habilitados para IPv6 na rede.

- Grupo multicast all-routers (todos os roteadores) que começam com FF02::2. É um grupo multicast do qual participam todos os roteadores IPv6. Quando um roteador se torna membro desse grupamento e ainda quando é ativado como roteador IPv6 com o comando de configuração global IPv6 unicast-routing, um pacote enviado para esse grupo será recebido e processado por todos os roteadores IPv6 no link ou rede.

Dispositivos atribuídos para endereçamento IPv6 enviam mensagens ICMPv6 de solicitação de roteador (RS) para o endereço multicast all-routers. Uma mensagem de RS solicita uma mensagem de RA do roteador endereçado em IPv6 para ajudar o dispositivo em sua configuração de endereço.

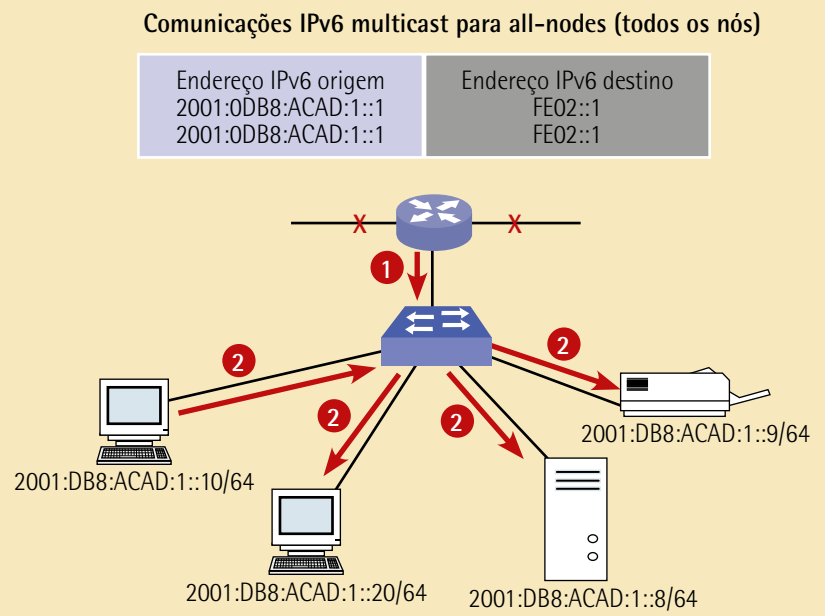


Figura 44 – Exemplo de endereçamento IPv6 multicast all-nodes

Endereços IPv6 multicast solicited-node

Todo endereço multicast solicited-node é bem parecido com o endereço multicast all-nodes. A principal vantagem do endereço multicast solicited-node é que ele é mapeado para um endereço multicast Ethernet especial. Esta condição permite que a placa de rede Ethernet filtre o quadro, examinando o endereço MAC de destino sem enviá-lo ao processo IPv6 para ver se o dispositivo é o alvo pretendido do pacote IPv6.

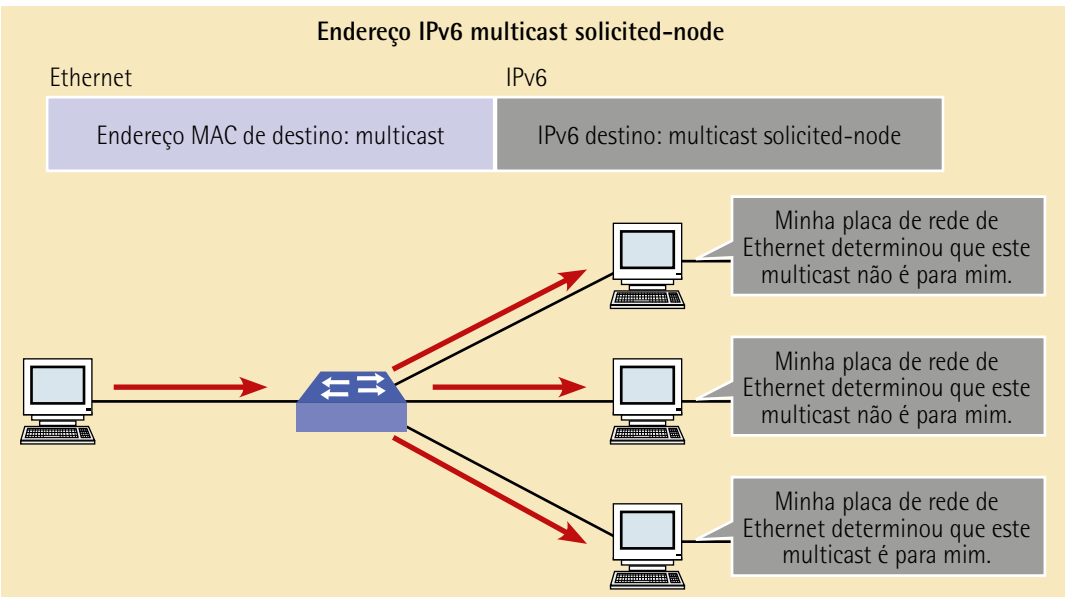


Figura 45 – Exemplo de endereço solicited-node em endereçamento IPv6

4.7 Usando as sub–redes

Já é sabido que os endereços IPv4 são compostos por grupamentos de 32 bits separados em conjuntos de 8 bits, resultando em 4 octetos representados por notação decimal e separados por pontos. Ainda vimos que o endereçamento IPv4 possui uma porção dedicada à rede e uma outra porção dedicada ao host. A porção de rede nos permite endereçar redes e ainda são compostos de um conjunto de computadores que pertence ao mesmo grupamento. Esse grupamento de hosts é endereçado pela porção de host, esta, por sua vez, pertence a uma das redes endereçadas pela porção da rede. Quem atribui qual porção pertence à rede e qual porção pertence ao host é a máscara de cálculo usada e na qual o número de redes e hosts é baseado simplesmente no tamanho da porção de rede e da porção de hosts. Para representar esse tamanho a ser obtido empregam-se os números de bits que são usados para representar cada uma das porções.

O tamanho da porção de rede também é conhecido pelo prefixo da rede, que é o número de bits que define tal porção de rede do endereço de IP. Estas também são divididas em classes A, B, C, D e E, em uma divisão em classes conhecida por endereçamento de IP classfull, ou classes cheias. As classes A, B e C são normalmente usadas para endereçar redes e hosts. A tabela a seguir apresenta essas classes.

Tabela 2 – Classes e máscaras com seus formatos decimal e binário

Endereço de classe A (decimal)	10.0.0.0
Endereço de classe A (binário)	00001010.00000000.00000000.00000000
Máscara de classe A (binário)	11111111.00000000.00000000.00000000
Máscara de classe A (decimal)	255.0.0.0
Comprimento padrão	/8
Endereço de classe B (decimal)	172.16.0.0
Endereço de classe B (binário)	10010001.00001000.00000000.00000000
Máscara de classe B (binário)	11111111.11111111.00000000.00000000
Máscara de classe B (decimal)	255.255.0.0
Comprimento padrão	/16
Endereço de classe C (decimal)	192.168.100.0
Endereço de classe C (binário)	11000000.10101000.00101010.00000000
Máscara de classe C (binário)	11111111.11111111.11111111.00000000
Máscara de classe C (decimal)	255.255.255.0
Comprimento padrão	/24

Cada classe é responsável por prover um determinado número de redes e de hosts. A classe A, por exemplo, fornece mais hosts e a classe C fornece mais redes. Você pode observar essa relação na tabela a seguir. Entretanto, esse formato de uso baseado em classes gera um significante desperdício de endereços de rede.

Tabela 3 – Número de hosts por rede

Classes	Número de redes	Número de hosts
A	128	16.777.214
B	16.384	65.534
C	2.097.152	254

Ao atribuir um endereço de classe A para uma empresa, ela receberá uma rede com 16.777.214 hosts. Nem mesmo grandes empresas possuem hosts suficientes para ocupar todo o espaço de endereçamento de uma rede classe A. Já no caso de uma rede classe B são 65.534 hosts e, embora seja um número bem menor, ainda é bastante grande, porque alocar uma classe B para uma rede de 500 hosts deixaria 65.034 endereços sem uso. Já uma classe C ofereceria somente 254 hosts, valor muito baixo para a grande maioria de empresas do mercado. Inevitavelmente, as empresas acabam aumentando de tamanho e precisando de mais endereços de rede de classe C. Vemos que muitas empresas possuem endereços classe A, como Apple, Xerox, HP e IBM.

Esquemas de endereçamento usados pelo IP versão 4 logo se mostraram limitantes, principalmente enquanto a internet crescia de forma vertiginosa. Para contornar o problema da má distribuição e também de uma futura previsão do crescimento de endereçamento IP versão 4, criou-se um novo mecanismo de divisão, as sub-redes, que consiste em repartir as redes de classe A, classe B e até as classe C em redes menores. Atribuindo essa divisão em sub-redes, temos uma ocupação de uso mais eficiente dos endereçamentos IP versão 4 através da alocação mais precisa do número de redes e, consequentemente, de hosts necessários para cada empresa. Também ajuda na divisão da rede de uma empresa em redes menores, como por departamento ou por política de acesso e recursos de rede. Fazer a divisão de redes menores ou sub-redes ajuda também na redução dos domínios broadcast e em um melhor gerenciamento do conjunto da rede.

A forma de dividir redes em sub-redes é muito simples. Primeiro devemos escolher qual endereço IP será dividido e em quantas redes iremos dividi-lo. Depois podemos fazer a divisão a partir do número de hosts que desejamos para cada rede, sem ter grande atenção diretamente com o número de redes.

Como já estudamos anteriormente, quem define a porção de rede e a porção de hosts é a máscara de rede. É ela que nos permite identificar quantos bits temos em cada porção. Para executar essa divisão vamos obter os bits da porção de host e transferi-los para a porção de rede, fazendo essa transferência estamos criando uma porção sub-rede que fica intermediária entre a porção de rede e a porção de hosts, como observado na figura a seguir.

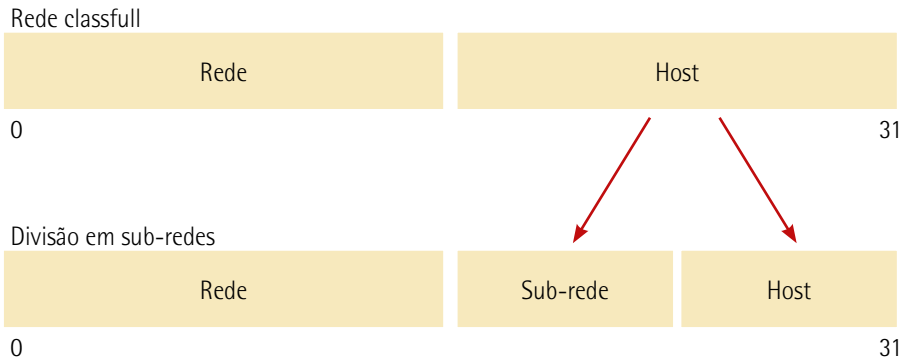


Figura 46 – Porções do endereço de rede

Agora podemos dividir o endereço 172.16.0.0 em quatro sub-redes. Vemos que este endereço é de classe B, então temos que **roubar** bits emprestados da porção de host. No endereço de classe B, a porção de rede corresponde aos primeiros 16 bits e a de host, aos 16 bits seguintes. Vamos pegar emprestados os bits mais significativos ou, ainda, o mais à esquerda da porção de host para atribuir quatro sub-redes. Depois vamos pegar bits suficientes para endereçá-las.

Para chegar ao número quatro usando a regra de 2^b, onde b é o número de bits que pegamos emprestados, precisaremos de 2 bits. Esses bits que foram retirados da porção de host vão fazer parte da porção de sub-rede e também serão contabilizados pela máscara de sub-rede. A máscara de sub-rede nos indica o que é porção de rede e o que é porção de host. A seguir, vamos ver o processo finalizado para atribuir os quatro novos endereços IP.

Dividindo o endereço 172.16.0.0, 255.255.0.0 ou 172.16.0.0/16 em quatro sub-redes

Precisaremos de dois bits da porção de host, pois $2^2 = 4$. A máscara nos mostra que a porção de rede é composta dos dois primeiros octetos e a porção de host, dos dois octetos restantes. A figura a seguir apresenta a porção de rede e host em relação à máscara padrão e a escolha dos dois bits mais significativos que serão **roubados** para criar as sub-redes.

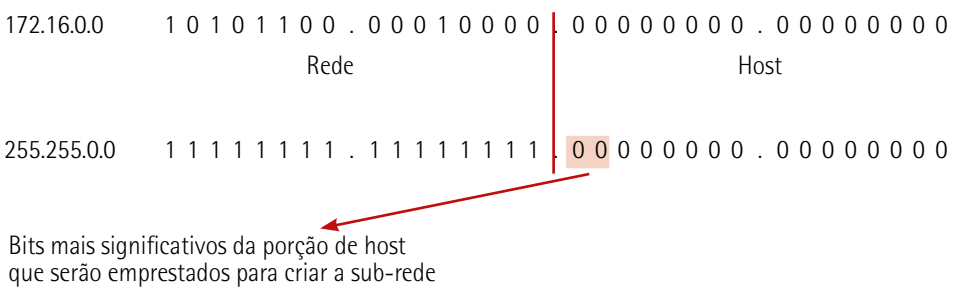


Figura 47 – Identificação da porção de rede

No momento em que os bits necessários para obter as quatro sub-redes são selecionados, passam a compor a proporção de sub-rede e ainda ocorre uma alteração significativa da máscara, que passa a ter uma nova denominação: máscara de sub-rede. Essa nova máscara indicará uma porção de

rede estendida, pois complementa os bits que foram emprestados para criar as sub-redes. A figura a seguir nos mostra a nova máscara, os bits emprestados para a porção sub-rede e as três porções que compõem o endereçamento IP.

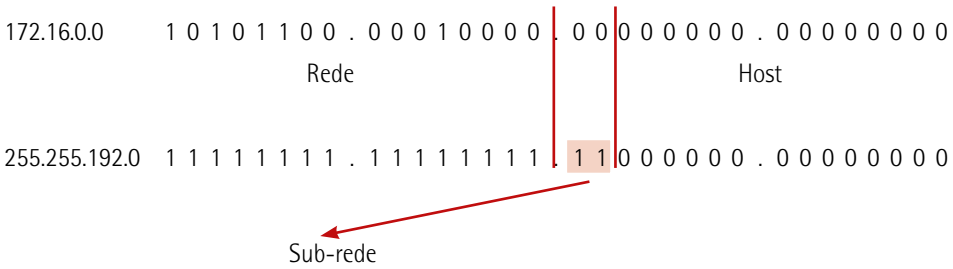


Figura 48 – Bits emprestados para a sub-rede

Calculada a nova máscara, ela passa a criar as sub-redes. Já sabemos que o endereço de rede possui todos os bits da porção host que foram definidos como o número 0 e que o endereço broadcast possui todos os bits da porção de hosts que foram definidos como número 1. Já podemos identificar o primeiro endereço de sub-rede, que será o próprio endereço da rede usado para realizar essa divisão, mas com uma nova máscara. Para identificar o endereço de broadcast colocamos todos os bits da porção de hosts com os números 1 em binário.

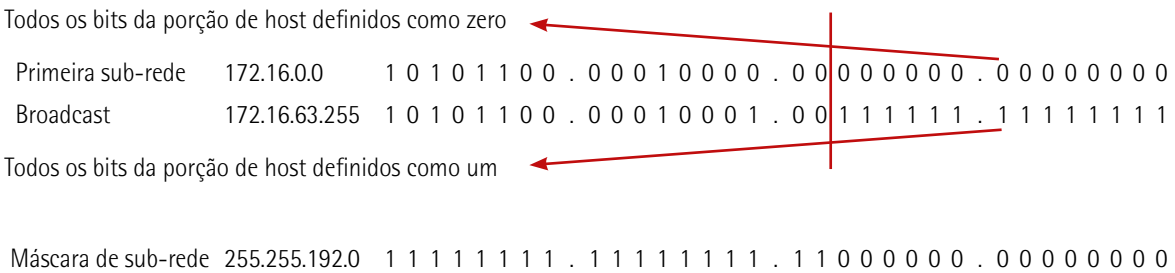


Figura 49 – Demonstração da primeira sub-rede

Os outros três endereços remanescentes de sub-rede serão obtidos por meio da manipulação dos dois bits da porção sub-rede. Então, realizando todas as combinações possíveis para obter os quatro endereços de sub-rede, as próximas sub-redes serão demonstradas na figura a seguir.

Lembrete

Vale lembrar que sempre que manipularmos os bits da porção de sub-rede, devemos fazer a conversão em decimal do octeto completo e não somente dos dois bits manipulados.

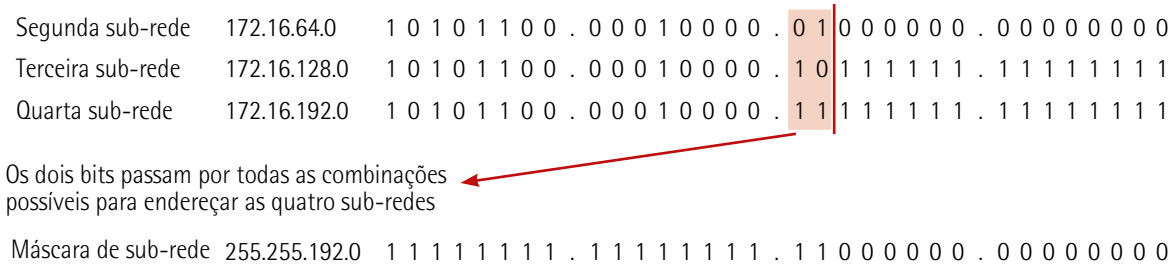


Figura 50 – Combinando os bits das sub-redes

Conforme o exemplo, foi necessário manipular apenas os 2 bits para obter todas as combinações possíveis, entretanto, quando se manipula mais bits, essa tarefa será bem difícil e complexa. A seguir, vamos descobrir como obter mais redes sem a necessidade de manipular os bits, a fim de obter todas as combinações necessárias.

4.7.1 Resolvendo o cálculo de sub-redes

Já vimos que os endereços IP são compostos de uma porção de rede e outra porção de host, ainda, já estudamos as diferentes classes de serviço, que, claro, possuem diferentes tamanhos para porção de rede e para porção de host. Para escolher qual parte do endereço de 32 bits representa a porção de rede é preciso usar a máscara. Ela é um número de 32 bits, assim como o próprio endereço IP versão 4, só que este possui um conjunto de bits com números 1 indicando o grau de relevância que essa máscara possui. A leitura é sempre feita da esquerda para a direita, indicando quais bits do endereço são significativos e quais são de interesse para o uso do roteamento. Bits significativos apresentam exatamente a porção de rede, como já vimos.

A primeira fase para executar um cálculo de sub-rede é definir qual é o endereço que será aplicado para uma implementação. Esse endereço pode ser obtido de diversas formas, por meio de um provedor de comunicação, por exemplo, ou ainda podemos optar por utilizar endereços de IPs privados.

Uma vez sabendo esse endereço, é preciso determinar o número de redes e o número de hosts que desejamos. O número de redes ou número de hosts irão nos indicar quantos bits vamos precisar para a porção de hosts e criar as devidas sub-redes desejadas.

Podemos fazer uma escolha por um determinado número de sub-redes e, ainda assim, pegar os bits emprestados para uma porção de hosts. Desse jeito, esses serão suficientes para endereçar o número de redes que precisamos. Se queremos, por exemplo, 28 sub-redes, vamos pegar 5 bits. Apenas 5 bits nos dão a percepção de obter 32 novas sub-redes, ou seja, 4 sub-redes a mais do que precisamos. Porém, se usássemos apenas 4 bits, teríamos somente 16 sub-redes, um número bem menor do que precisamos. Devemos então considerar que, ao usar 5 bits, teremos 4 redes disponíveis para um crescimento futuro. Essa é uma boa técnica para prever futuros crescimentos na infraestrutura. Optando por redes com o número de mínimo de hosts, precisamos observar atentamente quantos bits serão necessários para ter uma porção de hosts. Com o número de hosts a serem escolhidos, os bits que sobraram da porção de hosts serão os que emprestaremos para criar essas sub-redes.

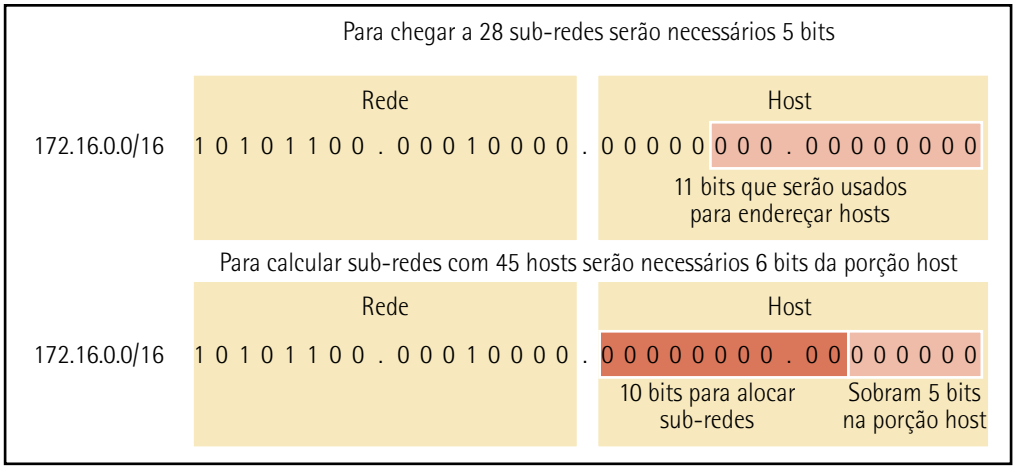


Figura 51 – Exemplificando a alocação de bits para hosts e sub-redes

A figura mostra que, quando feita uma opção por 28 sub-redes, teremos na verdade 32 sub-redes, embora cada uma das sub-redes terá 2.046 hosts. Quando feita a opção por 45 hosts, teremos exatamente 62 hosts no lugar dos 10 bits restantes, que serão reservados para sub-rede, dando um total de 1.024 sub-redes, cada uma delas com 62 hosts.

Escolhido o número de bits que vão ser emprestados da porção de hosts, incluiremos esses bits na máscara com a função de determinar as sub-redes e os endereços de host. Finalmente, podemos atribuir os endereços aos hosts de rede. No exemplo usado, as máscaras seriam o seguinte:

- Para a opção pelo número de redes: 255.255.248.0.
- Para a opção pelo número de hosts: 255.255.255.192.

Sempre que manipulamos os bits de um endereço para criar as sub-redes, devemos ficar atentos à classe a que pertence aquele endereço.

Se for um endereço de classe A, a porção de redes possui 8 bits e a porção de hosts, logicamente, 24 bits. Podemos ainda pegar os bits emprestados da porção de hosts a partir do nono bit do endereço. No caso de endereço de classe B, a porção de redes possui apenas 16 bits, assim como a de hosts.

A figura seguinte nos dá um exemplo de um endereço de classe C com máscara padrão classfull. Depois de pegar emprestada a porção de host, devemos pegar o décimo bit do endereço – no caso da classe C, que possui 24 bits da porção host, podemos ainda pegar o vigésimo quinto bit em diante –, os bits devem ser sempre adquiridos do mais significativo para o menos significativo. Ou seja, a leitura deve ser feita da esquerda para a direita, em sequência, sem faltar nenhum bit. Os bits disponíveis que podem ser emprestados em cada classe são mostrados na figura a seguir.

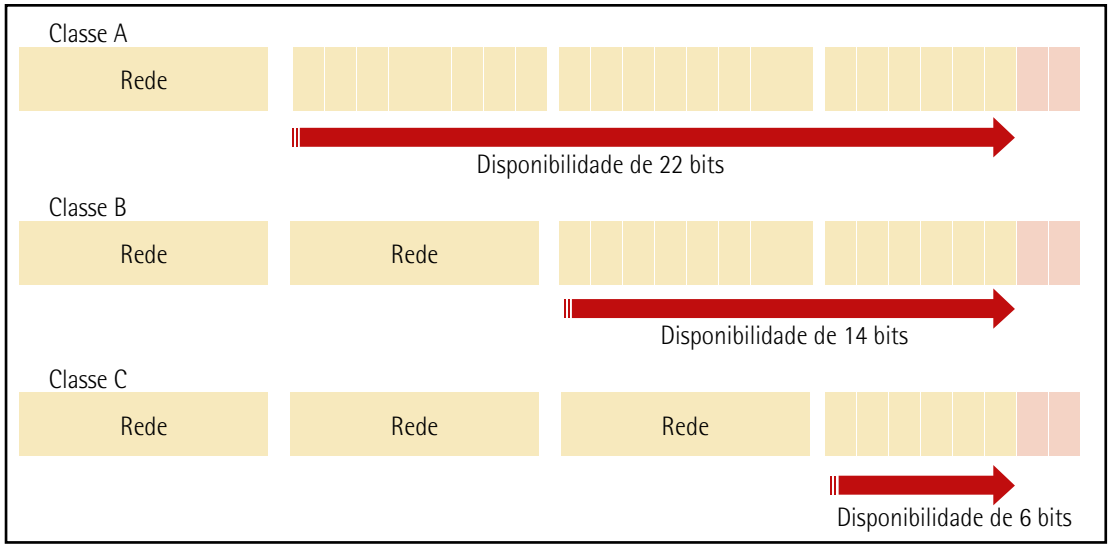


Figura 52 – Disponibilidade de bits para cada classe de rede

Depois de pegar os bits emprestados, precisamos ter muita atenção para deixar pelo menos 2 bits para porção de host. Isso é necessário para ter sempre 2 hosts válidos em cada rede. Dois bits nos permitem obter 2 hosts, pois $2^2 - 2 = 2$, o que permite, nessa condição, números de host válidos para um endereço de rede que possui 2 bits disponíveis na sua porção de host, que é o caso da máscara /30.

A seguir vamos ver alguns exemplos da criação de sub-redes. Esses exemplos vão proporcionar cálculos usando como base o número de redes e o número de hosts desejados, e os endereços usados serão das classes A, B e C. Para os exemplos 1 e 2, vamos usar o endereço de classe A 10.0.0.0/8; para os exemplos 3 e 4, o endereço de classe B 172.16.0.0/16; e, para os exemplos 5 e 6, o endereço de classe C 192.168.1.0/24.

Exemplo 1 – Classe A: 10.0.0.0/8

Precisamos dividir o endereço em 400 sub-redes. Usaremos o endereço de classe A 10.0.0.0, que tem como máscara padrão 255.0.0.0. Sabendo o número de sub-redes, temos que verificar quantos bits são necessários para termos o número 400, ou maior, utilizando a regra de 2^b, onde b é o número de bits necessários.

No caso de 400 sub-redes, precisaremos de 9 bits, pois 2^9 é igual a 512. Caso usemos 8 bits, teríamos somente 256 sub-redes, número insuficiente para a nossa necessidade.

Fazendo o cálculo de 2^b descobrimos que devemos pegar 9 bits emprestados da porção de host para que sejam utilizados na porção de sub-rede.

Pegamos emprestados os 9 bits mais significativos da porção de host em destaque na figura a seguir. Identificamos que a porção de host ficou com 15 bits. Esses bits serão utilizados para endereçar os hosts, totalizando 32.766 hosts por sub-rede, número atingido no cálculo.

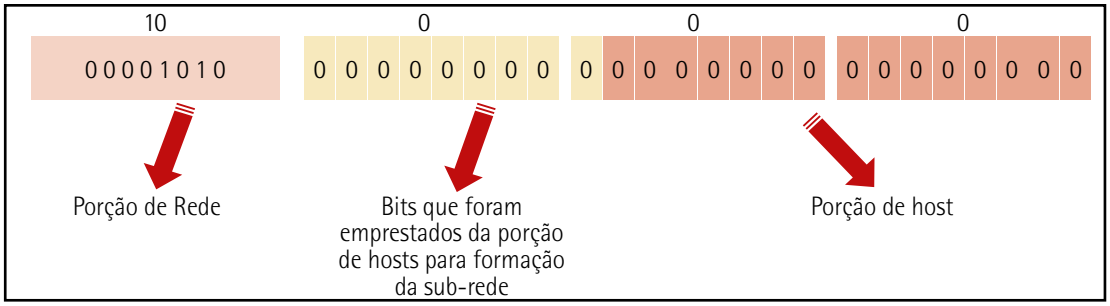


Figura 53 – Calculando 400 redes IPv4 para classe A

Para saber o primeiro endereço de rede e seu endereço de broadcast, é preciso definir todos os bits da porção de rede com 0 e 1, na ordem. A figura apresenta a porção de host com todos os bits em zero e a figura seguinte, com todos os bits de host definidos em um (broadcast).

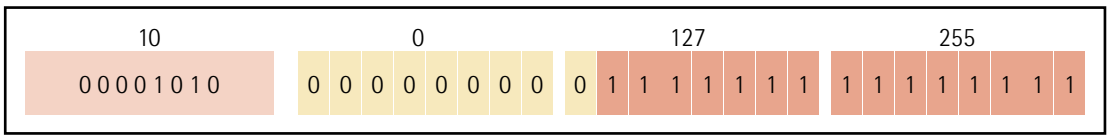


Figura 54 – Endereço broadcast da primeira rede classe A

Nesse exemplo, identificamos que o primeiro endereço de host válido para a rede 10.0.0.0/17 é obtido marcando todos os bits de host como zero, menos o último, ou seja, o menos significativo.

Desse jeito teremos o endereço 10.0.0.1 como primeiro endereço válido. Para obter o último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast e teremos o endereço 10.0.127.254 como o último endereço válido, ou ainda usar a fórmula conhecida $2^n - 2$ = hosts, onde **n** é a quantidade de zeros mais à direita na máscara resultante.

Ainda, temos que alterar a máscara de 255.0.0.0 para a máscara de sub-rede. Precisamos disso para definir como binários o número de bits referentes à porção de sub-rede, neste caso 9 bits. A figura a seguir mostra a máscara padrão e a máscara de sub-rede calculada.

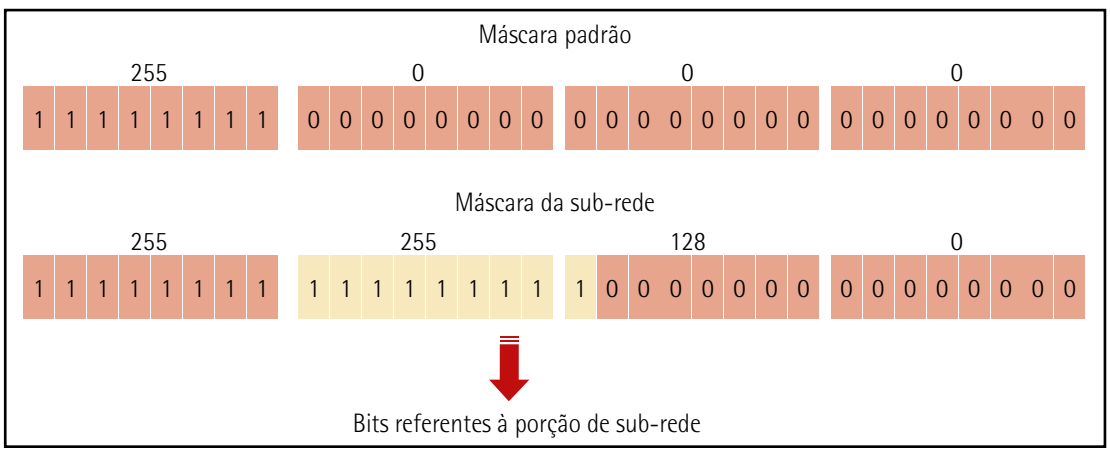


Figura 55 – Representação da máscara de sub-rede classe A

O resto dos endereços serão resultantes do cálculo, por meio da manipulação dos 9 bits emprestados para a porção de sub-rede. Precisamos fazer todas as combinações possíveis de 0 e 1 para chegar a todas as sub-redes, no entanto, realizar essa operação para muitos bits é cansativo.

Para obter o próximo endereço de rede, basta adicionar uma unidade ao último octeto do endereço de broadcast. Entretanto, ao fazer essa soma, chegaremos ao número 256. Como o valor de cada octeto deve estar entre 0 e 255, em vez de colocar 256, colocamos zero e adicionamos uma unidade saltando ao terceiro octeto. Teremos então o número 128 no terceiro octeto. O endereço calculado depois das adições será 10.0.128.0, o segundo endereço de rede da divisão. A figura seguinte mostra o endereço do cálculo de rede e do cálculo de broadcast em binários. O primeiro endereço válido da segunda rede será obtido da mesma forma que na primeira rede, definindo o bit menos significativo da porção de host como um. Teremos o endereço 10.0.128.1 como primeiro endereço válido para a segunda rede. No caso do endereço do último endereço válido, diminuímos uma unidade do último octeto do endereço de broadcast, ou seja, teremos o endereço 10.0.255.254, e assim sucessivamente.

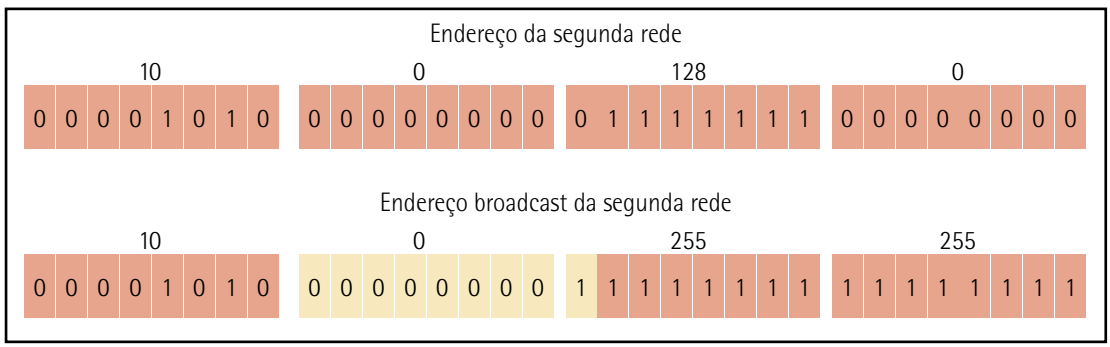


Figura 56 – Endereço da segunda rede e da broadcast classe A

Para chegar ao terceiro endereço de rede, o procedimento é o mesmo realizado para obter a segunda rede. Ao somar uma unidade ao quarto octeto, teremos o valor 256, ou seja, mudamos o octeto para zero, saltando para o próximo octeto, e adicionamos um ao terceiro octeto. Entretanto, o terceiro octeto também nos dará o valor 256 ao ser adicionado em um. Devemos alterar o terceiro octeto para zero e adicionar uma unidade ao segundo octeto. Teremos o valor um no segundo octeto e obteremos o terceiro endereço de rede, 10.1.0.0.

A tabela a seguir mostra os primeiros e últimos endereços de rede e seus respectivos endereços de broadcast para a divisão em sub-redes do endereço aplicado a este primeiro exemplo.

Tabela 4– Endereços de sub-rede e broadcast do exemplo 1

	Endereço de rede	Endereço de broadcast
1º endereço	10.0.0.0	10.0.127.255
2º endereço	10.0.128.0	10.0.255.255
3º endereço	10.1.0.0	10.1.127.255
510º endereço	10.254.128.0	10.254.255.255
511º endereço	10.255.0.0	10.255.127.255
512º endereço	10.255.128.0	10.255.255.255

Exemplo 2 – Classe A: 10.0.0.0/8

Tendo o objetivo de dividir o endereço na forma de obter, pelos menos, 400 hosts por sub-rede, usaremos o endereço de classe A 10.0.0.0/8, com sua máscara padrão 255.0.0.0. Sabendo o número de hosts desejados, vamos descobrir quantos bits são necessários para chegarmos ao número 400. Assim, usaremos a regra 2^b-2, onde b é o número de bits necessários para endereçar os hosts. No caso de 400 hosts, vamos precisar de 9 bits, pois 2^9-2 é igual a 510.

Fazendo o cálculo de 2^b-2, descobrimos que vamos precisar usar 9 bits na porção de host para endereçar os 400 hosts. Vemos que, desta vez, os bits do cálculo não são os mesmos bits que devemos pegar emprestados, mas sim os bits que usamos para endereçar os hosts. Esses 9 bits serão os bits da nova porção de host. Para chegar ao número de bits da porção de sub-rede, devemos usar os bits da porção de host original e subtrair os bits que necessitamos, assim, 9 bits. Ao subtrair 9 de 24, obtemos 15. A porção de sub-rede terá 15 bits, que equivalem aos 15 bits mais significativos da porção de host original. Com esses 15 bits da porção de rede, chegaremos a ter até 32.768 redes, cada uma com até 510 hosts.

A figura a seguir mostra as porções originais e as porções encontradas após o cálculo.

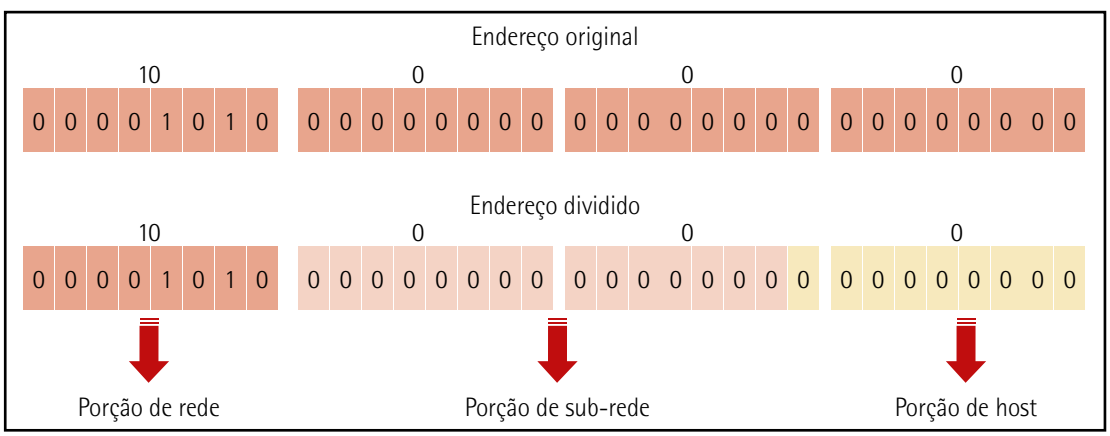


Figura 57 – Porção de sub-redes (exemplo 2; classe A)

Observação

Preste atenção que, ao fazer a opção por definir o número de hosts como base para realizar o cálculo de sub-rede, os bits que serão usados para a porção de sub-rede são aqueles que não precisamos mais para obter o número de hosts desejados. Já definidos quantos bits restaram para a porção de sub-rede, a obtenção dos endereços IP ocorre do mesmo jeito que foi apresentada no exemplo anterior. Neste exemplo, a máscara de sub-rede terá 15 bits definidos como um, além dos 8 bits originais. Sabemos que a máscara de sub-rede será, em decimal, 255.255.254.0 ou, em binário, 11111111.11111111.11111110.00000000.

Para chegarmos ao primeiro endereço de rede e seu endereço de broadcast, necessitamos definir todos os bits da porção de rede com 0 e 1, sucessivamente.

A figura do exemplo apresenta a porção de host com todos os bits em zero, e a figura seguinte, com todos os bits de host definidos em um (broadcast).

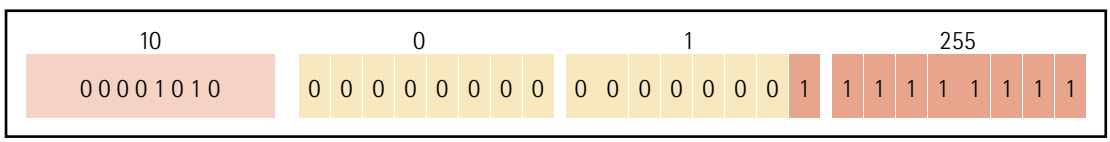


Figura 58 – Endereço broadcast da primeira rede do exemplo 2 (classe A)

O primeiro endereço de host encontrado depois dos cálculos feitos para a rede 10.0.0.0/23 é obtido definindo todos os bits de host como zero, exceto o último, ou seja, o menos significativo.

Deste jeito, chegaremos ao endereço 10.0.0.1 como primeiro endereço válido. Para obter o último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast. Teremos o endereço 10.0.1.254 como o último endereço válido.

Os demais endereços chegarão pelo cálculo e pela manipulação dos 15 bits emprestados para a porção de sub-rede. Precisamos realizar todas as combinações possíveis de zeros e uns para obter o resultado de todas as sub-redes.

Para chegar ao próximo endereço de rede, vamos acrescer uma unidade ao último octeto do endereço de broadcast. Entretanto, ao fazer esta soma, chegaremos ao número 256. Devemos colocar zero neste octeto e adicionar uma unidade, fazendo um salto ao terceiro octeto. Chegaremos, então, ao número um no terceiro octeto. O endereço obtido proveniente da sequencias de operações de somatória será 10.0.2.0, ou seja, o segundo endereço de rede da divisão.

A tabela a seguir mostra os endereços de rede e broadcast para as primeiras e últimas sub-redes.

Tabela 5 – Endereços de sub-rede e broadcast do exemplo 2 (classe A)

	Endereço de rede	Endereço de broadcast
1º endereço	10.0.0.0	10.0.1.255
2º endereço	10.0.2.0	10.0.3.255
3º endereço	10.0.4.0	10.0.5.255
32766º endereço	10.255.250.0	10.254.251.255
32767º endereço	10.255.252.0	10.255.253.255
32768º endereço	10.255.254.0	10.255.255.255

Exemplo 3 – Classe B: 172.16.0.0/16

Precisamos dividir o endereço em 12 sub-redes. Usaremos para esta tarefa o endereço de classe B 172.16.0.0, que tem como máscara padrão 255.255.0.0. Sabendo o número de sub-redes, temos que verificar quantos bits são aguardados para termos o número 12 ou maior, utilizando a regra de 2^b, onde b é o número de bits necessários. No caso de 12 sub-redes, precisaremos de 4 bits, pois 2^4 é igual a 16. Se usarmos 3 bits, teremos somente 8 sub-redes, número insuficiente para nossa solução.

Realizando este cálculo de 2^b, identificamos que devemos pegar 4 bits emprestados da porção de host para que sejam utilizados na porção de sub-rede. Pegaremos emprestados os 4 bits mais significativos da porção de host, no destaque da figura a seguir.

Observamos que a porção de host ficou com 12 bits. Tais bits vão ser usados para endereçar os hosts, totalizando 4.094 hosts por sub-rede.

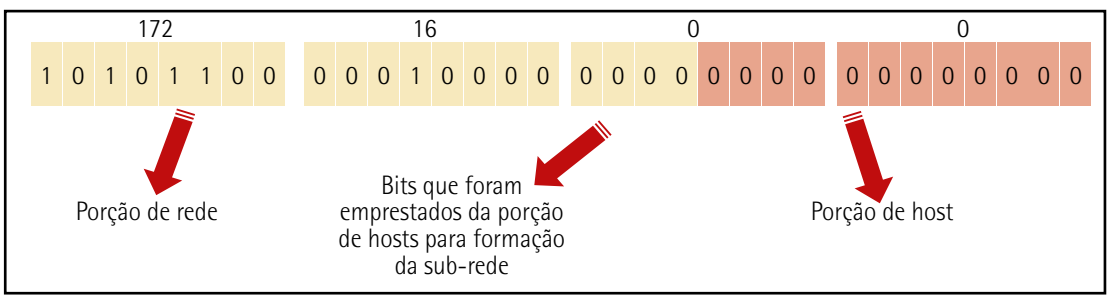


Figura 59 – Demonstração dos bits emprestados para cálculo de 12 sub-redes

Para chegar ao primeiro endereço de rede e seu endereço de broadcast, precisamos saber todos os bits da porção de rede com 0 e 1, sucessivamente. A figura anterior mostra a porção de host com todos os bits em zero, e a figura a seguinte, com todos os bits de host definidos em um (broadcast).

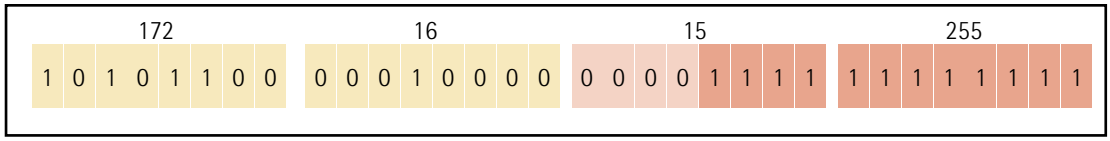


Figura 60 – Endereço broadcast da primeira rede do exemplo 3 (classe B)

Calculando o endereço de host válido para a rede 172.16.0.0/20, precisamos definir todos os bits de host como zero, exceto o último, ou seja, o menos significativo. Assim, chegamos ao endereço 172.16.0.1 como primeiro endereço válido.

Para chegar ao último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast. Teremos o endereço 172.16.15.254 como o último endereço válido.

Os demais endereços serão calculados pela manipulação dos 4 bits emprestados para a porção de sub-rede. Devemos realizar todas as combinações possíveis de 0 e 1 para obter todas as sub-redes.

Para chegar ao próximo endereço de rede, basta somar uma unidade ao último octeto do endereço de broadcast. Entretanto, depois de fazer esta somatória, chegaremos ao número 256. Como o valor de cada octeto deve estar entre 0 e 255, em vez de colocar 256, colocamos zero e saltamos ao próximo octeto, adicionando uma unidade ao terceiro octeto.

Chegaremos então ao número 16 no terceiro octeto. O endereço obtido depois das operações de soma será 172.16.16.0, o segundo endereço de rede da divisão. A figura a seguir mostra o endereço de rede e de broadcast em binários.

O primeiro endereço válido da segunda rede será conseguido do mesmo jeito, definindo o bit menos significativo da porção de host como um. Chegaremos ao endereço 172.16.16.1 como primeiro endereço válido para a segunda rede. No caso do último endereço válido, diminuímos uma unidade do último octeto do endereço de broadcast, ou seja, teremos o endereço 172.16.31.254.

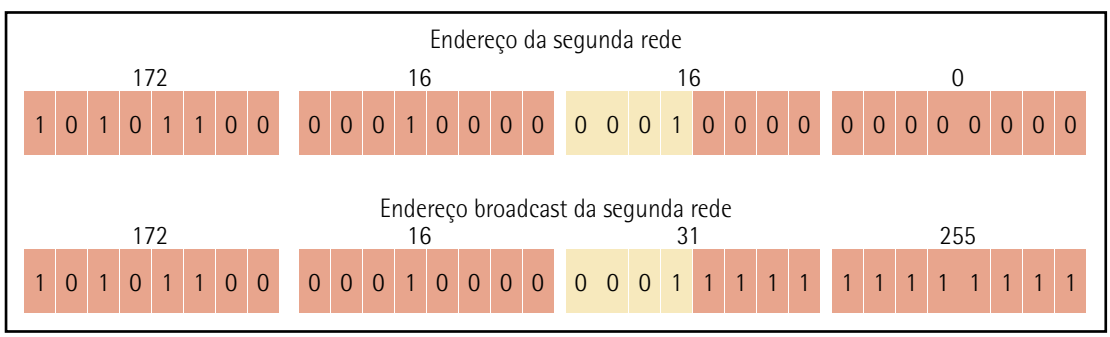


Figura 61 – Endereço e broadcast da segunda rede do exemplo 3 (classe B)

A tabela a seguir declara os 16 endereços de sub-rede e seus respectivos endereços de broadcast para a divisão em sub-redes do endereço usado no exemplo 3.

Tabela 6 – Endereços de sub-rede e broadcast do exemplo 3 (classe B)

	Endereço de rede	Endereço de broadcast
1º endereço	172.16.0.0	172.16.15.255
2º endereço	172.16.16.0	172.16.31.255
3º endereço	172.16.32.0	172.16.47.255
4º endereço	172.16.48.0	172.16.63.255
5º endereço	172.16.64.0	172.16.79.255
6º endereço	172.16.80.0	172.16.95.255
7º endereço	172.16.96.0	172.16.111.255
8º endereço	172.16.112.0	172.16.127.255
9º endereço	172.16.128.0	172.16.143.255
10º endereço	172.16.144.0	172.16.159.255
11º endereço	172.16.160.0	172.16.175.255
12º endereço	172.16.176.0	172.16.191.255
13º endereço	172.16.192.0	172.16.207.255
14º endereço	172.16.208.0	172.16.223.255
15º endereço	172.16.224.0	172.16.239.255
16º endereço	172.16.240.0	172.16.255.255

Exemplo 4 – Classe B: 172.16.0.0/16

Precisamos dividir o endereço para chegar ao cálculo de pelo menos 200 hosts por sub-rede. Usaremos o endereço de classe B 172.16.0.0, que tem como máscara padrão 255.255.0.0. Sabendo o número de hosts desejado, vamos calcular quantos bits são necessários para termos o número 200. Assim, usamos a regra 2^b-2, onde b é o número de bits necessários para endereçar os hosts. Neste exemplo, para alcançarmos 200 hosts, precisaremos de 8 bits, pois 2^8 – 2 é igual a 254.

Fazendo os cálculos, 2^b – 2, descobrimos que usaremos 8 bits na porção de host para endereçar os 200 hosts. Atenção: os bits do cálculo não se referem aos bits que devemos pegar emprestados, mas sim aos bits utilizados para endereçar os hosts. Esses 8 bits serão os bits da nova porção de host. Para chegar ao número de bits da porção de sub-rede, vamos pegar os bits da porção de host original e subtrair os bits de que necessitamos, ou seja, 8 bits. Ao subtrair 8 de 16, chegaremos a 8. A porção de sub-rede terá 8 bits, que equivalem aos 8 bits mais significativos da porção de host original. Com os 8 bits da porção de rede, teremos até 256 redes, cada uma com até 254 hosts. A figura a seguir nos mostra as porções originais e as obtidas depois de calculadas.

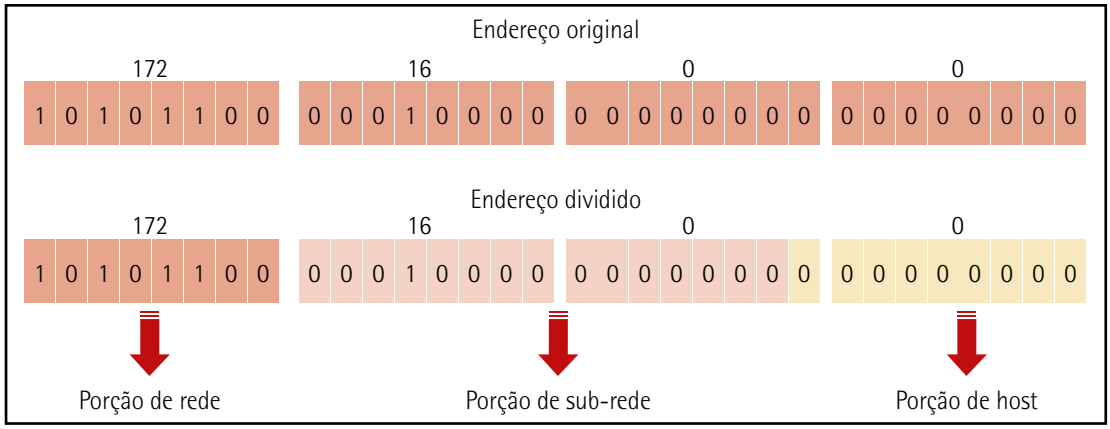


Figura 62 – Porção de sub-redes depois da divisão do exemplo 4 (classe B)

Perceba que, em nossa opção por dividir o número de hosts como a base para fazer o cálculo de sub-rede, os bits que serão emprestados para a porção de sub-rede serão aqueles que não precisamos para obter o número de hosts desejados. Feito o cálculo e definido quantos bits restaram para a porção de sub-rede, chegaremos aos endereços IP da mesma forma que foi apresentada no exemplo 2.

Aqui, a máscara de sub-rede terá 8 bits definidos como um, além dos 16 originais. Calculamos como máscara de sub-rede, em decimal, 255.255.255.0 ou, em binário, 11111111.11111111.11111111.00000000.

Para chegar ao primeiro endereço de rede e seu endereço de broadcast, precisamos atribuir todos os bits da porção de rede com 0 e 1, respectivamente. A figura anterior mostra a porção de host com todos os bits em zero, e a figura a seguir, com todos os bits de host definidos em um (broadcast).

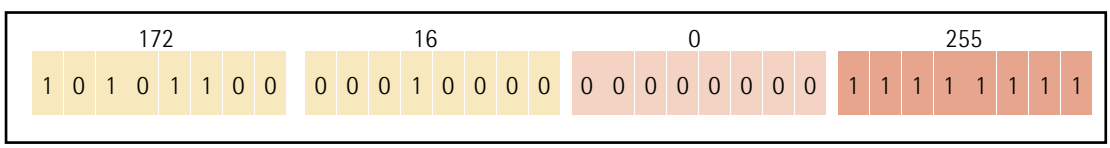


Figura 63 – Endereço broadcast da primeira rede do exemplo 4 (classe B)

O primeiro endereço de host válido para a rede 172.16.0.0/24 é alcançado mudando todos os bits de host como zero, exceto o último, ou seja, o menos significativo.

Assim, chegamos ao endereço 172.16.0.1 como primeiro endereço válido. Para chegar ao último endereço, subtraímos uma unidade do valor do último octeto do endereço de broadcast. Teremos o endereço 172.16.0.254 como o último endereço válido.

Executando os demais cálculos dos endereços, estes serão obtidos por meio da manipulação dos 15 bits emprestados para a porção de sub-rede. Vamos proceder todas as combinações possíveis de 0 e 1 para chegar a todas as sub-redes.

Para chegar ao próximo endereço de rede, calculamos a soma de uma unidade ao último octeto do endereço de broadcast. Entretanto, para executar essa adição, chegaremos ao número 256. Devemos

colocar zero neste octeto e proceder um salto ao próximo octeto e adicionar uma unidade ao terceiro octeto. Assim chegamos ao número um no terceiro octeto. O endereço obtido depois de todas as somatórias será 172.16.1.0, o segundo endereço de rede da divisão.

A tabela a seguir nos mostra os primeiros e últimos endereços de sub-rede para a divisão do endereço 172.16.0.0/16.

Tabela 7 – Endereços de sub-rede e broadcast do exemplo 4

	Endereço de rede	Endereço de broadcast
1º endereço	172.16.0.0	172.16.0.255
2º endereço	172.16.1.0	172.16.1.255
3º endereço	172.16.2.0	172.16.2.255
510º endereço	172.16.253.0	172.16.254.255
511º endereço	172.16.254.0	172.16.254.255
512º endereço	172.16.255.0	172.16.255.255

Exemplo 5 – Classe C: 192.168.1.0/24

Precisamos dividir o endereço em três sub-redes. Usaremos o endereço de classe C 192.168.1.0, que tem como máscara padrão 255.255.255.0. Sabendo o número de sub-redes, precisaremos saber quantos bits são necessários para chegarmos ao número três ou maior utilizando a regra de 2^b, onde b é o número de bits necessários para o cálculo.

Para chegar a três sub-redes, precisamos apenas de 2 bits, pois 2^2 é igual a 4. Ao fazer este cálculo de 2^b saberemos que é preciso pegar dois bits emprestados da porção de host para serem aplicados na porção de sub-rede. Vamos emprestar os 2 bits mais significativos da porção de host, veja em destaque na figura a seguir. Importante observar que a porção de host ficou com 6 bits. Esses bits serão usados para endereçar os hosts, chegando ao total de 62 hosts por sub-rede.

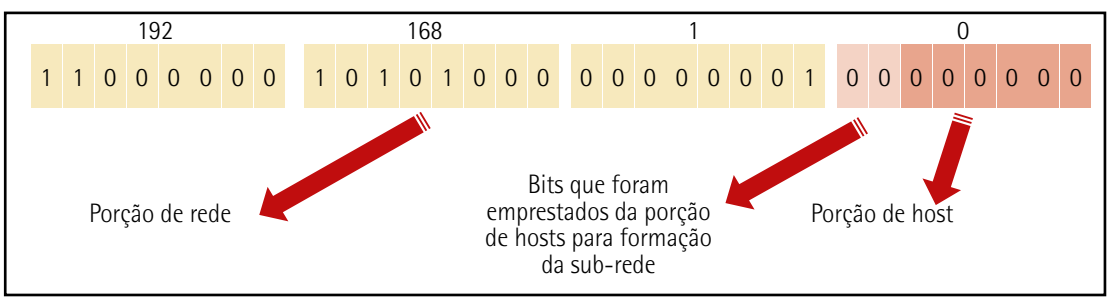


Figura 64 – Cálculo de 10 sub-redes do exemplo 5 (classe C)

Para chegar ao primeiro endereço de rede e seu endereço de broadcast, precisamos saber o valor de todos os bits da porção de rede com 0 e 1, respectivamente. A figura anterior mostra a porção de host com seus bits em zero, e a figura a seguir, com todos os bits de host definidos em um (broadcast).

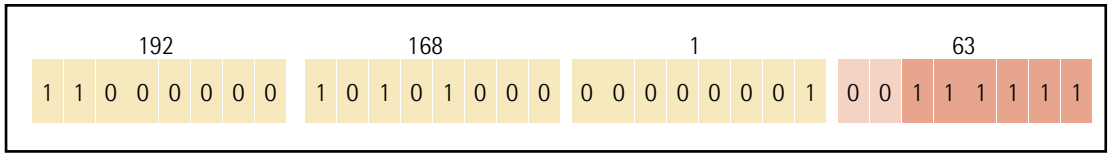


Figura 65 – Endereço broadcast da primeira rede do exemplo 5 (classe C)

Para chegar ao primeiro endereço de host válido para a rede 192.168.1.0/26, basta calcular todos os bits de host como zero, exceto o último, ou seja, o menos significativo.

Dessa forma, chegaremos ao endereço 192.168.1.1 como primeiro endereço válido para esta rede. Para saber o último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast. Chegamos ao endereço 192.168.1.62 como o último endereço válido.

Os cálculos dos endereços restantes serão alcançados pela manipulação dos 2 bits emprestados para a porção de sub-rede. Precisamos fazer todas as combinações possíveis de 0 e 1 para chegar ao valor das sub-redes. Para obter o próximo endereço de rede, precisamos apenas somar uma unidade ao último octeto do endereço de broadcast. Fazendo esta somatória chegaremos ao número 64. O valor obtido depois da adição será 192.168.1.64, o segundo endereço de rede depois da divisão. O primeiro endereço válido da segunda rede será obtido do mesmo jeito como feito na primeira rede, atribuindo o bit menos significativo da porção de host da máscara como um. O resultado é o endereço 192.168.1.65 como primeiro endereço válido para a segunda rede. Em relação ao último endereço válido, subtraímos uma unidade do último octeto do endereço de broadcast, assim, teremos o endereço 192.168.1.126. A figura a seguir nos mostra o endereço de rede e de broadcast em binários e mostra os endereços de rede e broadcast para a segunda rede.

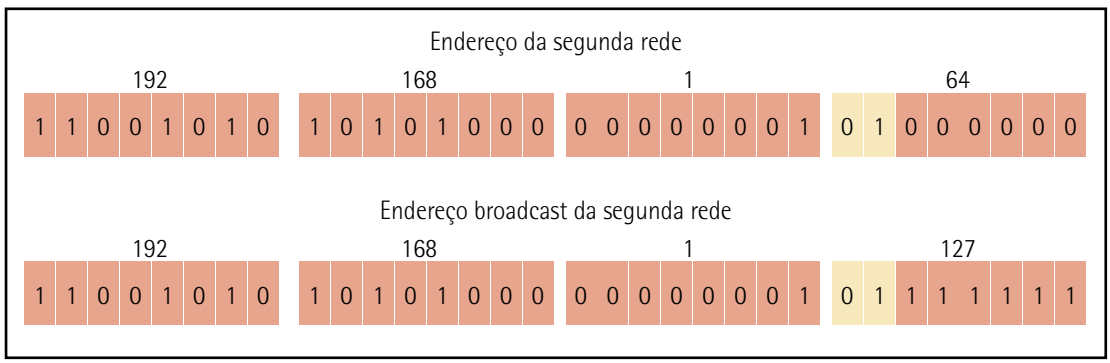


Figura 66 – Endereço e broadcast da segunda rede do exemplo 5 (classe C)

A tabela a seguir nos mostra os quatro endereços de sub-rede e seus respectivos endereços de broadcast para a divisão em sub-redes do endereço do exemplo 5.

Tabela 8 – Endereços de sub-rede e broadcast do exemplo 5 (classe C)

	Endereço de rede	Endereço de broadcast
1º endereço	192.168.1.0	192.168.1.63
2º endereço	192.168.1.64	192.168.1.127
3º endereço	192.168.1.128	192.168.1.191
4º endereço	192.168.1.192	192.168.1.255

Exemplo 6 – Classe C: 192.168.1.0/24

Agora o objetivo é dividir o endereço para atingir, ao menos, 100 hosts por sub-rede. Usaremos o endereço de classe C 192.168.1.0 com máscara padrão 255.255.255.0. Sabendo o número de hosts desejados, basta calcular quantos bits serão necessários para atingir o número 100.

Assim, usaremos a regra $2^n - 2$, onde n é o número de zeros da máscara atribuídos para endereçar os hosts. No caso de 100 hosts, usaremos 7 bits, pois $2^7 - 2$ é igual a 128.

Executado o cálculo de $2^n - 2$, identificamos que devemos utilizar 7 bits na porção de host para endereçar os 100 hosts. Atenção: os bits do cálculo não se referem aos bits que devemos tomar emprestado, mas sim aos bits usados para endereçar os hosts. Esses 7 bits serão os bits da nova porção de hosts. Para atingir o número de bits da porção de sub-rede, podemos pegar os bits da porção de host original e subtrair os bits de que precisamos, assim, 7 bits. Ao executar a subtração 7 de 8, obtemos um. A porção de sub-rede terá um bit, que equivale ao bit mais significativo da porção de host original. Com o bit da porção de rede, poderemos ter até duas redes com até 126 hosts cada. A figura a seguir nos mostra as porções originais e as obtidas depois de calcular.

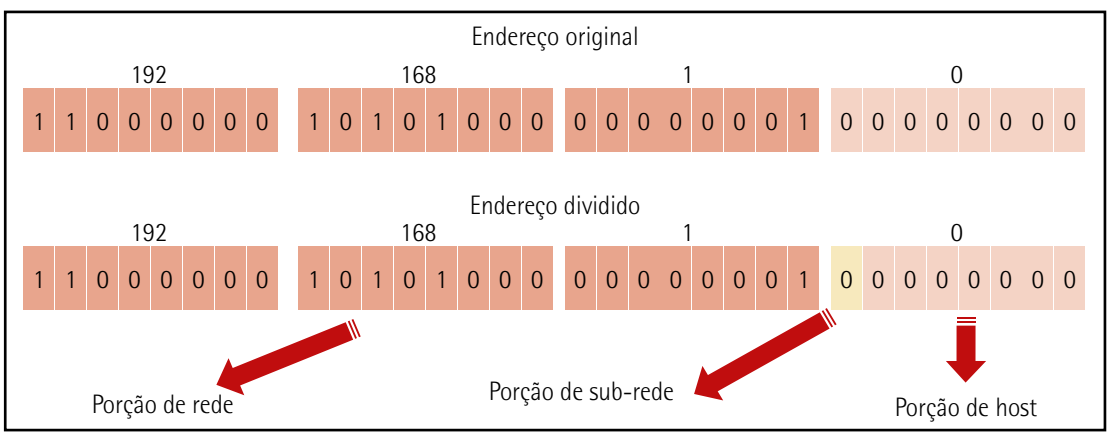


Figura 67 – Porções de sub-rede depois da divisão para o exemplo 6

Observação

Observe com atenção, pois depois de optar por definir o número de hosts como base para realizar o cálculo de sub-rede, o bit a ser emprestado para a porção de sub-rede será aquele de que não necessitamos para obter o número de hosts que precisamos. Após escolher quantos bits restaram para a porção de sub-rede, a atribuição dos endereços IP ocorre da mesma forma que foi apresentada no exemplo 1 e 3. Agora, a máscara de sub-rede terá um bit definido como 1, além dos 24 originais.

Teremos como máscara de sub-rede, em decimal, 255.255.255.128 ou, em binário, 11111111.11111111.11111111.10000000.

Para chegar ao primeiro endereço de rede e seu endereço de broadcast, precisamos atribuir todos os bits da porção de rede com 0 e 1, respectivamente. A figura a seguir mostra a porção de host com todos os bits em zero e a figura seguinte com todos os bits de host definidos em um (broadcast).

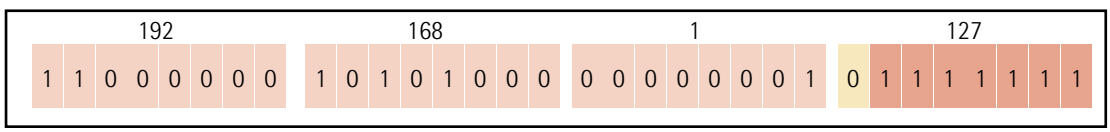


Figura 68 – Endereço broadcast da primeira rede do exemplo 6 (classe C)

O primeiro endereço de host válido para a rede 192.168.1.0/25 é obtido definindo todos os bits de host como zero, exceto o último, assim, o menos relevante. Deste jeito obteremos o endereço 192.168.1.1 como primeiro endereço válido. Para obter o último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast. Atribuímos então o endereço 192.168.1.126 como o último endereço válido.

Os endereços restantes chegarão por meio da manipulação do bit emprestado para a porção de sub-rede. Assim construímos todas as combinações possíveis de zeros e uns para atribuir todas as sub-redes, neste momento pode ser somente 0 ou 1.

Para chegar ao próximo endereço de rede, é preciso adicionar uma unidade ao último octeto do endereço de broadcast. Chegamos ao número 128, que nos dará 192.168.1.128, sendo, assim, o segundo e último endereço de rede da divisão.

A tabela a seguir nos mostra os endereços de rede e broadcast para a divisão do exemplo 6.

Tabela 9 – Endereços de sub-redes e broadcast do exemplo 6 (classe C)

	Endereço de rede	Endereço de broadcast
1º endereço	192.168.1.0	192.168.1.127
2º endereço	192.168.1.128	192.168.1.255

Resumo

Nesta unidade aprendemos sobre as seguintes camadas: apresentação, sessão e rede.

Vimos as principais funções da camada apresentação, que é responsável, basicamente, pelo ingresso da mensagem original para a camada de aplicação, responsável ainda pela compactação dos dados, pela criptografia dos dados e pelo estabelecimento dos padrões de formatação dos arquivos, que são definidos pelas aplicações, como os padrões de imagem GIF, JPG e os padrões de vídeo em MPEG e MP4.

Também aprendemos a interação da camada de sessão, que é responsável pelo intercâmbio das informações e pelo estabelecimento das regras de troca de dados entre as entidades pertencentes a uma comunicação, estabelecendo parâmetros de comunicação, como half-duplex e full-duplex, e igualmente responsável pela distribuição do token de transmissão/recepção que dá o indicativo para cada uma das entidades no momento de transmitir e receber dados.

Avançando pelo módulo aprendemos as funções das atribuições da camada de transporte, na qual o serviço orienta a conexão, a entrega ordenada, a entrega confiável, o controle de fluxo e a identificação das diferentes aplicações nesse nível de serviço. Aprendemos ainda os conceitos da janela deslizante e o estabelecimento de portas para que as aplicações possam interceptar os dados vindo das camadas inferiores.

Além disso, aprendemos as principais funções da camada de rede e as variações dos principais protocolos operados nos dias atuais, como o IP versão 4 e o IP versão 6. Desvendamos os mecanismos de cálculo do protocolo IP, suas variações e ainda o aproveitamento dos mecanismos de endereçamento quando aprendemos as técnicas de sub-redes em todas as classes. Ainda decodificamos as instruções oriundas de todas as interfaces ligadas ao IP versão 6 e, além de explorarmos em detalhes a construção dos datagramas desses dois protocolos, avaliamos a sua importância para as aplicações da atualidade, como o uso na Internet das Coisas.

Exercícios

Questão 1. No modelo de sete camadas da OSI, a camada de apresentação tem a função de:

A) Definir o formato para troca de dados entre computadores.

B) Garantir que os pacotes cheguem ao seu destino livre de erros, sem perdas ou duplicações e em sequência, fornecendo, portanto, uma comunicação fim a fim confiável.

C) Rotear os pacotes da origem para o destino, determinando qual o melhor caminho para fazê-lo, baseado em condições de rede, prioridade de serviço e outros fatores.

D) Estabelecer a conexão entre dois dispositivos físicos compartilhando o mesmo meio físico.

E) Transmitir um fluxo de bits pelo meio físico.

Resposta correta: alternativa A.

Análise das alternativas

A) Alternativa correta

Justificativa: essa é a camada de apresentação. Quando sistemas dissimilares precisam se comunicar, uma tradução e reordenação de byte deve ser feita. Ela é responsável por tradução de protocolos, criptografia, compressão de dados, entre outras tarefas.

B) Alternativa incorreta

Justificativa: essa é a camada de transporte. Essa confiabilidade se dá através de sinais de reconhecimento ACK enviados entre as partes. Fornece também controle de fluxo. O protocolo TCP opera nessa camada.

C) Alternativa incorreta

Justificativa: essa é a camada de rede. Essa camada não está preocupada com a confiabilidade da comunicação, até porque isso já faz parte da camada de transporte. Sua tarefa principal é endereçar os pacotes para o computador de destino. Traduz endereços lógicos em endereços físicos. Gerencia problemas de tráfego em uma rede. O protocolo IP opera nessa camada.

D) Alternativa incorreta

Justificativa: essa é a camada de enlace. Tem a função de detectar e corrigir erros que porventura venham a ocorrer no meio físico, garantindo assim que os frames sejam recebidos corretamente.

Passar os frames de dados da camada de rede para a camada física. Controlar os impulsos elétricos que entram e saem do cabo de rede.

E) Alternativa incorreta

Justificativa: essa é a camada do meio físico. É totalmente orientada a hardware e lida com todos os aspectos de estabelecer e manter um link físico entre dois computadores. Carrega os sinais que transmitem os dados gerados por cada uma das camadas mais altas. Essa camada define como o cabo é ligado ao NIC. Por exemplo, ele define quantos pinos o conector tem e a função de cada um. Além disso, define também qual técnica de transmissão será usada para enviar os dados através do cabo. Fornece codificação de dado e sincronização de bit. Essa camada é, às vezes, referenciada como camada de hardware.

Questão 2. A camada de rede é responsável pela atribuição de endereçamento lógico e também permite a transferência de dados da origem até o destino em uma rede de comunicação. Outro atributo dessa camada é permitir que dispositivos possam se comunicar através de diversas redes interconectadas. Entre os protocolos desenvolvidos para atender às funcionalidades básicas da camada de rede são encontrados o IPv4 e o IPv6.

Com relação às diferenças entre o IPv4 e o IPv6, é possível citar:

I – Número de endereços disponíveis. O IPv6 possui 1,5, o número de endereços do IPv4.

II – A introdução dos endereços de *anycast* no IPv6 e a retirada dos endereços de *broadcast* existentes no IPv4.

III – O tamanho do endereço do IPv4 é de 32 bits, e o do IPv6 de 64 bits.

Estão corretas, apenas, as afirmativas:

A) I e II.

B) II e III.

C) I e III.

D) II.

E) III.

Resposta correta: alternativa D.

Análise das afirmativas

I – Afirmativa incorreta

Justificativa: embora o número de endereços possíveis do IPv6 seja maior que o do IPv4, a relação entre eles é muito maior. O IPv6 pode gerar $3,4x10^{38}$ endereços, e o IPv4 $4,3 x10^{6}$.

II – Afirmativa correta

Justificativa: no IPv6, os endereços broadcast foram substituídos pelos anycast.

III – Afirmativa incorreta

Justificativa: o tamanho do endereço do IPv4 é de 32 bits, e o do IPv6 de 128 bits.