Resumo da Aula 19

Aluno: Luis Vitor Zerkowski - 9837201
Professor: Daniel Batista
Disciplina: Redes de Computadores e Sistemas Distribuídos (MAC-0352)

Introdução

Este resumo inclui unicamente os conteúdos tratados na aula de número dezenove do curso. Dessa forma, três serão os principais temas abordados: o Protocolo da Internet (IP), o processo endereçamento de pacotes, e a técnica de tradução de endereços NAT. Cada seção contará com um breve resumo teórico seguido de aprofundamentos que refletem meu entendimento sobre a aula. Quando dessas explicações, será possível encontrar não apenas texto, como também imagens e esquemas que devem ajudar no entendimento fluido e mais intuitivo do material.

O Protocolo da Internet (IP)

Em se tratando da internet, um tópico de maiúscula importância é a conectividade. Para entender o funcionamento básico desse recurso que hoje faz parte absolutamente intrínseca de nossas vidas, faz-se essencial compreender como os computadores comunicam-se. O Protocolo da Internet, tipicamente abreviado por IP, é parte fundamental desse processo. Sendo um protocolo da camada de rede - o principal protocolo desta camada, diga-se de passagem -, o IP nada mais é do que uma estratégia de identificação dos aparelhos conectados à internet que permite endereçar datagramas entre máquinas distintas. De forma objetiva e bastante simplificada, pois, o IP individualiza computadores na rede e torna-os aptos a receber pacotes - também a mandá-los, uma vez que outros computadores têm seus próprios IP's.

Entendida a natureza do protocolo e percebida sua relevância, passamos agora a melhor descrevê-lo através da detalhada descrição de seu cabeçalho:

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
⋮	⋮																																
60	480																																

Figura 1: Imagem esquemática da estrutura do cabeçalho do protocolo IP.

1. *Version* -> Quatro *bits* responsáveis por identificar a versão do protocolo utilizado para construir o datagrama - IPv4 ou IPv6.

2. *Internet Header Length* (IHL) -> Quatro *bits* responsáveis por indicar o tamanho do cabeçalho em blocos de trinta e dois *bits*, permitindo que o receptor do datagrama diferencie esse cabeçalho do conteúdo do pacote.
3. *DiffServ Code Point* (DSCP) -> Seis *bits* utilizados para fazer controle de diferenciação de serviços. Basicamente serve como código para identificar certas ações especiais a serem realizadas sobre um datagrama - torná-lo prioritário na fila de pacotes de um roteador, por exemplo. Na prática, por conta dos princípios de neutralidade da rede, os roteadores não são configurados para interpretarem o DSCP, o que o confere funcionalidade plena apenas em redes privadas.
4. *Explicit Congestion Notification* (ECN) -> Dois *bits* usados para notificar que há congestionamento de rede aos usuários de uma certa rota. Serve, portanto, como um alerta dos roteadores às máquinas que estão utilizando-os para que possam tomar alguma providência e prevenir grandes perdas de pacote por alto congestionamento da rede.
5. *Total Length* -> Dezesseis *bits* responsáveis por indicar o tamanho total do datagrama, incluindo cabeçalho e o próprio conteúdo do pacote - *payload*.
6. *Identification* -> Dezesseis *bits* que trabalham em conjunto com os próximos dois campos do cabeçalho para permitir que datagramas maiores do que o *Maximum Transmission Unit* (MTU) possam ser devidamente transmitidos. Toda rede naturalmente apresenta um tamanho máximo de pacote que consegue lidar, isso por conta das capacidades de transmissão e armazenamento dos roteadores. Para respeitar essa limitação de MTU e ainda assim ser capaz de transmitir pacotes de tamanho arbitrário, por vezes é preciso fragmentar os datagramas. Para corretamente reconstruir o pacote original quando de sua chegada no destino, é preciso, antes de mais nada, identificar os vários fragmentos do tal datagrama. Esse processo é feito através da leitura dos identificadores do pacote. Datagramas com campos *identification* iguais, portanto, são fragmentos do mesmo pacote.
7. *Flags* -> Três *bits* utilizados para sinalizar propriedades e ocorridos relacionados à fragmentação de datagramas - se um pacote tem outros fragmentos ainda ou se não pode ser fragmentado de forma alguma, por exemplo.
8. *Fragment Offset* -> Treze bits utilizados em conjunto com os *bits* de *identification* caso haja fragmentação do datagrama. Servem para indicar o ponto de ligação - em *bytes* e a partir do início do primeiro pacote - entre os datagramas fragmentados, permitindo, dessa forma, que a reconstrução do pacote original seja feita não só com os fragmentos corretos através do número de *identification*, mas na ordem correta.
9. *Time To Live* (TTL) -> Campo de oito *bits* que determina quantas vezes o pacote pode ser transmitido antes de indicar falha de transmissão. Esse campo serve principalmente para evitar que o datagrama entre em *loop*, escapando da possibilidade de ser enviado para uma sequência de roteadores num ciclo infinito. Toda vez que o pacote em transmissão passa para o próximo roteador, portanto, o valor de seu TTL é decrementado e, caso chegue a zero em algum momento, os roteadores são configurados para reenviar o pacote de volta à origem. Para o devido funcionamento dessa estratégia, é importante destacar que a internet como a conhecemos hoje é construída de maneira tal que uma máquina está a no máximo sessenta e quatro nós de

outra, o que permite que usemos tal campo do cabeçalho sem a confusão de achar que o pacote não chegou ao destino por conta de um TTL pequeno demais.

10. *Protocol* -> Oito *bits* indicando o tipo de protocolo sendo usado na camada de transporte - TCP ou UDP, por exemplo. Serve para preparar o sistema do destinatário para o tipo de comunicação que será estabelecida.
11. *Header Checksum* -> Dezesseis *bits* utilizados para verificação da integridade do protocolo através do cálculo da soma de todos os *bits* do datagrama. Caso o destinatário some todos os *bits* do pacote e encontre um número diferente do que indica esse campo do datagrama, fica evidente que o pacote não foi transmitido intacto. Essa é uma das medidas de proteção contra possíveis alterações do pacote original mais básicas e amplamente utilizadas por vários protocolos de várias camadas.
12. *Source IP Address* -> Trinta e dois *bits* responsáveis por identificar o IP da máquina remetente.
13. *Destination IP Address* -> Trinta e dois *bits* responsáveis por identificar o IP da máquina destinatária.
14. *Options* -> Segmento do cabeçalho de tamanho variável composto por blocos de trinta e dois *bits*. Caso o IHL seja maior do que cinco, esse campo do protocolo será de tamanho (IHL - 5)*32 *bits*. Caso o IHL não seja maior do que cinco, esse campo não existirá. Vale ressaltar também seu valor máximo de dez blocos de trinta e dois *bits*. Falando agora sobre o conteúdo do campo, pode ser bastante diverso, perpassando por informações de nível de segurança do pacote em questão, marcações de tempo do datagrama para todo roteador pelo qual passa, e outras das mais variadas.

Endereçamento

Esta seção trata de endereçamento, natural continuação do protocolo IP, uma vez que seu principal objetivo é justamente a identificação de pacotes. Aqui trataremos mais diretamente de como funciona o protocolo na prática, discutindo um pouco mais sobre os tais endereços IP de cada máquina, subredes, e como funciona a troca de pacotes entre aparelhos de origem e destino.

Apesar do IP ser um protocolo da camada de rede, é também utilizado para designar um número identificador de uma máquina na rede. Para abordarmos endereçamento de forma mais ampla, nos ateremos agora a essa segunda atribuição dos IP's. Dessa forma, de agora em diante devemos pensar no IP como um número que identifica unicamente cada aparelho conectado à internet.

Esse número cuja principal propriedade é a unicidade relacionada, pode vir de duas principais formas: o IPv4 ou IPv6 - variedade essa que se deve à versão do protocolo implementada. O IPv4, forma que tomaremos como base para o resto deste documento por simplicidade, segue um padrão muito bem estabelecido que se dá através de um número de trinta e dois *bits* separados por pontos em quatro grupos de oito *bits*. Segue abaixo um pequeno esquema ilustrativo de sua configuração:

Padrão:

xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

Exemplo:

11011111.00000001.00000001.00000001

OBS: Nota-se aqui a importância do desenvolvimento e utilização do IPv6, uma vez que estamos alcançando cada vez mais dispositivos identificados na rede e o IPv4 tem apenas 2^32 = 4,294,967,296 identificadores únicos disponíveis - número esse completamente arbitrário e escolhido há muito.

Por uma questão de fluidez na leitura, no entanto, o IP acaba não sendo representado dessa forma para humanos. A nível esquemático, pois, o IP caracteriza-se da seguinte maneira:

Padrão:

(0-255).(0-255).(0-255).(0-255)

Exemplo:

11011111.00000001.00000001.00000001 = 223.1.1.1

Trazendo um pouco mais de realidade a esses identificadores, não seria possível utilizá-los de maneira única para todos os computadores no mundo. Por conta disso, o IP é dado, na verdade, apenas para máquinas que estão verdadeiramente conectadas à internet, sendo tipicamente designados para *hosts* - servidor responsável por algum serviço - ou para roteadores. Os IP's fornecidos aos computadores de uso mais comum, pois, são fictícios e funcionam exclusivamente dentro de uma rede menor para facilitar a diferenciação de várias comunicações por parte de um roteador. A essas redes menores populadas com IP's virtuais dá-se o nome de subredes.

Uma subrede, desse modo, nada mais é do que um conjunto de IP's teóricos que permitem dois principais processos: comunicação local com outros computadores sem intermédio da internet, e comunicação com a internet via roteador. Ambos os fenômenos advêm da tentativa de lidar com a já citada dificuldade de oferecer um endereço de IP único para todo e qualquer dispositivo que queira fazer uso da internet. Por conta da falta de IP's, as máquinas não *host* e nem roteadores não são alcançáveis por outras máquinas na internet - seja via DNS ou via endereço de IP diretamente. Conectadas a uma subrede, contudo, máquinas conseguem comunicar-se através do uso de seus IP's fictícios. Para além disso, devido à possibilidade de diferenciar máquinas numa subrede, o roteador, que efetivamente é dono de um IP e comunica-se com a internet, consegue endereçar devidamente as respostas a requisições fora da subrede com os computadores que as fizeram.

Entendido o funcionamento e o potencial de uma subrede, ainda permanece o questionamento de como funciona a criação e distribuição dos IP's. Para o roteador ou *host*, dono do verdadeira IP, é preciso contatar uma agência de alocação de endereços que disponibilizará um ou mais endereços para o dispositivo. Já para as máquinas dentro da subrede, um IP aleatório é dado para um novo computador que se conecta na subrede, respeitando apenas o *Classless InterDomain Routing* (CIDR) da rede - determinação da parte

do endereço IP modificável. Para melhor entendimento do CIDR, segue um esquema ilustrativo:

Padrão:
a.b.c.d/x -> x indica o número de bits fixo na subrede.

Exemplo:

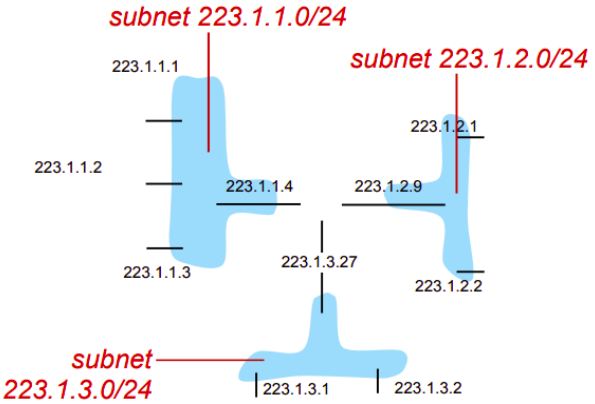

Figura 2: Imagem ilustrativa com três subnets, todas elas com vinte e quatro *bits* fixos e apenas oito *bits* modificáveis para distribuição de IP's virtuais.

Network Address Translation **(NAT)**

Essa última seção do resumo trata do *Network Address Translation* (NAT), processo responsável por fazer justamente a devida alocação das respostas advindas da internet na comunicação dos dispositivos conectados a uma subrede. O NAT, portanto, é basicamente uma tabela implementada no roteador capaz de traduzir respostas advindas da *Wide Area Network* (WAN) para seu endereço de IP numa certa porta para *Local Area Network* (LAN) no IP virtual da máquina que realmente fez a requisição e na porta correta por onde esta máquina fez a tal requisição. Segue, por fim, uma imagem ilustrativa do funcionamento em alto nível do NAT:

NAT translation table	
WAN side	LAN side
138.76.29.7, 5001	10.0.0.1, 3345
. . .	. . .

Figura 3: Imagem que mostra um exemplo de tradução da tabela NAT. Nesse caso, a tabela transforma uma requisição enviada pelo computador no IP virtual 10.0.0.1 pela porta 3345 numa mensagem enviada pelo IP verdadeiro - do roteador - 138.76.29.7 pela porta 5001. Na resposta da requisição enviada posteriormente pelo serviço com o qual o computador tenta comunicar-se, a mensagem será traduzida de volta de 138.76.29.7 na porta 5001 para 10.0.0.1 na porta 3345.