**Capítulo 4 - Camada de Rede**

**Rede de Datagrama:** - camada de rede sem conexão - não existe estabelecimento de chamada na camada de rede - pacotes encaminhados baseado no endereço de host de destino(podem percorrer destinos diferentes) - elástico.

**Rede de Circuito Virtual:** - camada de rede orientado a conexão - desempenho - estabelecimento da chamada antes do fluxo - identificador **VC.**

**VC:** caminho de origem ao destino - números VC - entradas nas tabelas de encaminhamento - número VC pode ser diferente para cada enlace do caminho.

**Arquitetura de roteadores:** funções chaves(**plano de controle:** executar protocolos de roteamento - **plano de dados:** comutar datagramas do enlace de entrada para o de saída).

**Estruturas de comutação:** 1 - comutação via memória(controle direto do CPU, pacote é copiado para memória do sistema, limitado a largura de banda do acesso a memória). 2 - comutação via barramento(barramento de entrada e saída compartilhado, limitado pela largura da banda do barramento). 3 - comutação via rede de interconexão(supera limitação de banda de barramento, fragmentar o datagrama em células de tamanho fixo, comutar as células e remontar na saída).

Filas = chegada > transmissão - Disciplina de escalonamento = escolhe os datagramas com prioridade.

**Bufferização:** chegada datagramas > transmissão datagramas, overflow.

Tamanho do buffer(**RFC3439**) - RTT\*C(capacidade do enlace)

Recomendação recente -( RTT\*C)/raiz(N) -> N fluxos

**HOL** - datagrama na cabeça da fila impede que outros na fila sejam comutados para a porta de saída -> atraso na fila e perda devido ao overflow do buffer de entrada.

**IPV4:** datagramas podem ser fragmentados dentro da rede e remontados no destino final - cabeçalho usado para remontagem - cabeçalho(20bytes) - identificador 32 bits - endereços IP são associados com interfaces e não com hospedeiros ou roteadores- subrede(interfaces com o mesmo endereço de subrede) - CIDR(máscara de subrede de tamanho arbitrário - formato a.b.c.d/x onde x corresponde a parte da sub rede no endereço).

**DHCP:** obtém dinamicamente o endereço IP a partir de um servidor:plug and play - permite o reuso de endereços - pode renovar.

**Como ISP obtém bloco de endereços? ICANN**

**NAT:** rede local usa somente um endereço IP para seu relacionamento com o mundo externo - alterações na rede local podem ocorrer sem a necessidade de notificar o mundo externo. Implementação roteador NAT-> datagramas de saída/entrada e armazenamento. 16 bits -> mais 60000 conexões - NAT é controverso - deveria estabelecer conexão com a outra parte diretamente. IPV6 para resolver - viola argumento fim-a-fim. Soluções: configurar estaticamente o NAT, UPnP(aprender endereço público), encaminhamento.

**ICMP:** usado para comunicar informações de nível 3 - informação de erro - camada de rede acima do IP

**IPV6:** esgotamento endereço de 32 bits - novo formato de cabeçalho ajuda na eficiência do processamento- cabeçalho com 40 bytes - não é permitido fragmentação -> prioridade entre os datagramas de fluxo, flow label(identifica datagramas em um mesmo fluxo), next header(identifica o protocolo da camada superior) => checksum removido, nova versão ICMPv6(packet too big).

**Transição IPV4 para IPV6:** tunelamento => IPV6 transportado como a carga nos datagramas IP enviados entre roteadores IPV4.

**Algoritmos de roteamento:**

* global: todos os roteadores possuem a informação sobre a topologia da rede, LS(link state)
* descentralizado: conhece as informações dos vizinhos, processo iterativo, DV(distance vector)
* estático: portas mudam esporadicamente
* dinâmico: atualização periódica automática em resposta ás alterações nos custos dos enlaces.

**Estado de enlace:** aprender algoritmo Dijkstra=>O(n²), pode O(nlogn), pode ter oscilações, informar custos de links incorretos, impacto de erros é limitado.

**Vetor de distância:** aprender algoritmo => assíncrono, iterativo, cada iteração local é disparado por alteração no custo de algum enlace, distribuído, trocas somente com vizinhos, pode haver loops de roteamento, problemas de contagem ao infinito, link incorreto, propagação pela rede.

**Roteamento hierárquico:** regiões chamadas de sistemas autônomos(AS) - roteadores do mesmo AS rodam o mesmo protocolo de roteamento - intraAS => destinos internos - interAs e intraAs => destinos externos, poupa espaço na tabela, questões políticas, desempenho….

**Roteamento na Internet:** intraAs mais comuns => rip, ospf, igrp

**RIP** => VD, max saltos 15, trocados a cada 30s, até 25 redes, enviados em pacotes UDP, 180 ta morto

**OSPF** => LS, anúncios por toda a AS, transportado sobre IP, segurança(autenticadas), OSPF hierárquico(grandes domínios).

**BGP:** usado na internet, propagar para roteadores internos, alcance sub-redes vizinhos, sobre TCP, eBGP(exterior), iBGP(interior)

**Roteamento Broadcast:** duplicação na rede=> flooding = recebe um pacote e enviar para todos os vizinhos(ciclos e tempestades broadcast) - flooding controlado = só faz broadcast se tinha enviado o mesmo pacote - spanning tree = nenhum pacote redundante é recebido por qualquer um dos nós da rede(cria a arvore e enviam copias atraves dela)

**Roteamento Multicast:** utiliza-se endereçamento indireto(identificador único - endereços classe D) => encontrar uma arvore conectando memros multicast

Arvore baseada no emissor(rpf)(LS) / Unica arvore pelo grupo

**IGMP:** opera entre o host e roteador default => membership\_(query(todos os hosts), report(host pertecente a grupo multicast), group(opcional))

**RPF:** baseada no conhecimento do roteador sobre o caminho mais curto unicast até o emissor

**Capítulo 4 - Camada de Enlace**

**Serviços:** enquadramento=> encapsula datagrama, adiciona cabeçalho, controle de fluxo, detecção de erros, correção de erro, hald-duplex(ambos lados podem transmitir exceto ao mesmo tempo) e full-duplex .

Implementada em um adaptador, conectada ao barramento do sistema.

**Detecção de Erro:** EDC error detection and correction, check de paridade, checksum, CRC = R = resto(d\*2^r /G) (D|CRC)/G = 0 => nao houve erro

r = número de bits de G

**Protocolos de acesso múltiplo:** ponto a ponto, broadcast

canal de broadcast único compartilhado => duas ou mais sinais são enviados(colisão) => identificação de colisão usa o próprio canal

**Protocolos MAC:** particionamento de canal em tempo, frequência, código - acesso aleatório(canal não é dividido, recuperação quando de ocorrência de colisões), passagem de permissão. TDMA = time, FDMA = frequency

Acesso aleatório => Slotted Aloha(quadros possuem o mesmo tamanho), particionado=>altamente descentralizado, simples(pros) - colisões, clock de sincronização(contras) ~37 de eficiencia => p(1-p)^(n-1) p(1-p)^2(n-1) ->

Acesso aleatório => simples, sem sincronização => transmite imediatamente ~18%

Acesso aleatório=> CSMA => escuta antes de transmitir, atraso de propagação pode fazer que aconteça colisões

Acesso aleatório=> CSMA/CD => colisões detectadas num tempo mais curto

Passagem de permissão =>polling(alta latência, única falha)

Passagem de permissão=>token passing(token overhead, latência)

**Endereços MAC e ARP:** mac => endereço físico gravado na rom(48bits)

mesma lan => A broadcast ffffffff => B unicast back A

lans diferentes => A cria quadro ethernet com ip do destino B, nas redes locais usa arp pra identificar.

**Ethernet:** preâmbulo usado para sincronizar as taxas de relógio, sem conexão, não confiável, mac ethernet: unslotted CSMA/CD => k aleatoriamente entre 0 e 2^m -1 onde m =min(n, 10), espera k x 512 tempos de bit

e = 1/(1 + 5tprop /ttrans) prop=atraso máximo propagação entre dois nós, ttrans = tempo de transmissão do quadro de tamanho máximo

**Hubs:** repetidores de camada física, todos os outros canais, podem colidir, não há bufferização, csma/cd

**Switch:** examina o cabeço seletivamente, usa CSMA/CD, hosts não tem conhecimento do switch, não necessitam de configuração, transmissões simultâneas sem colisões, bufferiza pacotes, tabela de comutação, switch aprende quais hosts podem ser alcançados a partir das interfaces de swtitch

**Roteadores** mantem tabelas de roteamento, implementam algoritmos

**Swichs** mantem tabelas de comutação, implementam filtrose algoritmos de aprendizagem.

DHCP E ARP = L2 BROADCAST