
Routers IP

*FEUP/MRSC/AMSR
MPR*

Bibliografia

- » Aula preparada com base seguinte bibliografia
 - S. Keshav, “An Engineering Approach to Computer Networking”, Addison-Wesley, 1997
 - L. Peterson and B. Davie, “Computer Networks”, Morgan Kaufmann, 2000
 - S. Keshav, R. Sharma, “Issues and Trends in Router Design”
 - C. Partridge et al, “A Fifty Gigabit per Second IP Router”
 - Cisco White Paper, “The Evolution of High-End Router Architectures”
 - N. McKeown, “Achieving 100% Throughput in an Input-Queued Switch”
 - M. Ruiz-Sanchez, E. Biersack, W. Dabbous, “Survey and Taxonomy of IP Address Lookup Algorithms”
 - P. Gupta, S. Lin, N. McKeown, “Routing Lookups in Hardware at Memory Access Speeds”
 - P. Gupta, N. McKeown, “Algorithms for Packet Classification”
 - J. Mogul, K. Ramakrishnan, “Eliminating Receive Livelock in an Interrupt-driven Kernel”
 - B. Chen, R. Morris, “Flexible Control of Parallelism in a Multiprocessor PC Router”
 - S. Karlin, L. Peterson, “VERA: An Extensible Router Architecture”



Introdução

- ◆ Router → elemento de rede
 - » Usado para interligar redes heterogéneas, criando a ilusão de rede (IP) única
 - » Equipamento activo

- ◆ Função de um router
 - » Transferir pacotes: portas de entrada → portas de saída

 - » Mas também,
 - Suportar tecnologias heterogéneas de rede (nível 2)
 - Diferenciar os serviços de transporte
 - Participar em algoritmos distribuídos de identificação de rotas

Routers

- ◆ Routers em todos os níveis de rede (IP)
 - » Redes de acesso → 
 - Interligação de pequenas empresas/ clientes domésticos ao ISP (Internet Service Provider)
 - » Redes empresariais
 - Interligam até dezenas de milhar de computadores
 - » Redes de transporte → 
 - Interligam ISPs ou redes empresariais através de ligações longas
 - Não directamente acessíveis a computadores

- ◆ Projecto de routers → requisitos diversos
 - » Redes de acesso
 - Muitas portas. Heterogéneas. Variedade de protocolos
 - » Redes empresariais
 - Baixo custo por porta. Muitas portas. Configuração fácil. Suporte de QoS
 - » Redes de transporte
 - Encaminhar a velocidades elevadas para poucos links

Router de Transporte

- ◆ Interliga ISPs e grandes redes empresariais
- ◆ Custo de links de transmissão normalmente elevado →
 - » custo do router não constitui a maior restrição
- ◆ Requisitos importantes → fiabilidade e tempo de comutação
 - » Fiabilidade → técnicas semelhantes às da rede telefónica
 - Fonte de alimentação redundante
 - Caminhos de dados duplicados (dentro do router)
 - » Tempo de comutação → inspeção da tabela de encaminhamento
 - Tabela de encaminhamento pode conter milhares de entradas

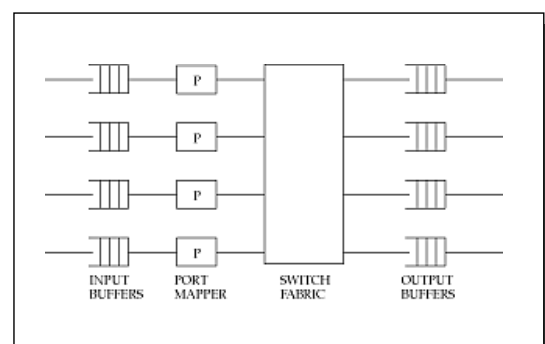
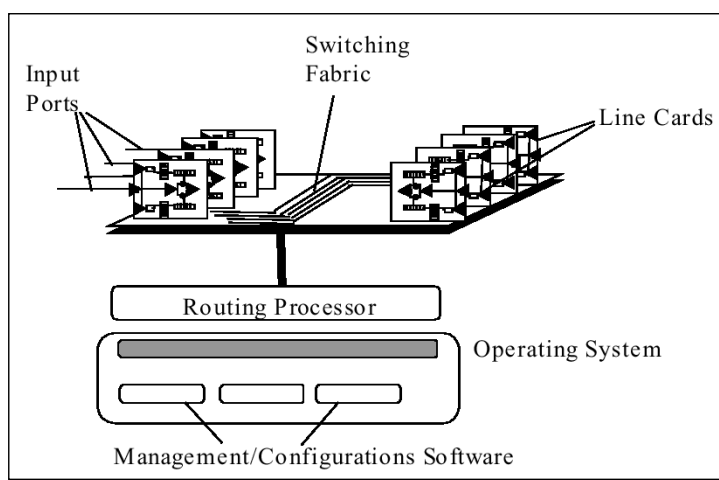
Router / Switch de Rede Empresarial

- ◆ Pode interligar número grande de portas
- ◆ Suporte de serviços diferenciados → QoS dentro da empresa
- ◆ Suporte adicional de
 - » tráfego multicast
 - » múltiplos protocolos de rede (IPv4, IPv6, IPX)
 - » Firewalls, filtros de tráfego, políticas de segurança, VLANs

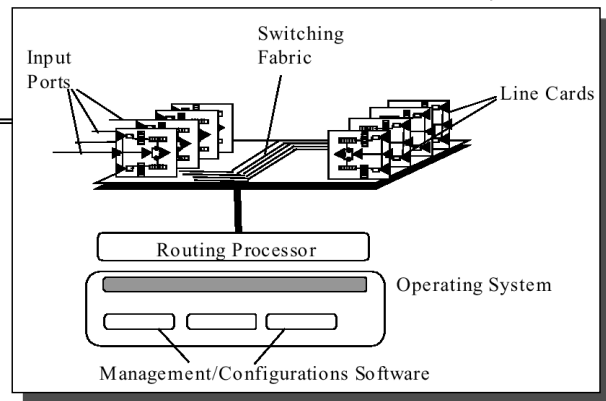
Router de Acesso

- ◆ Interliga utilizadores domésticos, pequenas empresas aos ISP
- ◆ Rede de acesso
 - » Tradicionalmente
 - bancos de modems,
 - interligados a concentradores de terminais,
 - servindo um grande número de ligações telefónicas de baixo débito
 - » Mudanças significativas
 - Tecnologia de acesso diversificada → modems de alta velocidade, ADSL e modems de cabo
 - Protocolos mais complexos em cada porta → SLIP, PPP + PPTP, IPSec
- ◆ Suporte de
 - » número elevado e heterogéneo de portas (podem ser de alto débito)
 - » Protocolos diversos e complexos
- ◆ Projecto complexo!

Modelização de um Router (Keshav)



Componentes de um Router



◆ Router genérico, 4 componentes

» Portas de entrada

- Ligação ao link físico. Entrada de pacotes.
- Portas agrupadas em cartas (4, 8, 16 portas)

» Portas de saída

- Armazena pacotes. Escalonamento de serviço no link de saída

» Comutador

- Interliga portas de entrada e saída
- Define o tipo de router

- ◆ Fila à saída → $B_{comutador} > \sum B_{portaEntrada}$

- ◆ Fila à entrada → $B_{comutador} < \sum B_{portaEntrada}$

» Processador de rotas

- Protocolos de routing . Criação de tabela de encaminhamento (de pacotes)

Porta de Entrada

◆ Desencapsulamento do nível 2

◆ Selecção da porta de saída

- » Identificação de endereço de destino dos pacotes
- » Inspecção da tabela de encaminhamento.
- » Selecção da porta saída

◆ Classificação dos pacotes em classes pré-definidas

- » garantias de QoS

◆ Implementação de protocolos de

- » nível 2 (PPP, SLIP) ou 3 (PPTP, IPSec)

◆ Controlo do acesso ao comutador

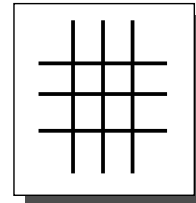
Comutador

» Barramento

- Interliga todas as portas de entrada e saída
- Débito limitado →
 - ◆ pela capacitância do barramento, por overhead da lógica de acesso

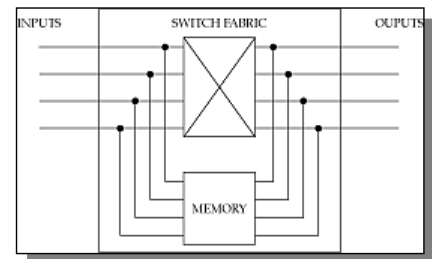
» Matriz

- 2N barramentos. N*N pontos de contacto
 - ◆ Transferência simultânea de pacotes
- Escalonador →
 - ◆ abre, fecha pontos de contacto
 - ◆ limita velocidade de comutação/ débito



» Memória partilhada

- Pacotes armazenados em memória partilhada
- Comutação de apontadores ou cabeçalhos
- Débito do comutador limitado pelos tempos de acesso à memória



Porta de Saída, Processador de Rotas

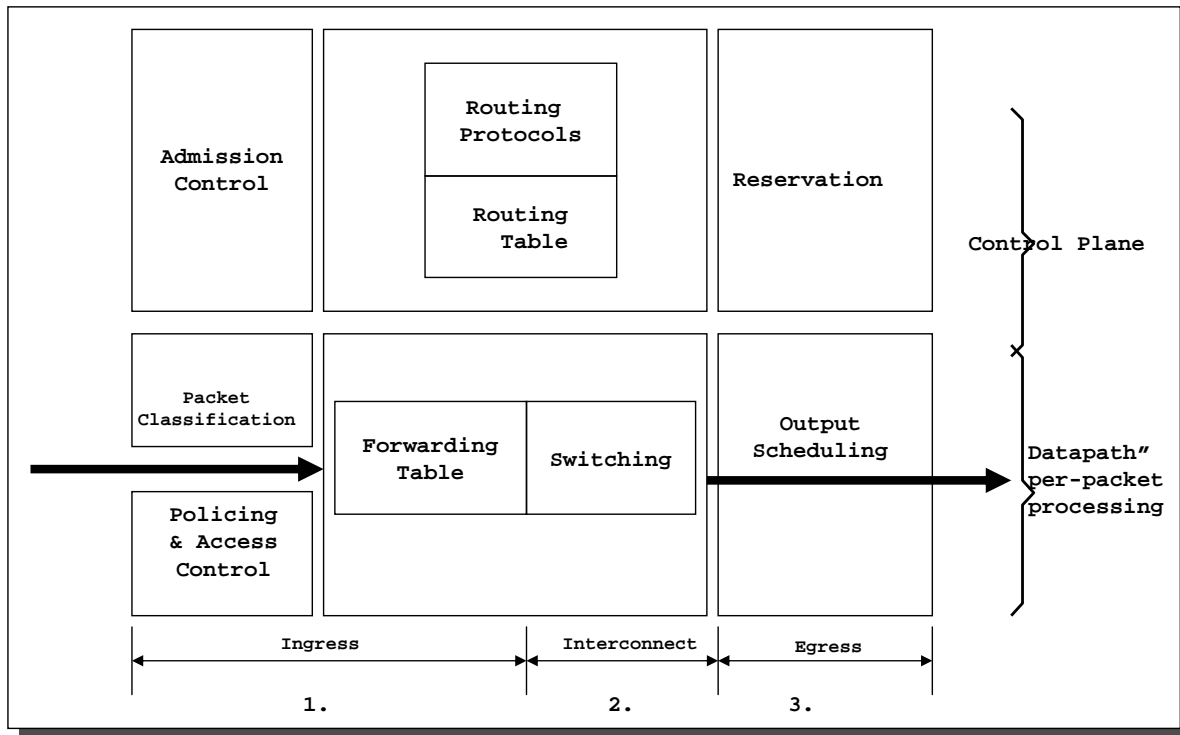
» Porta de saída

- Armazena pacotes antes da transmissão pelo link de saída
- Podem implementar algoritmos de escalonamento (prioridades, garantias)
- Encapsulamento e implementação de protocolos de nível 2

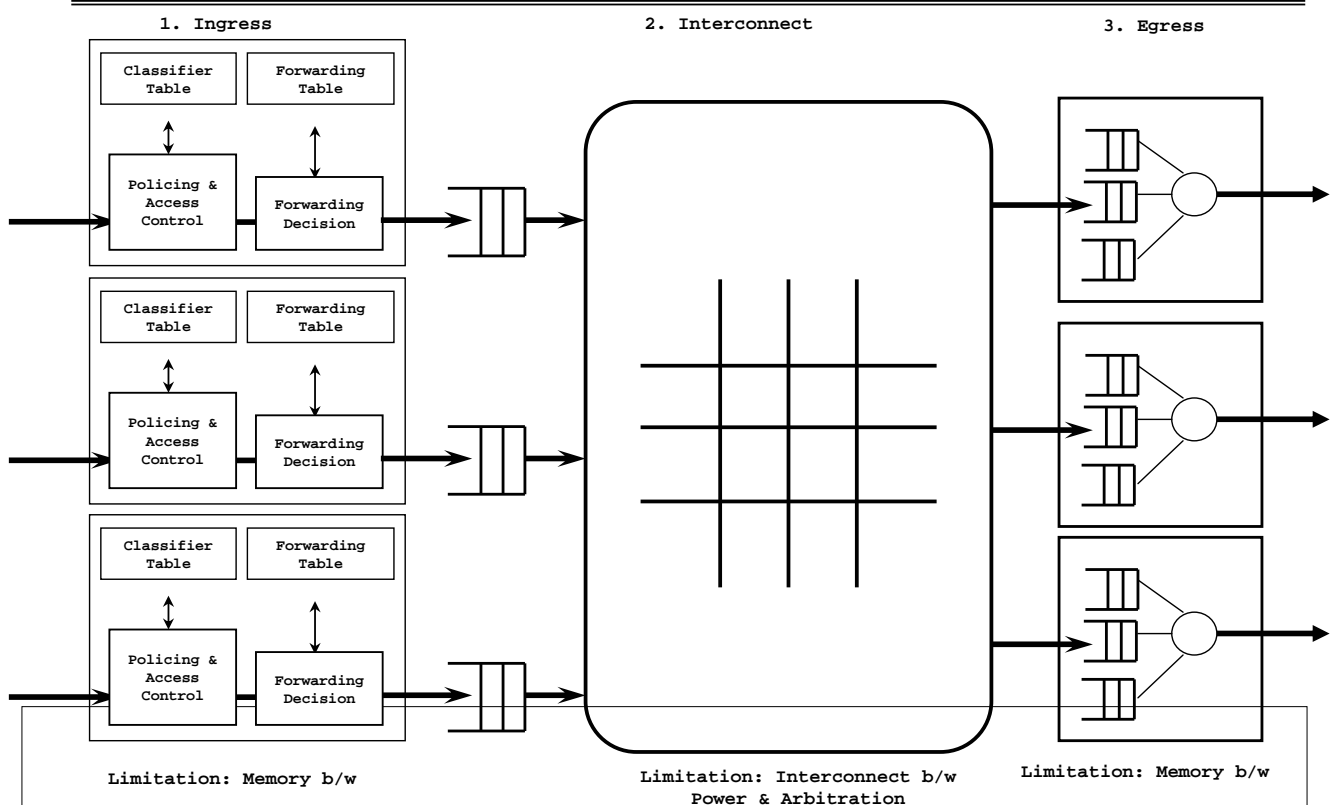
» Processador de rotas

- Calcula tabelas de rotas e de encaminhamento
- Executa protocolos de rotas
- Executa software de configuração e gestão do router
- Trata pacotes não explicitados na tabela de encaminhamento da carta de entrada

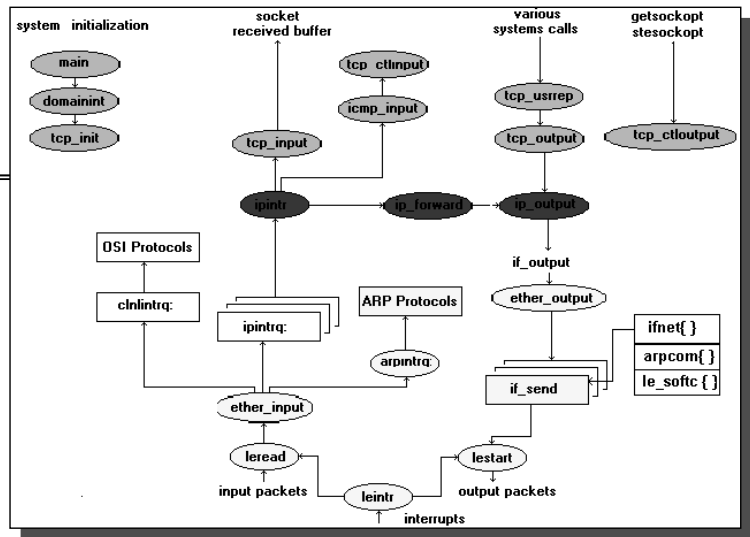
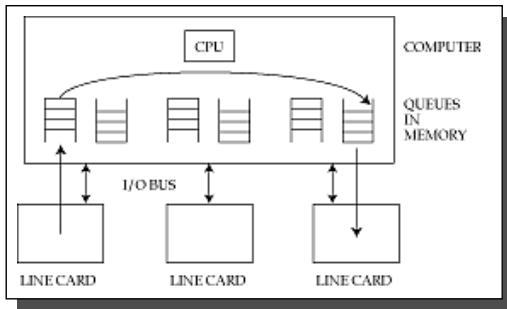
Representação Alternativa (McKeown)



Representação Alternativa



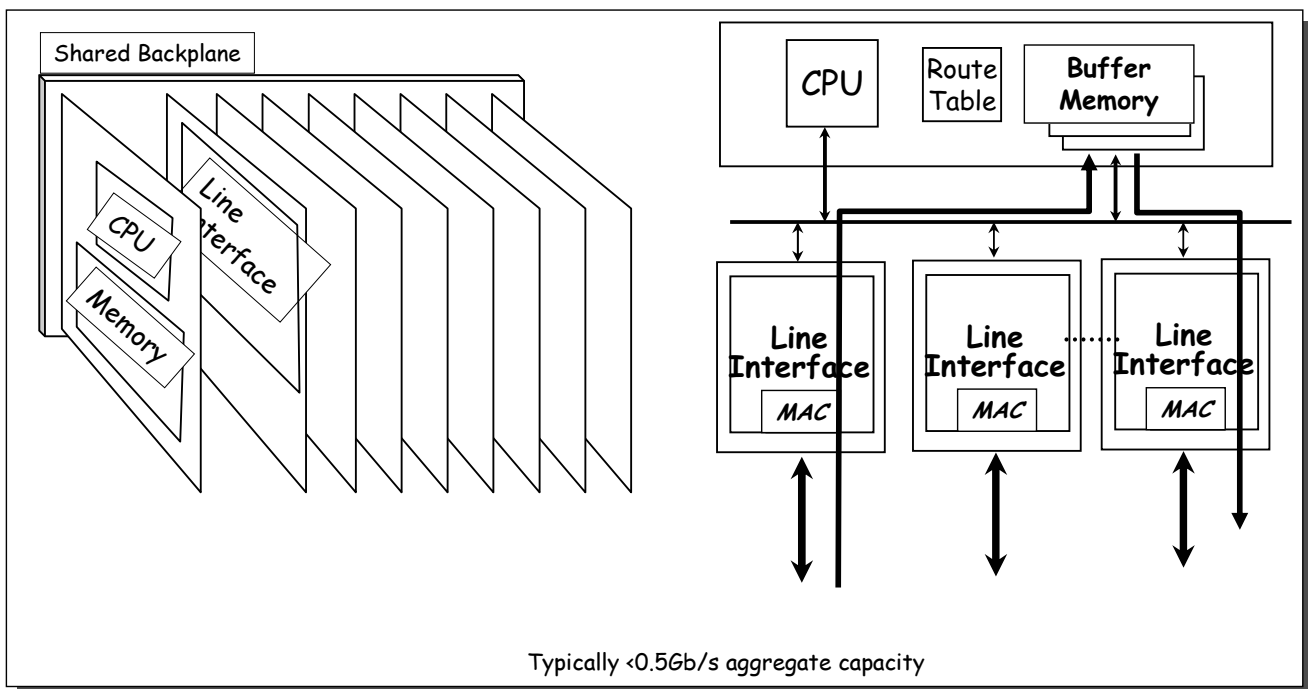
Router de Primeira Geração



- » Routers (e comutadores Ethernet) de baixo custo
- » Bottleneck
 - CPU, barramento
 - Software →
 - ◆ processamento normal de pacotes é interrompido com chegada de novos pacotes
 - ◆ Mau comportamento em cargas elevadas → latência grande, débito baixa

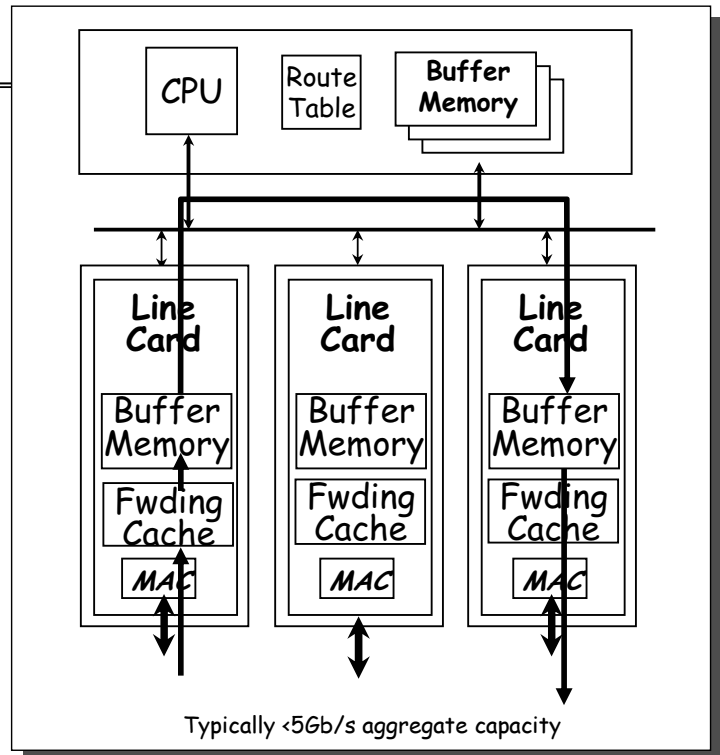
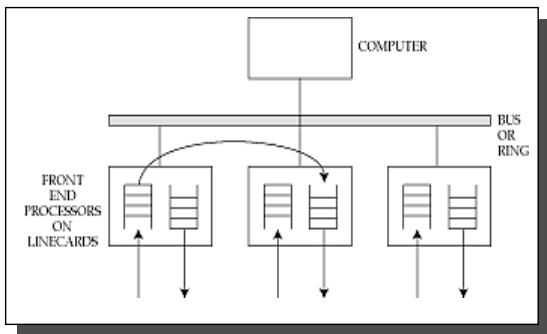
RT 16

Router de Primeira Geração



Router de Segunda Geração

RT 17

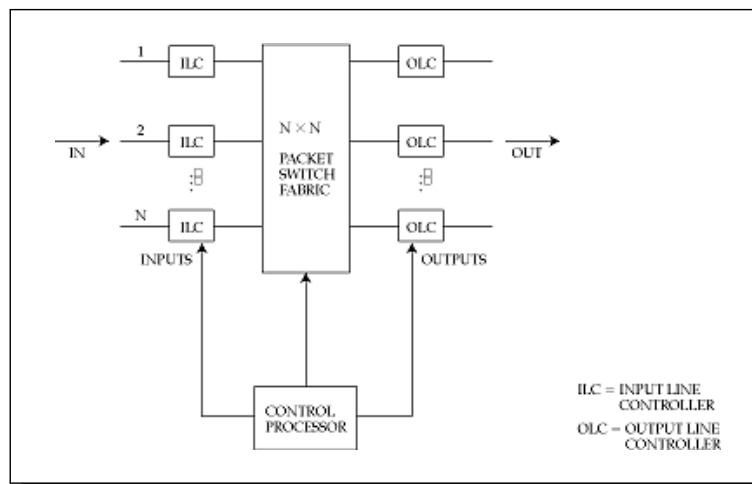


» Cartas inteligentes → encaminham directamente

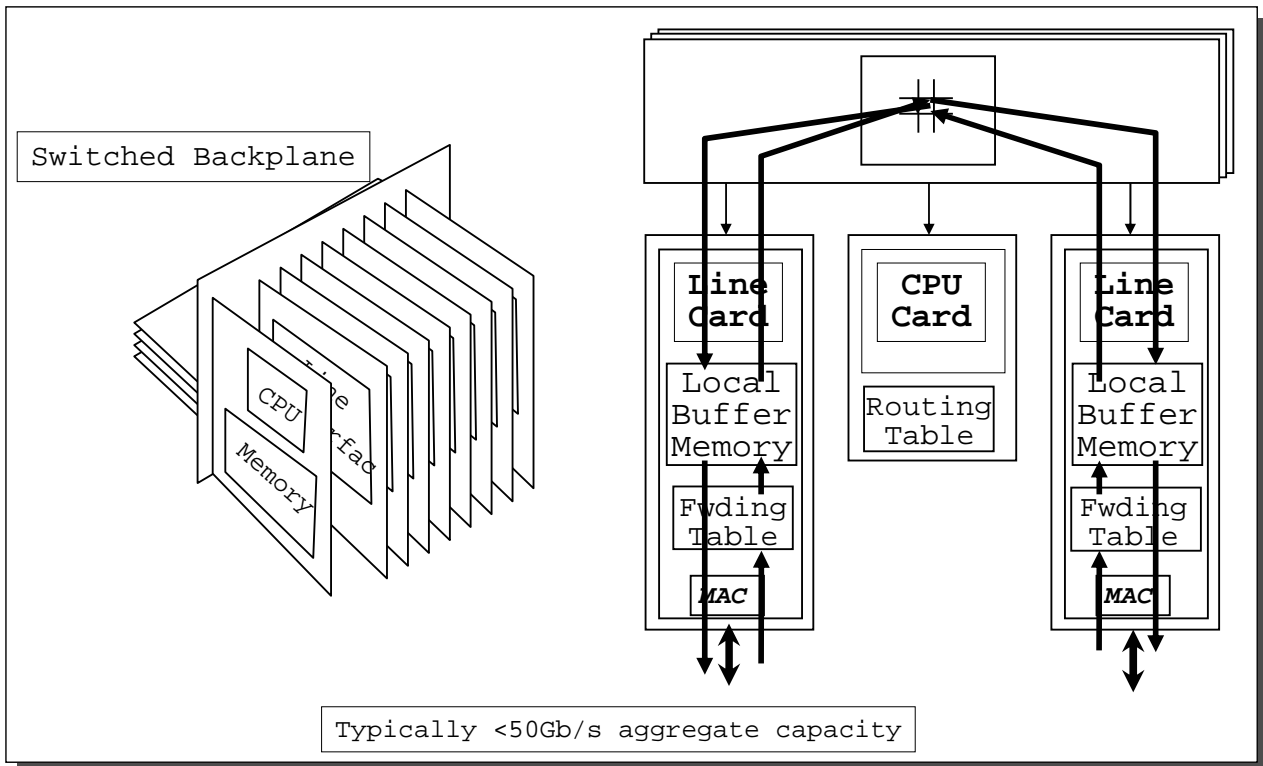
RT 18

Router de Terceira Geração

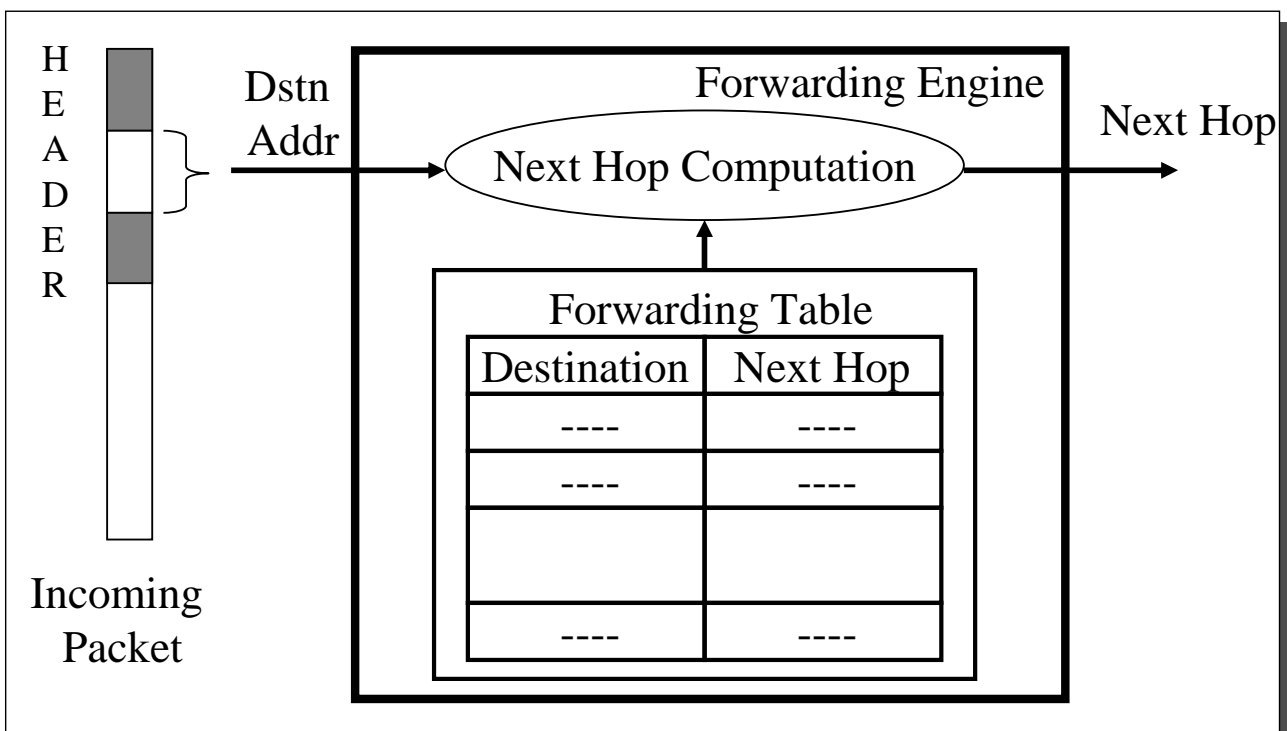
» Equipado com matriz → trajectos paralelos



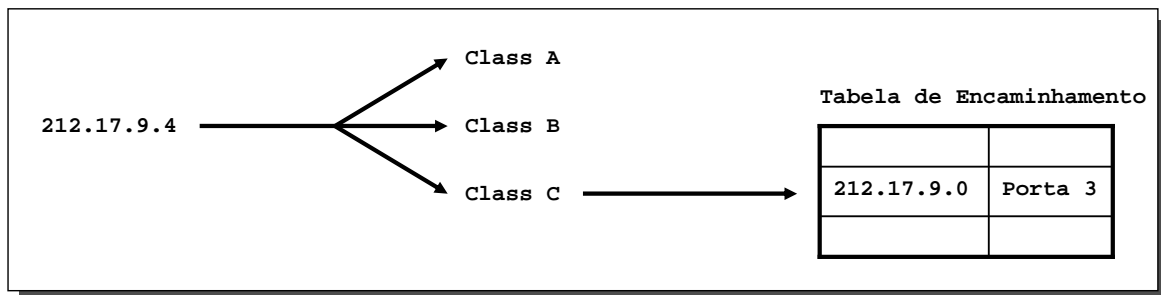
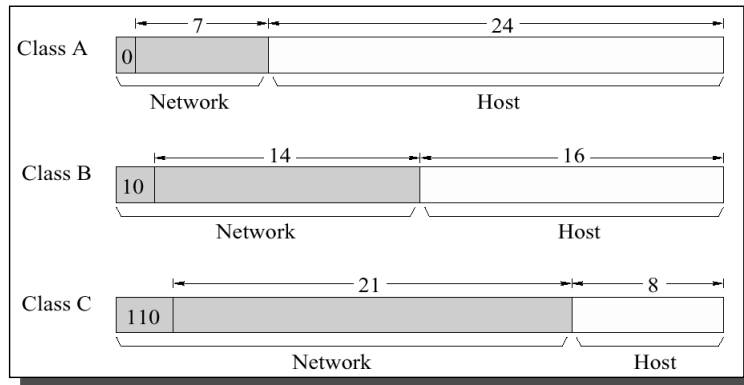
Router de Terceira Geração



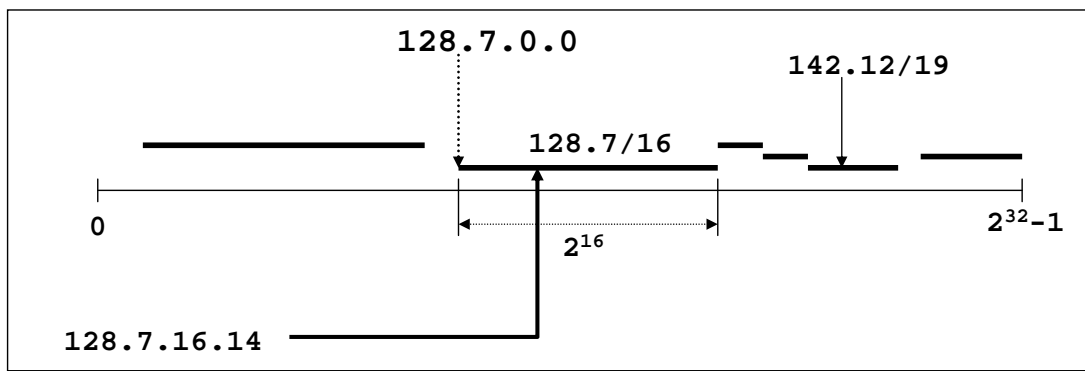
Encaminhamento de um Pacote



Endereços Baseados em Classes



CDIR – Classless Inter-Domain Routing



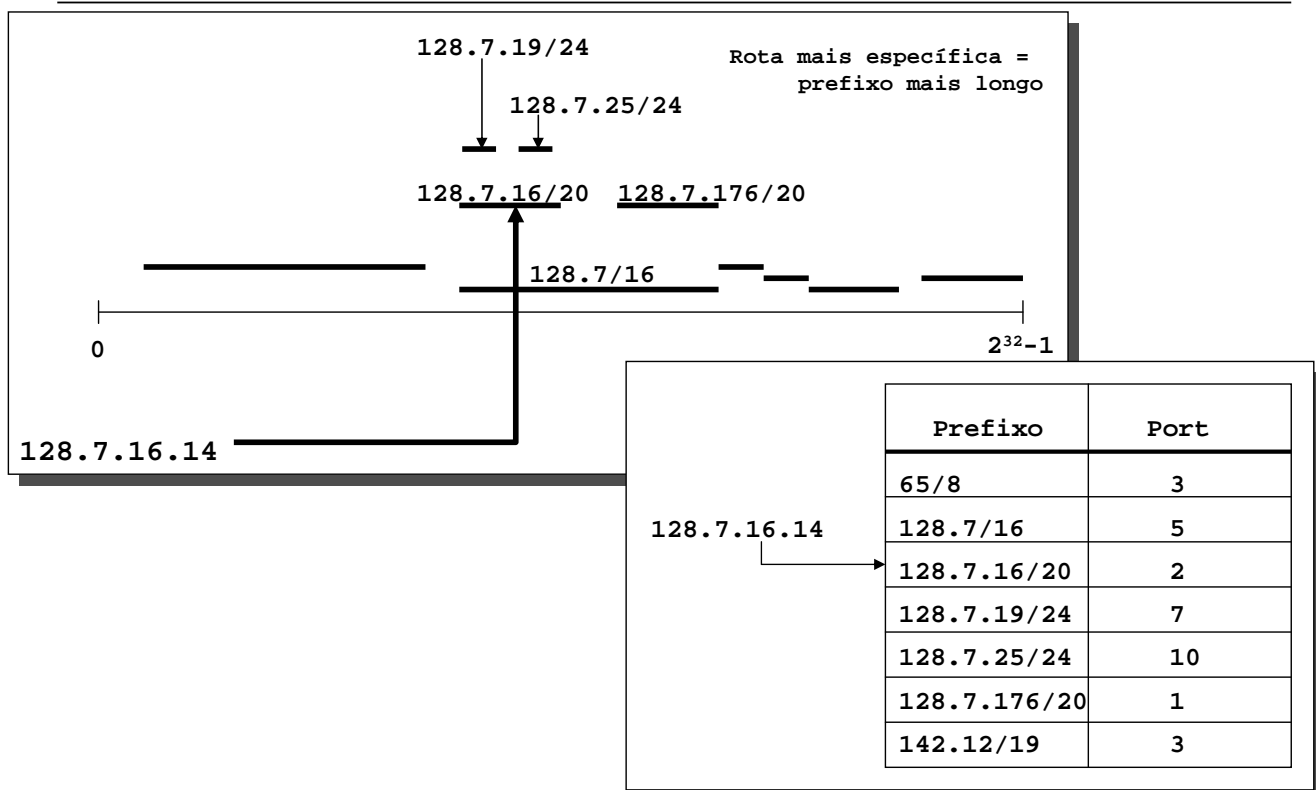
Inspeção da Tabela de Encaminhamento – A função

RT 23

- » Na recepção de um pacote
 - Porta de entrada inspecciona endereço destino do pacote
 - Tabela de encaminhamento no formato
 - ◆ `<networkAddress/mask, port>`
- » Se recebido pacote com endereço A, porta de entrada (conceptualmente)
 - Para cada entrada da tabela encaminhamento
 - ◆ $val = A \& mask^*$ (ex., $mask=8$, $mask^*=255.0.0.0$)
 - ◆ Se ($val == networkAddress \& mask^*$)
 - adiciona porta ao conjunto de portas de saída candidatas
 - Escolhe porta correspondente à maior máscara → rota mais específica
 - Ex.
 - ◆ $tabEnc = \{ \langle 128.32.1.5/16, 1 \rangle, \langle 128.32.225.0/18, 3 \rangle, \langle 128.0.0.0/8, 5 \rangle \}$
 - ◆ Pacote com destino $A=128.32.195.1$
 - ◆ Conjunto de portas de saída candidatas → $\{1, 3, 5\}$.
 - ◆ Porta seleccionada → 3 (maior máscara)

RT 24

CDIR - Classless Inter-Domain Routing



Inspeção da Tabela de Encaminhamento – O Problema

RT 25

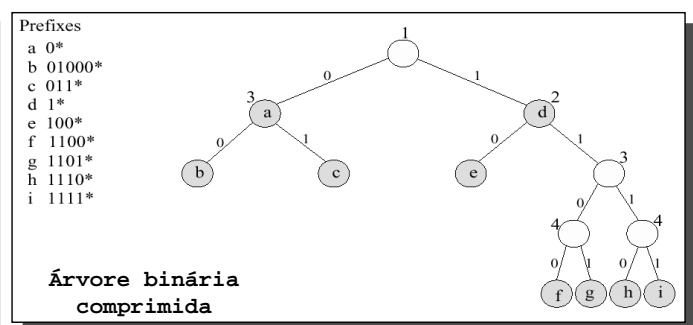
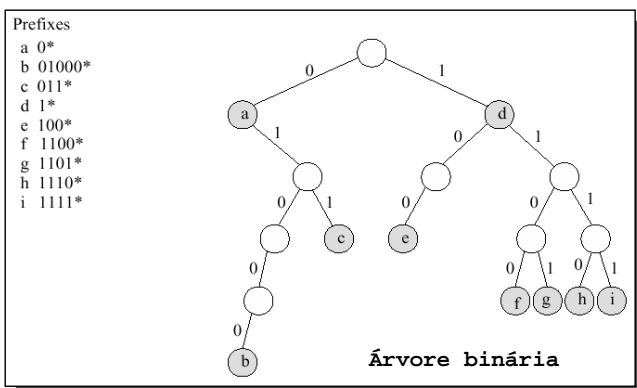
- » Encontrar na tabela de encaminhamento o prefixo mais longo
- » Tempo de procura
 - depende do número e tempos de acesso à memória
 - ◆ Ex. Algoritmo de procura → 8 acesso à memória
 - ◆ $t_{acMem} = 60 \text{ ns}$, $t_{procura} = 480 \text{ ns}$ → $2 * 10^6$ endereços/s
 - ◆ $t_{acMem} = 10 \text{ ns}$, $t_{procura} = 80 \text{ ns}$ → $12.5 * 10^6$ endereços/s

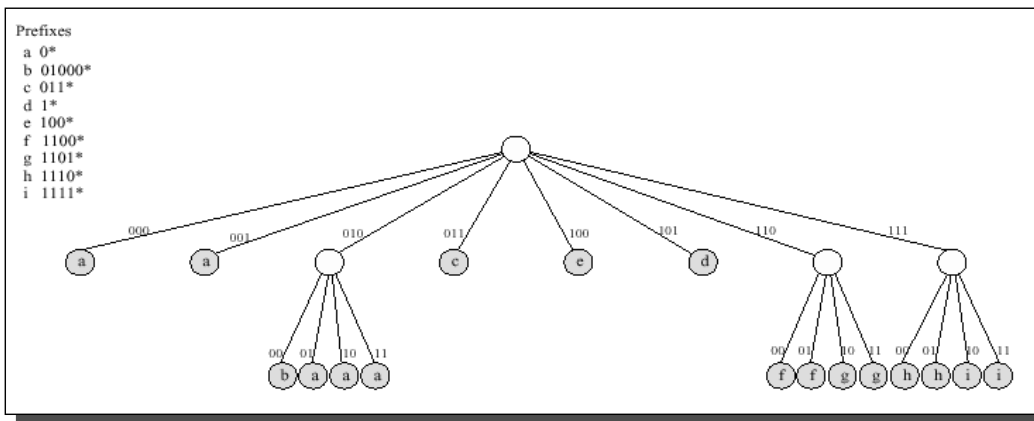
Linha	Débito	Núm Pacotes Comp=40 byte	Núm Pacotes Comp=240 byte
E1	2 Mbit/s	6 kpac/s	1 kpac/s
OC3	155 Mbit/s	480 kpac/s	80 kpac/s
OC12	622 Mbit/s	2 Mpac/s	323 kpac/s
OC48	2.5 Gbit/s	8 Mpac/s	1.3 Mpac/s
OC192	10 Gbit/s	31 Mpac/s	5 Mpac/s

RT 26

Técnica de Procura Clássica

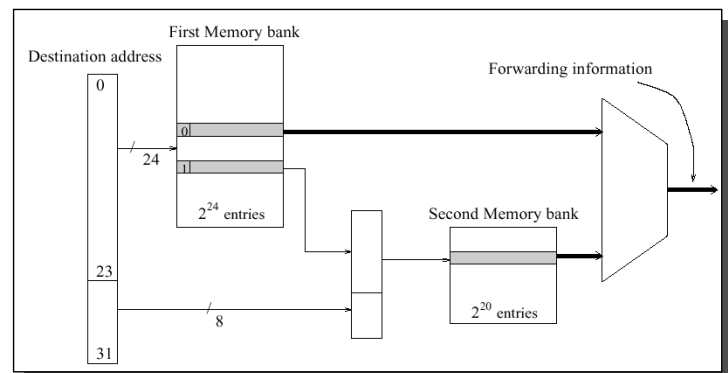
- » Entradas armazenadas em forma de árvore
 - Cada percurso, da raiz à folha → 1 entrada na tabela
 - Prefixo mais longo == percurso mais longo que satisfaz endereço destino pacote
 - Usado nas implementações UNIX BSD
- » Características
 - Minimização de espaço de memória usada. Necessário navegar na árvore
 - Preço memória baixa → aproximação errada para os routers rápidos





Desempenho melhorado por

- Técnicas de hardware
- Tabelas de hash
- Compactação de tabelas

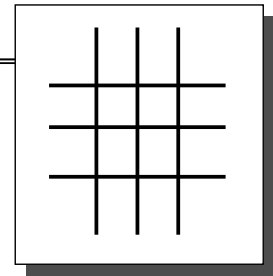


Matrizes de Comutação

- ◆ Década de 80
 - Descoberta dos conceitos de comutação de pacotes mais comuns (redes de Banyan)
 - Não usados → débitos dos barramentos suficientes
- ◆ Década de 90
 - » Aplicação dos conceitos à construção de matrizes de células ATM
 - Pacotes pequenos. Identificador de circuito virtual
- ◆ Actualmente, routers usam técnicas de comutação semelhantes ao ATM
 - » Circuitos virtuais permanentes estabelecidos entre portas de entrada e saída
 - » Na porta de entrada, depois de determinada porta de saída
 - Pacote fragmentado em células
 - Células comutadas e reassembladas na porta de saída. Antes da transmissão como pacote
 - » Vantagens da solução
 - Suporte QoS simplificado
 - Problemas de comutação de pacotes de comprimento variável evitado
 - » Dificuldades de implementação de
 - multicast
 - algoritmos de controlo de tráfego IP ← orientados a pacotes

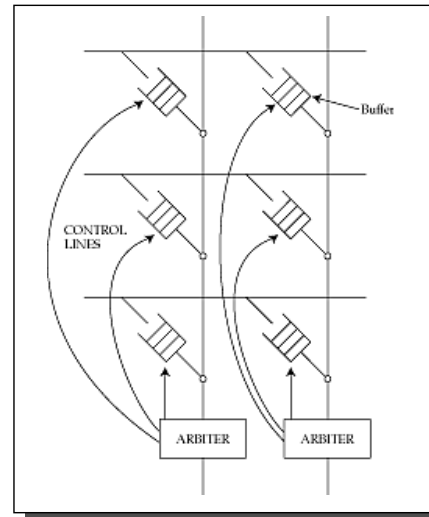
Matriz de Comutação

» Modelo simplificado → 2N barramentos em paralelo

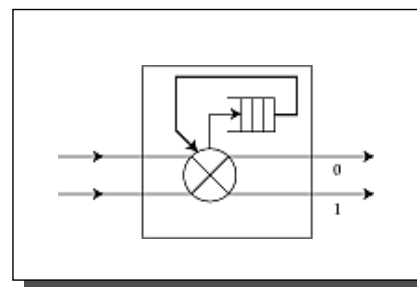


» Na comutação de pacotes
– Ponto de contacto aberto o tempo suficiente para transferir pacote da entrada para a saída

» Pode ser conter buffers internos



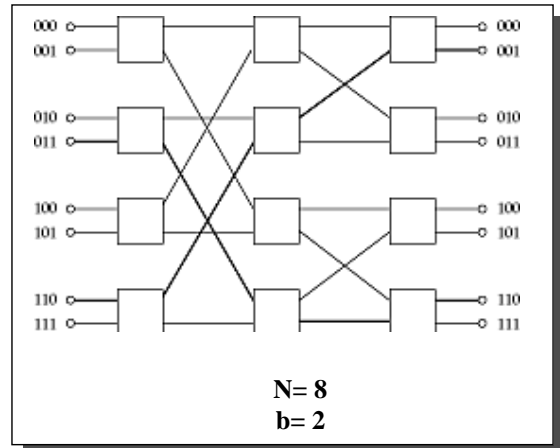
Matriz Elementar



- ♦ Encaminhamento
 - » Se 0 → pacote enviado pela linha superior
 - » Se 1 → pacote enviado pela linha inferior
- ♦ Se 2 pacotes para mesma saída
 - » Bufferiza ou elimina pacote

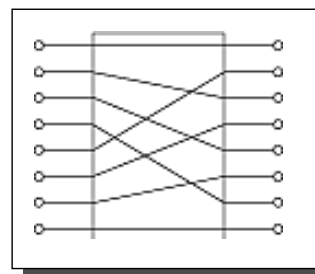
Matriz Banyan

- ◆ Matriz $N \times N$ composta elementos $b \times b$
 - » $\log_b N$ andares
 - » N/b elementos por andar
- ◆ Matriz
 - Encaminhamento automático
 - Síncrona ou assíncrona
 - Regular → implementação fácil em VLSI
- ◆ Em qualquer porta de entrada
 - Pacote com endereço x é entregue na saída x !
- ◆ Se dois pacotes endereçados para x
 - Situação de bloqueio



Matriz não Bloqueante

- ◆ Bloqueio pode ser evitado → escolha da ordem de apresentação
- ◆ Para isso
 - » Ordenar pacotes
 - » Remover duplicados
 - » Remover linhas vazias
 - » Baralhar entradas



◆ Exemplo

```
[x, 011, 010, x, 011, x, x, x] - (ordenar) →
[010, 011, 011, x, x, x, x, x] - (remover duplicados) →
[010, 011, x, x, x, x, x, x] - (baralhar) →
[010, x, 011, x, x, x, x, x]
```


Filas de Saída – Como Melhorar a Velocidade

RT 33

- ◆ Problema das filas de saída → tempo de acesso às filas

- ◆ 2 técnicas para resolver o problema
 - » Construção de memórias da largura de uma célula
 - Memórias em paralelo, alimentadas por barramento dados da largura da célula
 - Escrita num ciclo de memória
(Preço da memória continua a descer 60% ao ano)

 - » Integração dos buffers + controlador da porta num único chip

Comutador com Fila à Entrada

RT 34

- ◆ Problema das filas à entrada → bloqueio do primeiro pacote
 - » Se primeiro pacote está bloqueado (porta de saída ocupada) →
 - Pacotes seguintes da fila não podem ser enviados (mesmo que suas portas estejam livres)

- ◆ Solução → manter em cada entrada uma fila para cada saída

- ◆ Outros problemas
 - » Escalonamento de pacotes para suporte de QoS → feito sobre fila de saída
 - Adaptação não trivial
 - Escalonamento depende → tipo do pacote, tipo de nível 2 da linha de saída
 - » Controlo de congestão (ex. Random Early Discard)
 - Actua directamente sobre fila de saída

- ◆ Routers de empresa → variedade de portas de saída e políticas
 - » Convivem dificilmente com filas à entrada
 - » Utilizam soluções híbridas