



Políticas de Filas

Fila FCFS ou FIFO

- Serve pacotes na ordem de chegada e descarta quando fila está cheia
- Não discrimina pacotes
- O atraso médio de uma fila FIFO é usado para comparação com outras disciplinas

Disciplinas

- Conservativa
 - Servidor sempre ocupado enquanto há fila de espera
- Não conservativa
 - Servidor pode ficar ocioso enquanto há fila de espera
- Uma disciplina não conservativa pode gerar atraso médio maior que FCFS
- Lei da Conservação para Disciplinas Conservativas

Lei da Conservação para Sistemas Conservativos

■ Premissas

- Assuma N conexões compartilhando enlace
- Conexão i
 - ☞ taxa média λ_i , serviço médio x_i
 - ☞ Utilização do enlace pela conexão i: $\rho_i = \lambda_i x_i$
 - ☞ Tempo de espera para a conexão i = w_i

■ Então,
$$\sum_{i=1}^N \rho_i w_i = \text{constante}$$

- Ou seja, ao alterar a disciplina, uma conexão só é favorecida (comparando com FCFS) às custas de uma outra conexão

Exemplo: Lei da Conservação

- Circuitos virtuais A e B, com taxas de 10 e 25 Mbps, compartilham enlace de 155 Mbps.
- Com disciplina FCFS, os atrasos nestes dois circuitos virtuais são de 0,5 ms.
- Se adotamos uma nova disciplina e o atraso de A é reduzido para 0,1 ms, qual será o novo atraso de B?

Solução

- A utilização no enlace será proporcional às taxas de operação
 - $\rho_a = 10/155, \rho_b = 25/155$
- Pela Lei da Conservação
 - $10/155*0,5+25/155*0,5 = 10/155*0,1 + 25/155*w_b$
- Então, $w_b = 0,66$ ms (maior que valor anterior de 0,5 ms)

Eqüidade ou Fairness

- Princípios
 - Recursos são alocados em ordem crescente de demanda
 - Nenhuma fonte obtém mais do que solicitou
- Algoritmo Max-Min para eqüidade
 - Demandas $x_1 < x_2 \dots < x_n$ de igual prioridade e capacidade do servidor igual a C
 - Aloque inicialmente C/n igualmente a todas as demandas
 - Interação
 - ☞ Se $(C/n - x_1) > 0$, aloque mais $(C/n - x_1)/(n-1)$ às outras demandas;
 - ☞ Repita para x_2 e sucessivamente até x_n .
 - Processo termina quando cada fonte obtém não mais do que pediu, e, se sua demanda não foi satisfeita, não menos do que qualquer fonte com demanda maior que a sua.

Exemplo: Max- Min

- 4 fontes com demandas = {2; 2,6; 4; 5}, compartilhando capacidade de 10
- Inicialmente, alocamos (2,5) a todas as fontes
- f1 necessita de 2, sobra (0,5), que é distribuído às outras fontes, dando mais (0,166) a cada
- f2 recebe (2,666), necessita de apenas (2,6) então sobra (0,066) que é dividido entre f3 e f4
- f3 e f4 recebem então (2,666+0,033=2,7) cada
- No final, f1 ganha 2, f2 ganha (2,6), f3 e f4 ganham (2,7)

Max-Min Weighted

- Fontes de 1 a n com pesos associados w_1, w_2, \dots, w_n que refletem a fatia relativa a ser alocada a cada fonte
- Recursos são alocados em ordem crescente de demanda, normalizados pelo peso
- Nenhuma fonte recebe mais do que pediu
- Fontes com demandas insatisfeitas obtém recursos na proporção de seus pesos

Exemplo: Max-Min Weighted

- 4 fontes com demandas = {4; 2; 10; 4} e pesos de {2,5; 4; 0,5; 1}, compartilhando capacidade de 16
- Normalize os pesos de modo que o menor seja 1
 - Pesos = {5; 8; 1; 2}
 - Tudo se passa como se o # de fontes fosse 16
- Num primeiro passo, cada fonte recebe proporcionalmente ao peso
 - f1 ganha 5; f2 ganha 8; f3 ganha 1; f4 ganha 2
 - f1 e f2 recebem a mais do que precisam, sobrando 7
- Reparte 7 entre f3 e f4 proporcionalmente aos pesos
 - f3 ganha $1 + 7 \cdot (1/3)$; f4 ganha $2 + 7 \cdot (2/3) = 6 + 2/3$
 - f4 recebe a mais, sobrando $2 + 2/3$
- f3 recebe então $1 + 7/3 + 2 + 2/3 = 6$
- No final, f1 recebe 4, f2 recebe 2, f3 recebe 6 e f4 recebe 4

Escalonando Best Effort

- Várias filas de espera são servidas por um único servidor (subfila)
- Round-Robin (RR)
 - Serve um pacote de cada fila não vazia por ciclo
- Weighted RR (WRR)
 - Serve uma conexão em proporção ao seu peso

Exemplo: WRR com pacotes de mesmo tamanho

- Conexões A, B e C têm mesmo tamanho de pacote e pesos (0,5), (0,75) e 1. Quantos pacotes de cada conexão são servidos por ciclo WRR?
- Solução
 - Normalize os pesos para inteiro: 2, 3 e 4
 - Serve 2 de A, 3 de B e 4 de C

Exemplo: WRR com pacotes de tamanhos diferentes

- Conexões A, B e C com tamanhos de 50, 500 e 1500 bytes e pesos de (0,5), (0,75) e 1. Quantos pacotes são servidos por conexão?
- Solução
 - Normalize: divida o peso pelo tamanho dos pacotes
 - ☞ (0,5/50), (0,75/500), (1/1500) ou (0,01), (0,0015) e (0,00066)
 - Obtendo pesos normalizados inteiros: 60, 9 e 4
 - Serve 60 de A, 9 de B e 4 de C (em pacotes)
 - ☞ Por ciclo, temos 3000 bytes de A, 4500 bytes de B e 6000 bytes de C, que está de acordo com os pesos

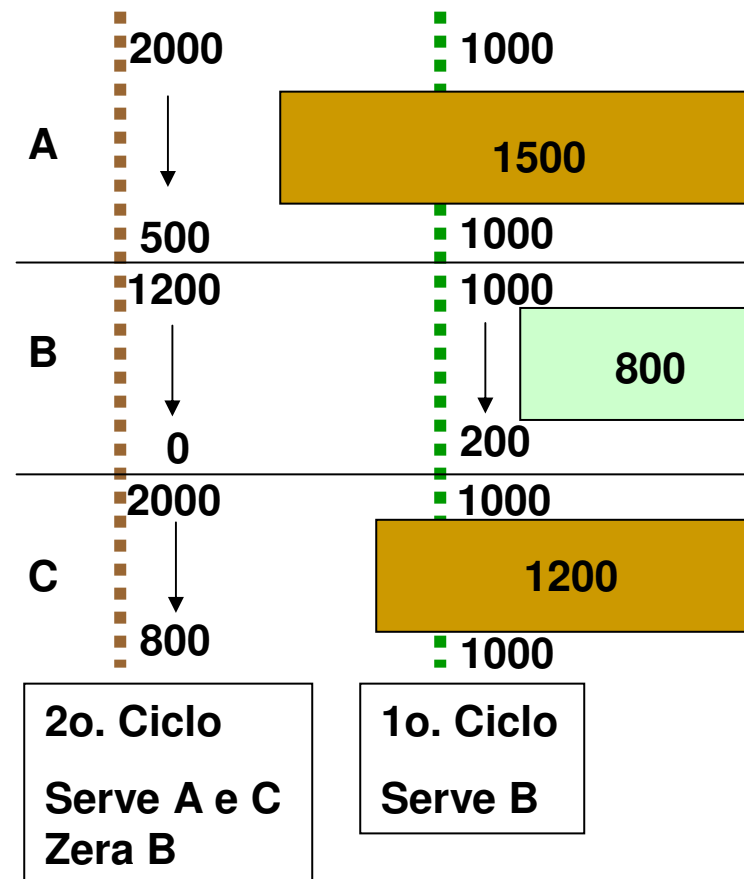
Exemplo: WRR com pacotes de tamanhos diferentes

■ Problemas

- Como saber o tamanho de pacote com antecedência?
- Iguatário somente se o tempo \gg ciclo
 - ▣ Para conexões com peso muito pequeno ou grande número de conexões, períodos longos de falta de equidade poderão ocorrer
- Bom para ATM que tem tamanho de célula pequeno e fixo

Deficit RR ou DRR

- Existe um tamanho de quantum a ser servido (no exemplo 1000)
- Contadores de déficit por conexão



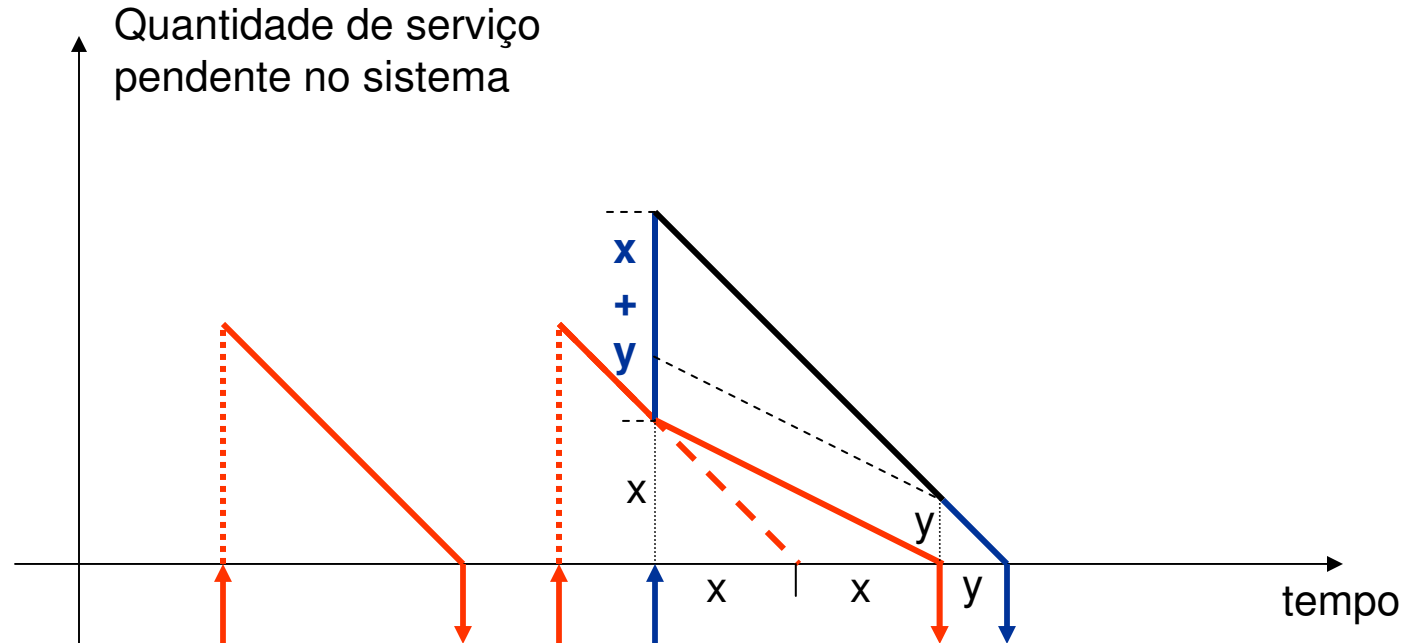
Weighted DRR

- Serve o tamanho do quantum x peso para a conexão
- Pode não ser equânime em intervalos de tempo pequenos em relação ao tamanho do ciclo (depende do número de conexões)
- Para pacotes pequenos é adequado

Generalized Processor Sharing - GPS

- Uma subfila para cada fluxo
- Para tráfego best effort e classes de fluxo com mesmo peso, a disciplina ideal é aquela que provê eqüidade como descrito no algoritmo max-min
- GPS é uma disciplina ideal que atinge este objetivo
 - Serve todas as subfilas ativas em round robin, dedicando a cada uma delas uma fatia de tempo infinitesimal
 - Se um peso é dedicado a cada fluxo, GPS serve proporcionalmente ao peso e provê weighted max-min fair share

Exemplo de GPS



- Quando somente um pacote está no sistema, ele recebe um segundo de serviço a cada segundo de tempo
- Quando dois pacotes estão no sistema, cada um recebe apenas meio segundo por segundo de serviço
 - Para receber x é preciso passar $2x$ de tempo

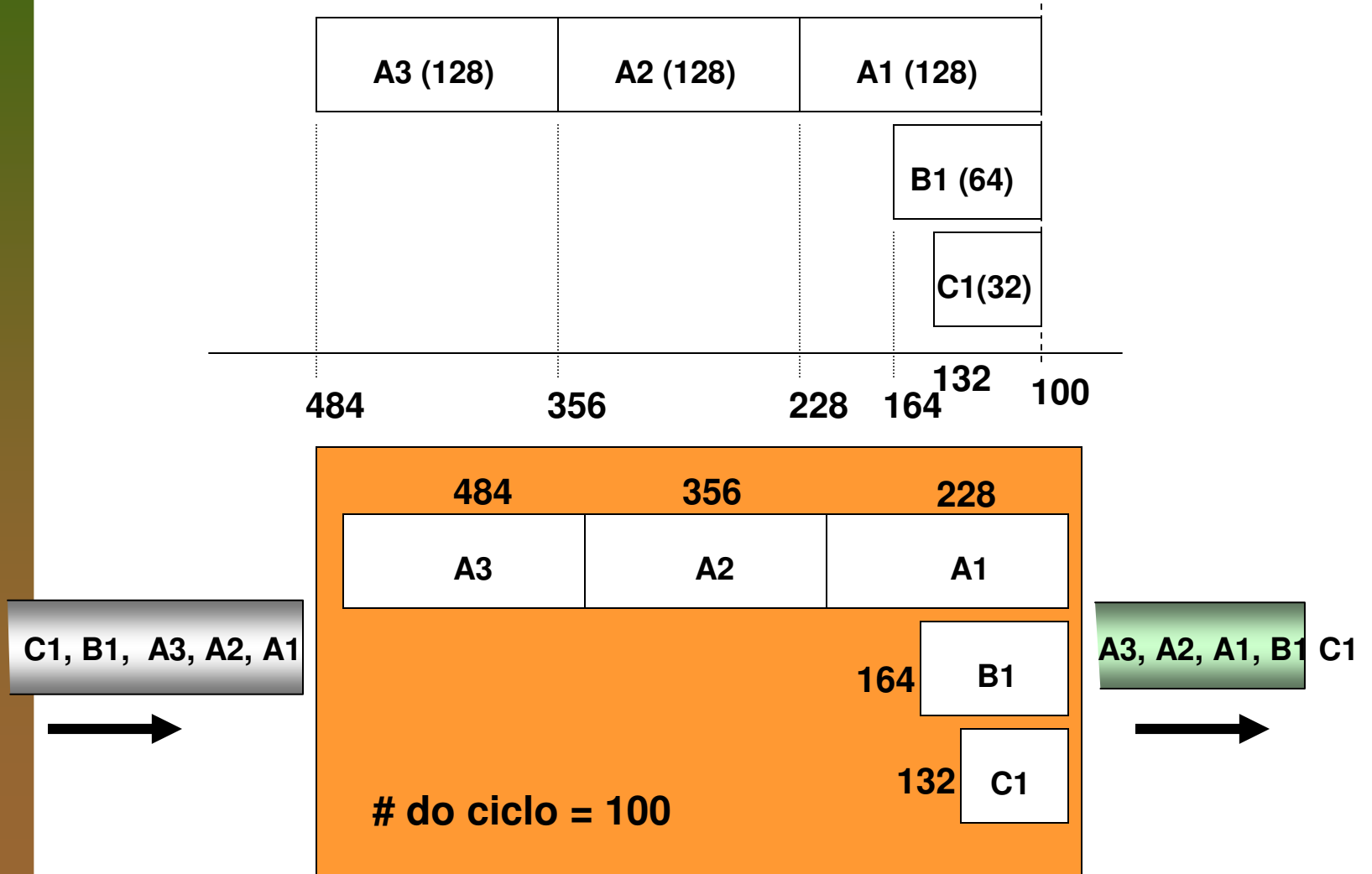
Fair Queuing (FQ)

- Fair Queuing é a disciplina real que se aproxima de GPS quando todos os fluxos tem mesmo peso ou tratamento
 - Procura calcular o instante em que um pacote completaria o serviço se estivesse sendo servido por GPS e então serve os pacotes na ordem desses tempos de término
 - Opera orientado a byte
- Uma fila FQ envolve múltiplas subfilas

Fair Queuing

- Pacote recebe um número **SN** para ordem de serviço
- Mantemos uma variável **# do ciclo** que representa quantos ciclos servindo um byte/ciclo o escalonador já completou
- Pacotes que chegam a uma subfila vazia
 - **SN** = tamanho do pacote em bytes + # do ciclo
- Pacotes que chegam a uma subfila ocupada
 - **SN** = tamanho do pacote em bytes + SN do último pacote enfileirado
 - Subfilas são servidas FIFO
- O número do ciclo é sempre atualizado para o **SN** do último pacote selecionado da FQ

Exemplo: FQ



WFQ – Weighted Fair Queuing

- Fluxo no WFQ
 - Pacotes são classificados pela inspeção do TOS **excetuando os três de precedência IP**, número da porta de fonte e destino, endereços IP de fonte e destino, etc
- Escolha da fila no WFQ
 - Hash do fluxo para encaminhar para uma subfila
 - ☞ Pacotes de um mesmo fluxo mas com precedência IP diferentes ainda são servidos FIFO na mesma subfila
 - Se # de fluxos > # de filas, fluxos diferentes irão compartilhar mesma subfila
 - ☞ Cisco: # de filas máximo (default) = 256
- Cada fila tem prioridade ou peso associado a ela
 - Tamanho do pacote multiplicado pelo peso para cálculo do SN (operação idêntica a FQ no resto)

WFQ – Weighted Fair Queuing

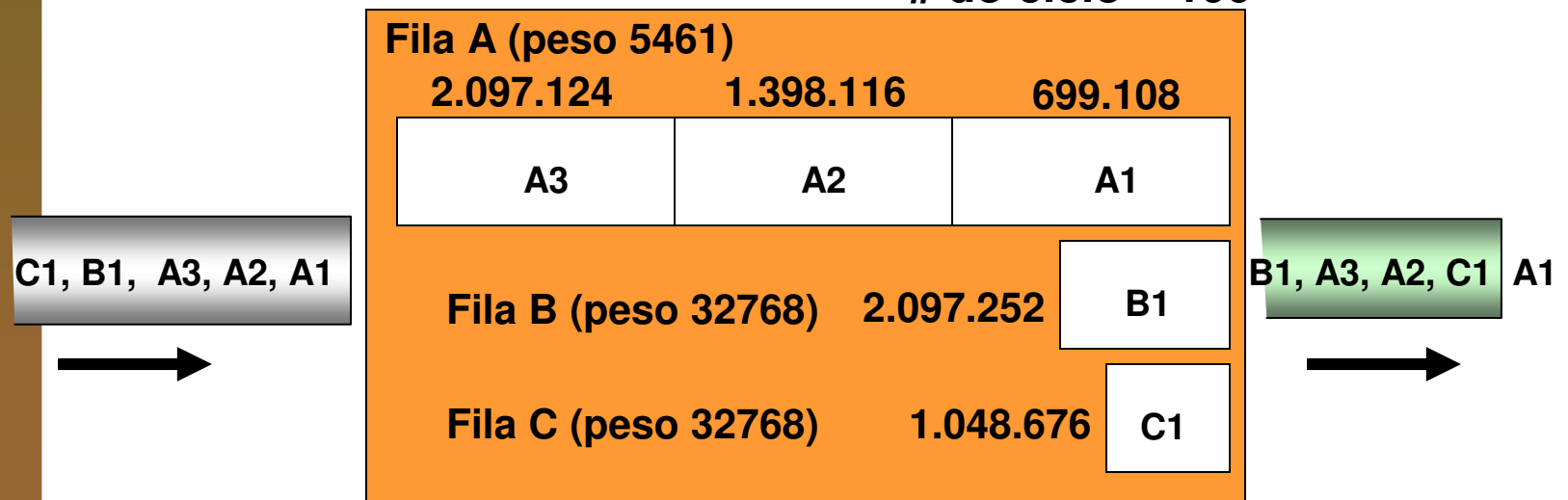
- Para IOS 12.0(5)T e acima
 - $\text{Peso} = 32768 / (\text{precedência IP} + 1)$
 - ☞ ou seja, o peso será tanto maior quanto menos prioritário for o pacote
- Para versões anteriores de IOS
 - $\text{Peso} = 4096 / (\text{precedência IP} + 1)$

Exemplo: WFQ

- Exemplo anterior supondo A com precedência 5, B e C com precedência 0

	Tamanho/pct	Precedência	Peso
— Fila A	128 bytes	5	5461
— Fila B	64 bytes	0	32768
— Fila C	32 bytes	0	32768

do ciclo = 100



WFQ

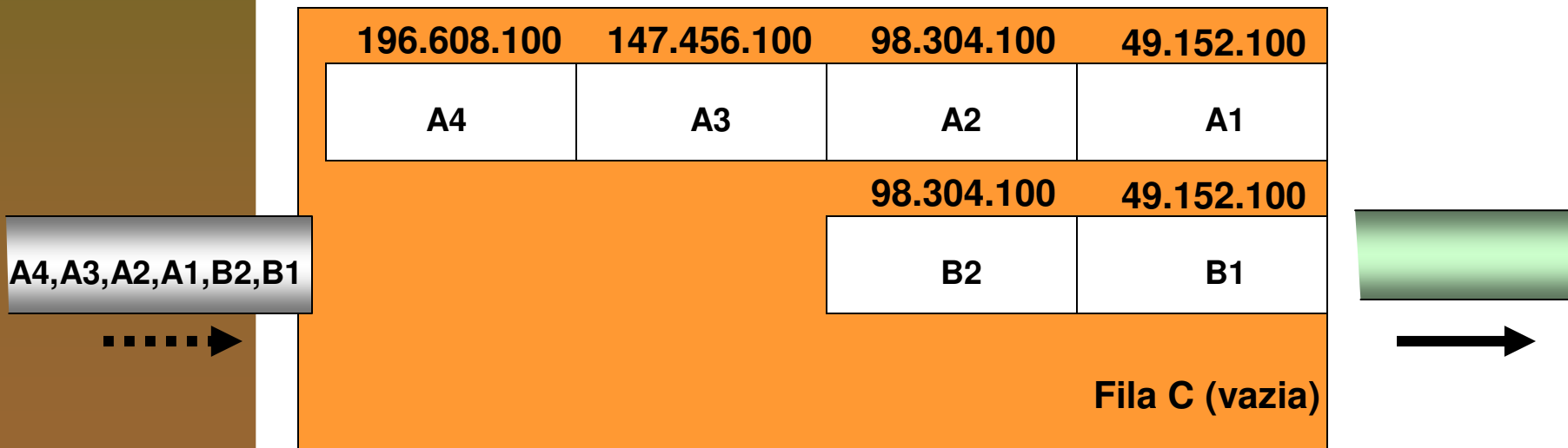
- No caso # de fluxos < # de filas, cada fila será alocada a um único fluxo
- Fluxos intensos interferirão muito pouco com os fluxos com pouca intensidade de tráfego, onde pacotes encontram quase sempre a fila correspondente vazia
 - $SN = \text{ciclo atual} + \text{tamanho em bytes} * \text{peso}$
- Precedência IP pode não ser necessária nestes casos, ou ter nenhum efeito

Ex.: fluxos intensos c/ precedência IP

- A e B com precedência 0, chegam em rajada

	Tamanho	Precedência	Peso
— Fila A	1500 bytes	0	32768
— Fila B	1500 bytes	0	32768
— Fila C	128 bytes	5	5461

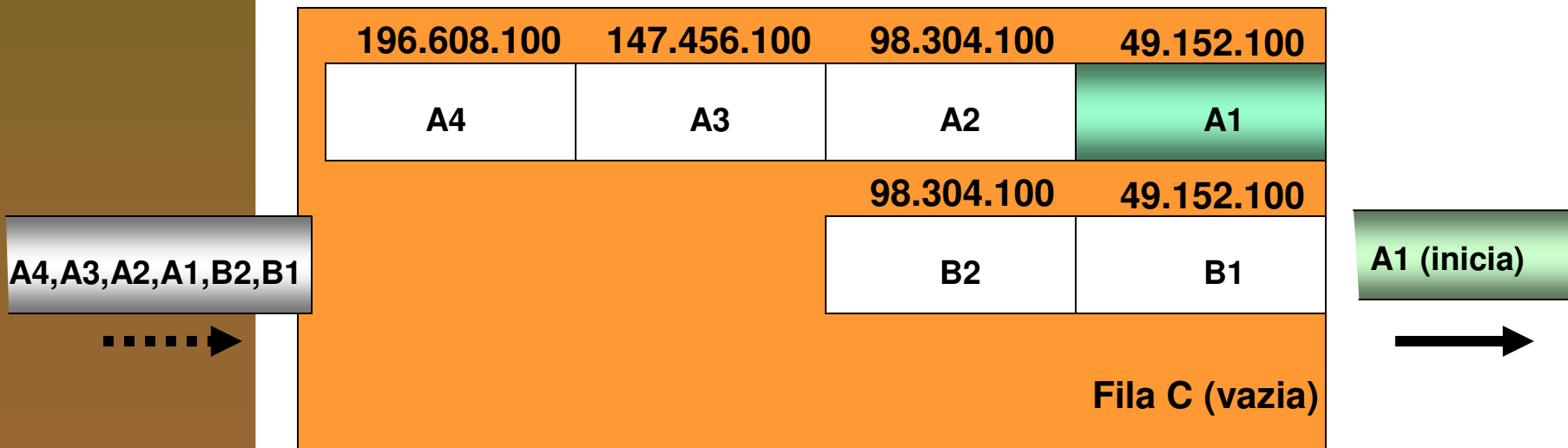
do ciclo = 100



Ex.: fluxos intensos c/ precedência IP

- A1 é selecionado para tx e SN = 49.152.100
- O pacote A1 é possivelmente enviado para buffer na interface física para tx

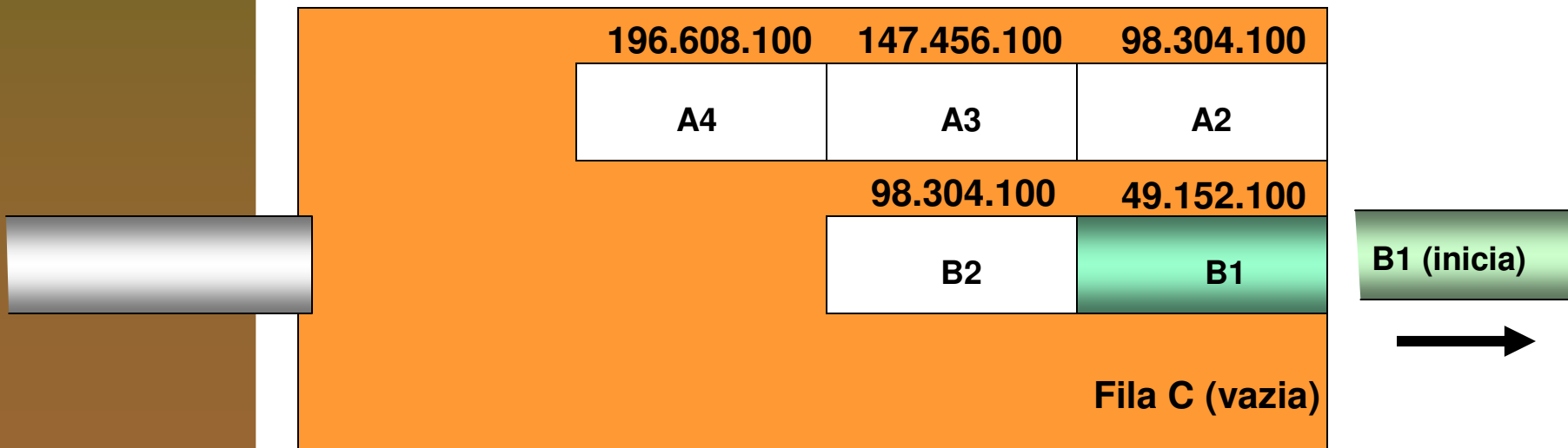
do ciclo = 49.152.100



Ex.: fluxos intensos c/ precedência IP

- A1 termina e B1 é selecionado e enviado para tx
 - B1 irá para o buffer da interface física
 - SN=49.152.100

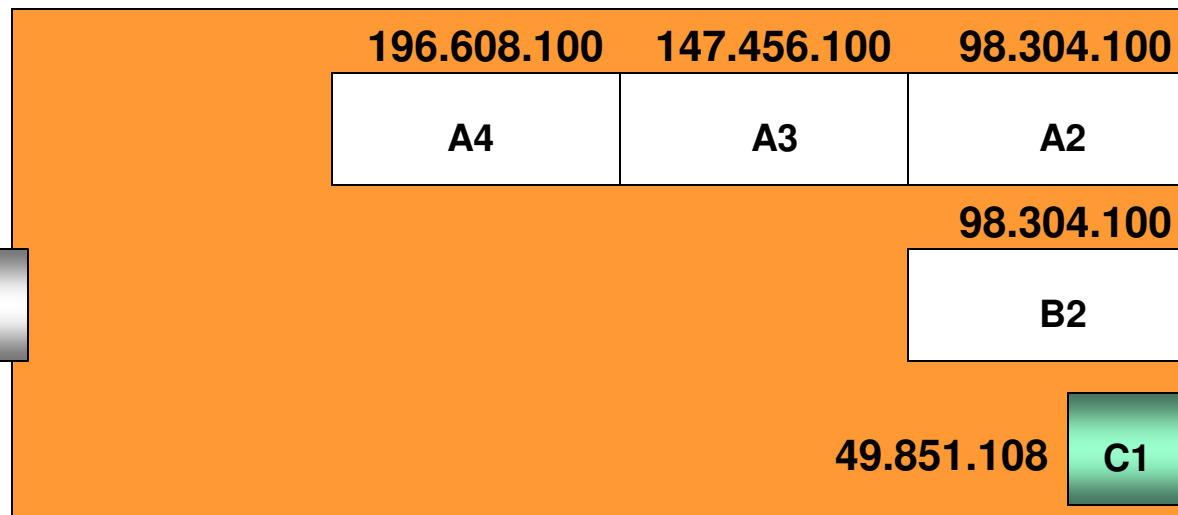
do ciclo = 49.152.100



Ex.: fluxos intensos c/ precedência IP

- C1 inicia tx, após a tx do pacote B1 acontecer (espera pela transmissão de 1100 bytes)

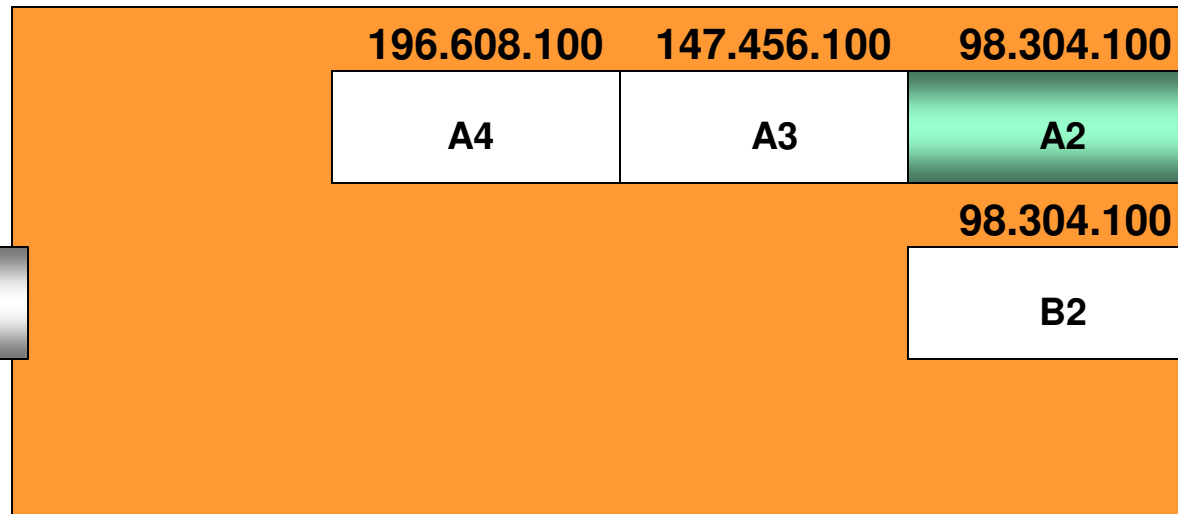
do ciclo = 49.851.108



Ex.: fluxos intensos c/ precedência IP

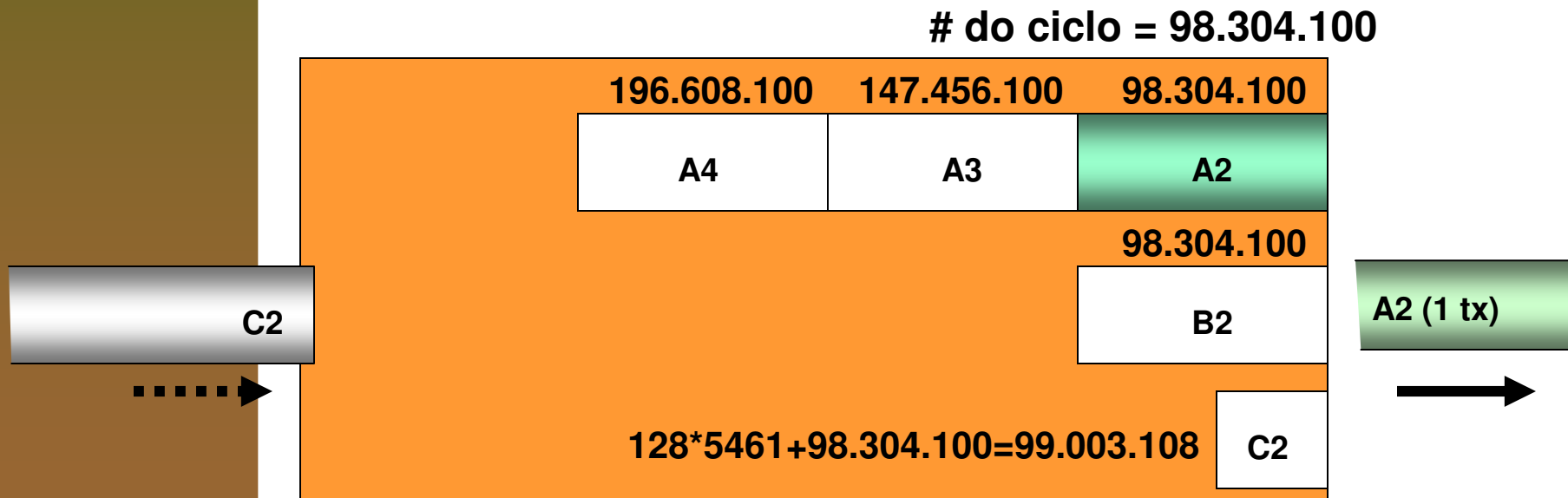
- A2 inicia tx e SN vai para 98.304.100

do ciclo = 98.304.100



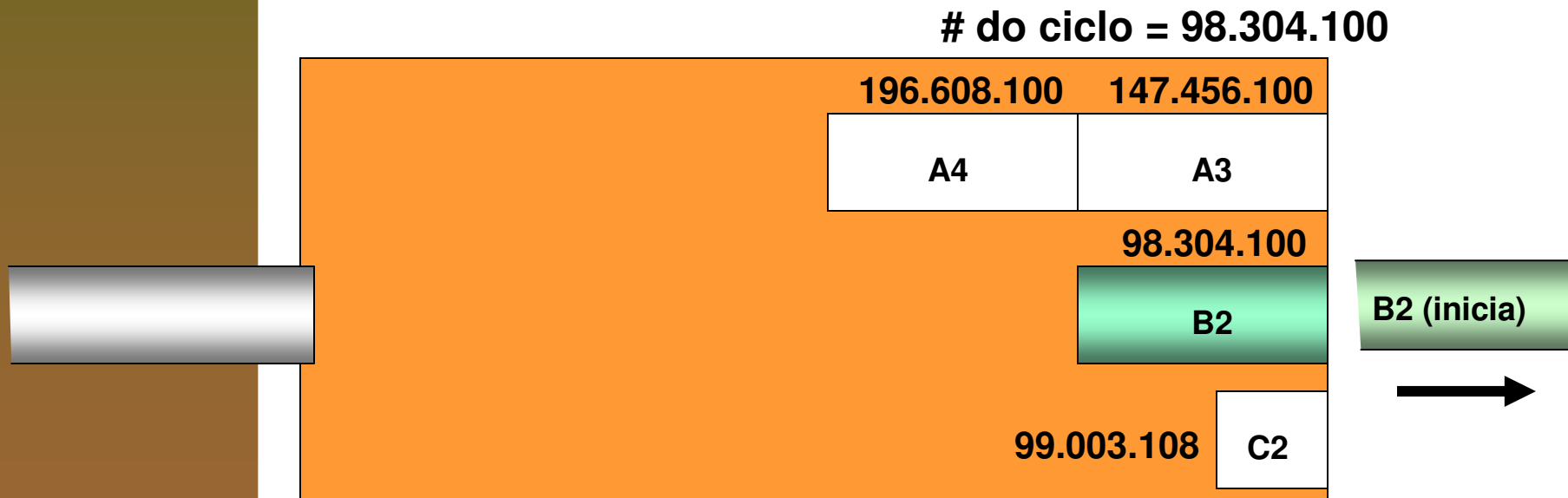
Ex.: fluxos intensos c/ precedência IP

- Imediatamente após, C2 chega com precedência 5
 - Transmissões em andamento não são abortadas



Ex.: fluxos intensos c/ precedência IP

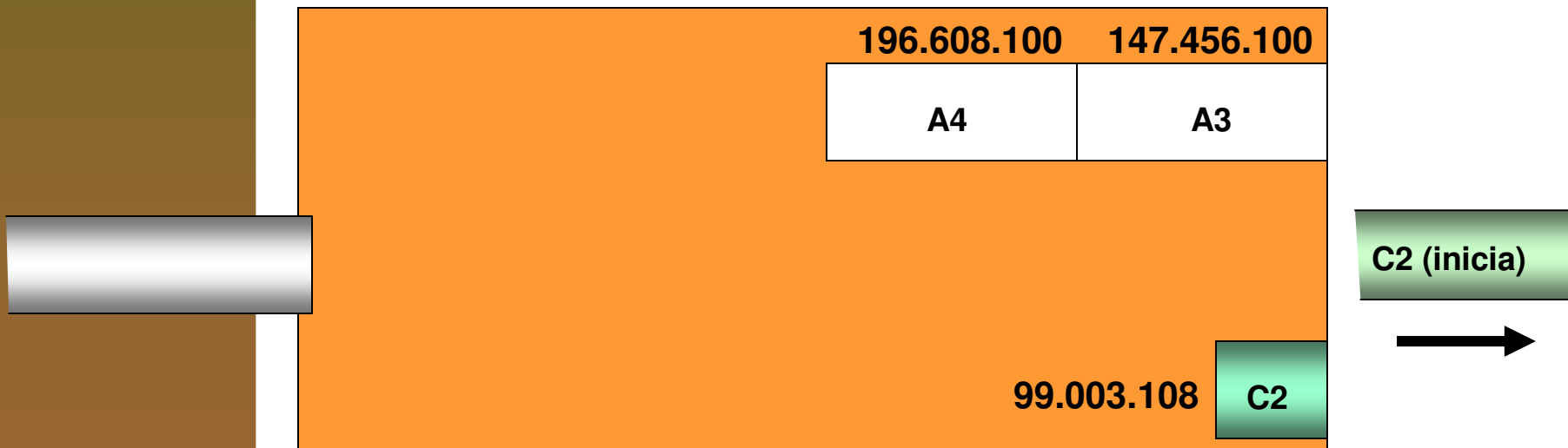
- B2 será transmitido antes de C2, mantendo o SN = 98.304.100



Ex.: fluxos intensos c/ precedência IP

- C2 somente será selecionado após B2
- Tempo de espera depende do tempo residual da transmissão em andamento na chegada e dos pacotes já presentes na fila
 - No exemplo, o pacote C2 esperou pela transmissão de dois pacotes grandes

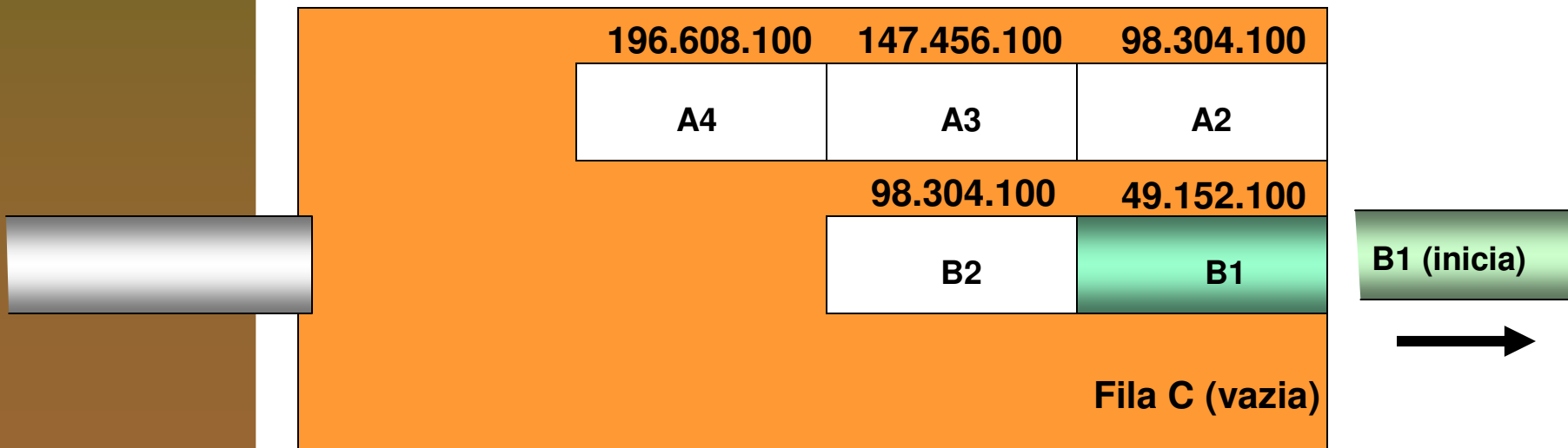
do ciclo = 99.003.108



Ex.: fluxos intensos sem uso de precedência IP

- A1 termina e B1 inicia transmissão

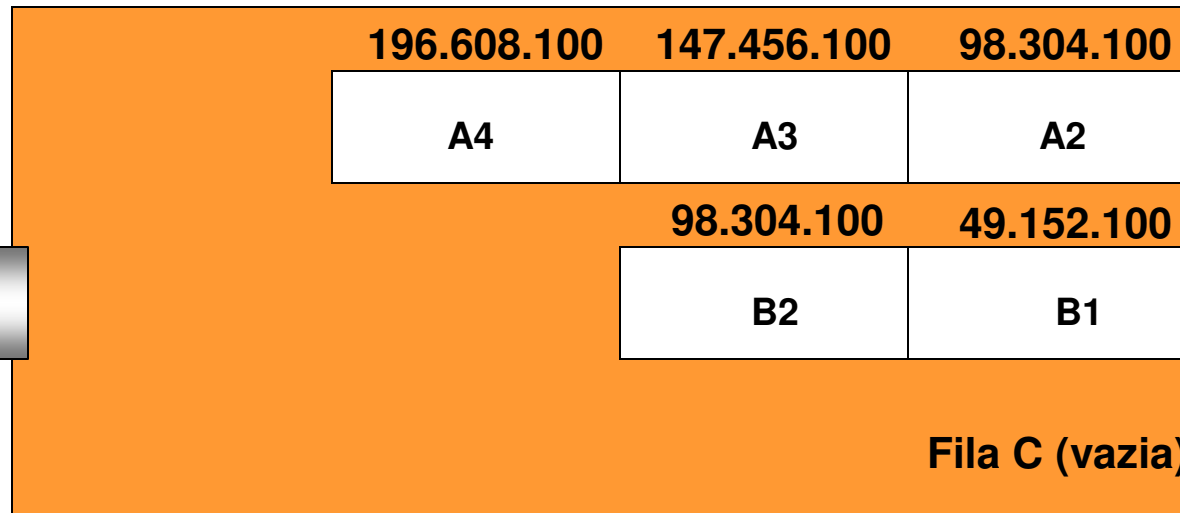
do ciclo = 49.152.100



Ex.: fluxos intensos sem uso de precedência IP (antigo)

- A1 termina e B1 inicia transmissão

do ciclo = 1600



B1 (inicia)

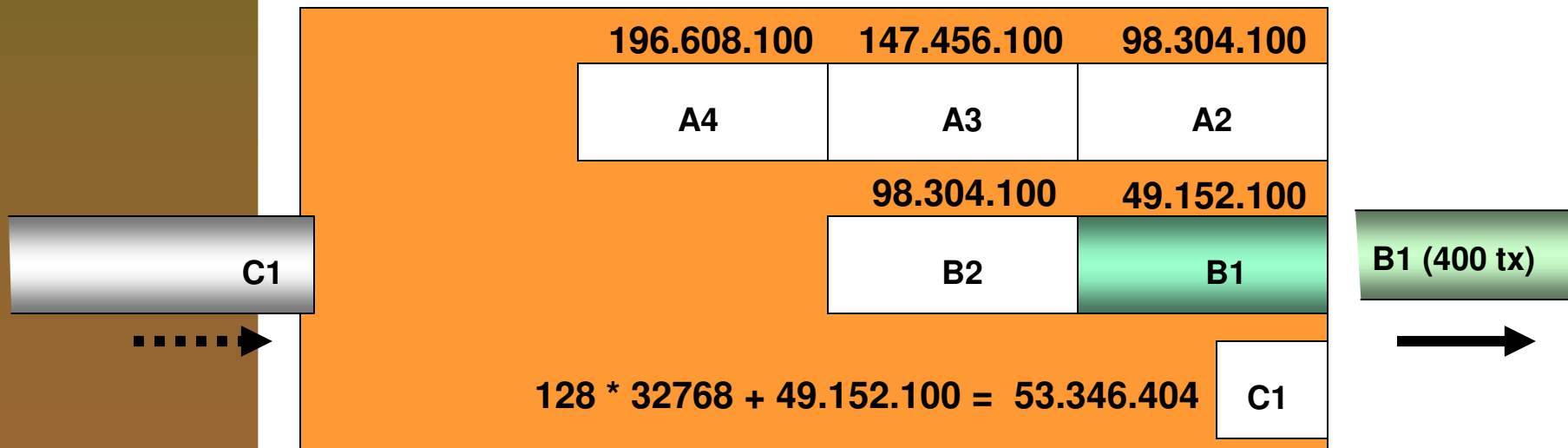


Ex.: fluxos intensos sem uso de precedência IP

- B1 em andamento, quando chega C1

	Tamanho	Precedência	Peso
— Fila A	1500 bytes	0	32768
— Fila B	1500 bytes	0	32768
— Fila C	128 bytes	0	32768

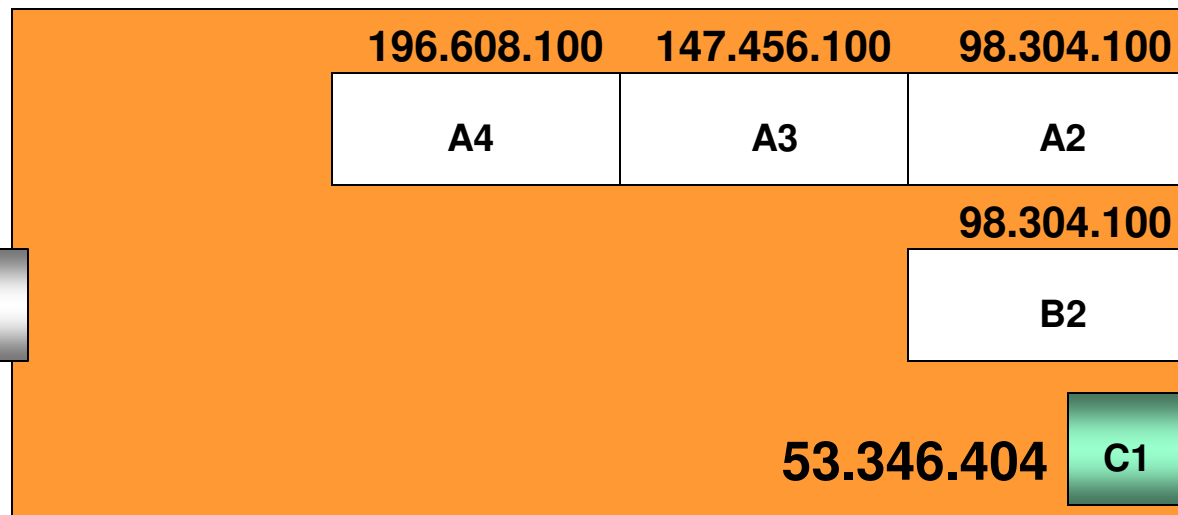
do ciclo = 49.152.100



Ex.: fluxos intensos sem uso de precedência IP

- C1 inicia transmissão após B1, mesmo comportamento que tivemos com o uso de precedência IP

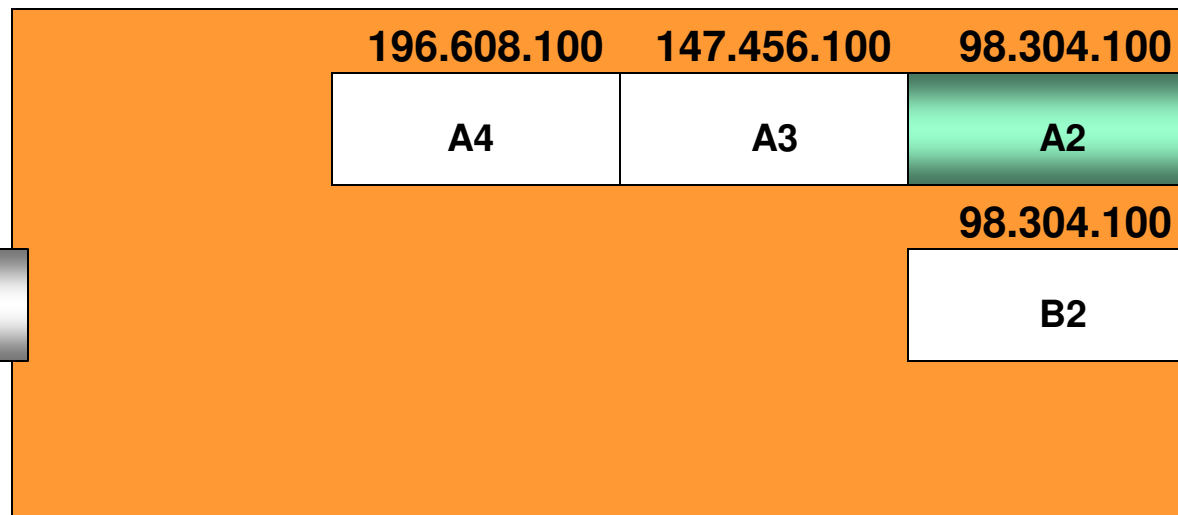
do ciclo = 53.346.404



Ex.: fluxos intensos sem uso de precedência IP

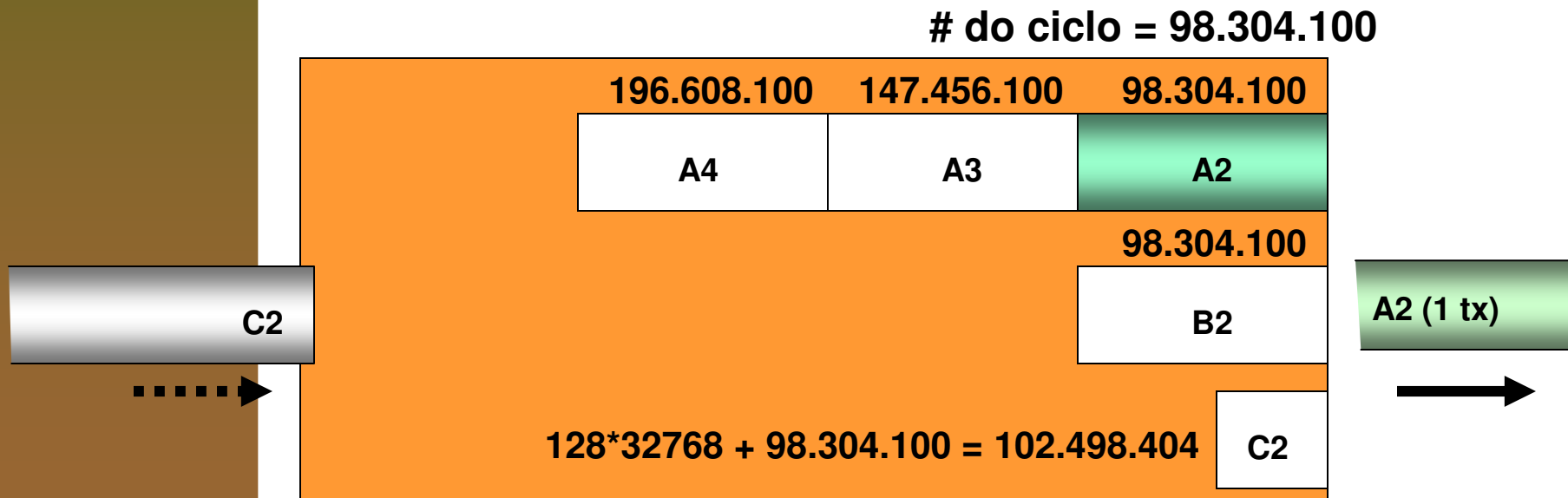
- A2 é o próximo selecionado para transmissão ficando SN = 98.304.100

do ciclo = 98.304.100



Ex.: fluxos intensos sem uso de precedência IP

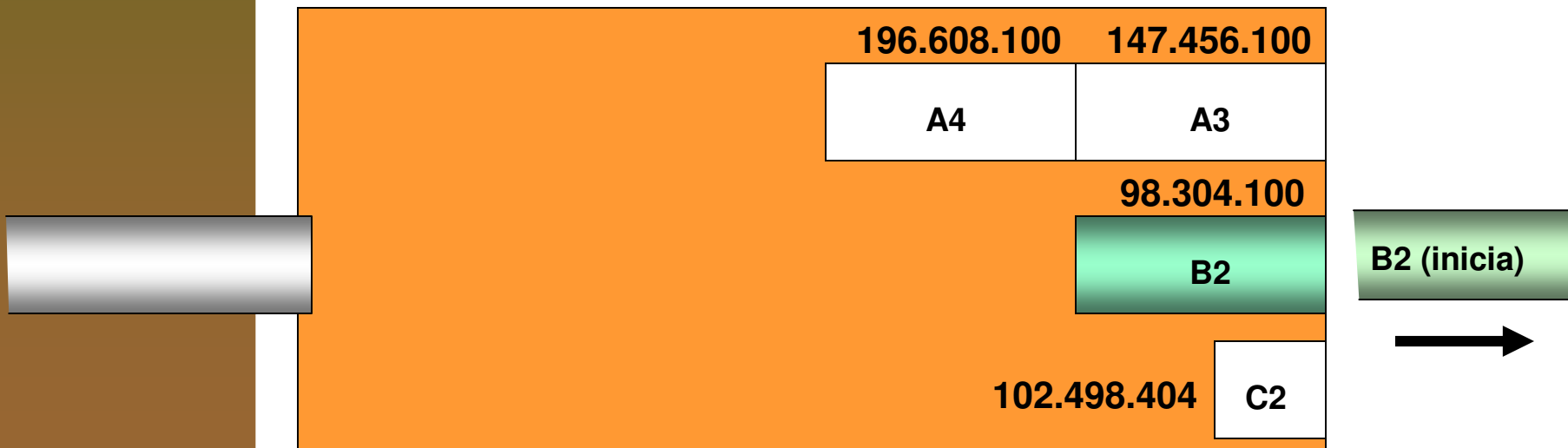
- Imediatamente após, C2 chega com precedência 0
 - Transmissões em andamento não são abortadas



Ex.: fluxos intensos sem uso de precedência IP

- B2 é o próximo e SN = 98.304.100

do ciclo = 98.304.100



Ex.: fluxos intensos sem uso de precedência IP

- C2 é servido após B2 e SN = 102.498.404
- A seqüência em que os pacotes são servidos e o tempo de espera de C2 é o mesmo quando tínhamos precedência

do ciclo = 102.498.404

196.608.100 147.456.100

A4

A3

102.498.404

C2

C2 (inicia)



WFQ

- **Atenção!**
 - Pacotes que chegam a uma fila WFQ vazia de um roteador são encaminhados para o buffer de transmissão da interface (camada 2)
- Como o buffer de camada 2 é FIFO, o atraso num roteador dependerá também do tamanho e utilização dessa file FIFO

WFQ

- Ainda que favoreça os pacotes de menor tamanho (que é o caso de VoIP, por exemplo, mas também de certos fluxos de dados), o atraso na WFQ não é limitado e dependerá do tráfego alimentando as filas
 - Se ocorrer num dado instante uma chegada em rajada de pacotes de tamanho máximo de fluxos diversos, e um destes pacotes for selecionado, o # do ciclo passar a ser do SN dos vários pacotes na WFQ
- Variações de atraso na WFQ poderão ocorrer e afetarão o tráfego interativo de tempo real

Priority Queuing

- Filas PQ são filas FIFO, servidas com prioridade até esvaziar
 - Uma PQ menos prioritária só é atendida se as outras PQ mais prioritárias estão vazias
 - Outras filas só são servidas se as PQ estiverem vazias
- PQ oferece o tratamento preferencial máximo ao tráfego de alta prioridade
- Cisco chama o esquema de uso de PQ como LLQ (*low latency queueing*)

Priority Queuing

- Tráfego entrante prioritário é classificado e enfileirado em filas PQ
 - Classificação pode levar em conta protocolo, interface de entrada, tamanho dos pacotes, fragmentos e listas de acesso
 - Pacotes não classificados para a PQ são enfileirados em outra fila
- Na Cisco, são definidas 4 prioridades: alta, média, normal e baixa

Priority Queuing

- Em casos extremos, com PQ muito utilizada, tráfego de outras filas (como WFQ) pode receber pouca banda ou não ser transmitido
- Para evitar isso, limita-se a banda disponível para a PQ
 - Podem ocorrer descartes na PQ se tráfego ultrapassa a banda reservada

Custom Queuing

- Filas separadas mantidas para cada classe de tráfego e servidas WRR
- Algoritmo requer alocação de um número máximo de bytes para transmissão por fila por ciclo, garantindo uma banda mínima por fila
- CQ circula entre as filas, apanhando o requerido número de bytes de cada uma
 - Se uma fila está vazia, passa para a próxima

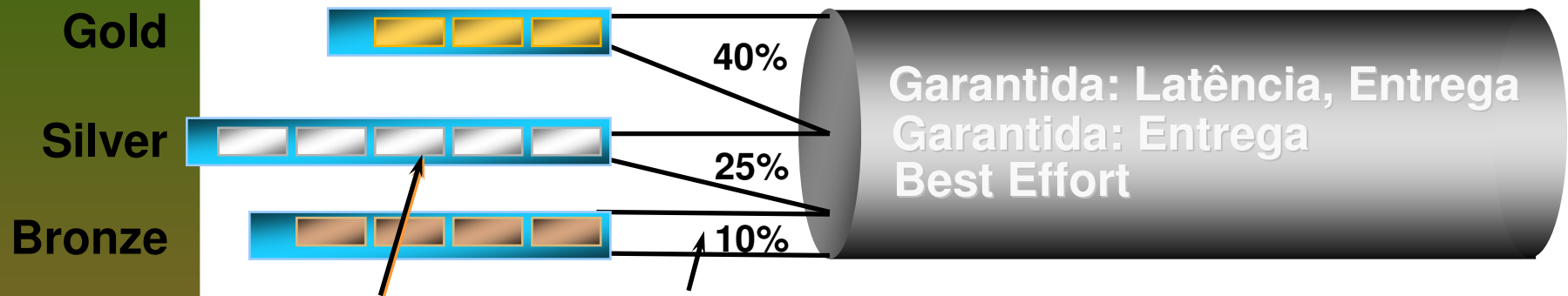
Exemplo: Custom Queuing

- Alocar 20% para protocolo A, 60% para B, 20% para C
- Tamanho dos pacotes em bytes
 - A \Rightarrow 1086 , B \Rightarrow 291, C \Rightarrow 831
- Passos
 - Divida a % pelo tamanho dos pacotes
 - ☞ (20/1086), (60/291), (20/831)
 - Normalize dividindo pelo menor número
 - ☞ 1, (0,20619/0,01842), (0,02407/0,01842)
 - Arredonde para o inteiro mais próximo superiormente
 - ☞ 1, 12, 2
 - Multiplique pelo tamanho do pacote
 - ☞ 1086, 3492, 1662
 - Estes serão os bytes máximos transmitidos a cada visita
- Estática, não é adaptável à carga da rede e consome processador do cartão de interface para classificação

CBWFQ - Class-Based Weighted-Fair Queueing

- Combina características de diferentes políticas
 - Garantia de banda (do Custom Queueing)
 - Distribuição justa entre diversos fluxos (do Weighted-Fair Queueing)
- Cada fila WFQ atende a uma classe de tráfego
- As filas WFQ de uma CBWFQ são atendidas usando o algoritmo de WRR (Weighted-Round-Robin)
 - Algoritmo extremamente eficiente para aplicações de dados
- Não é adequado para garantir prioridade absoluta para aplicações de tempo real, como voz e vídeo, que devem fazer uso de PQ

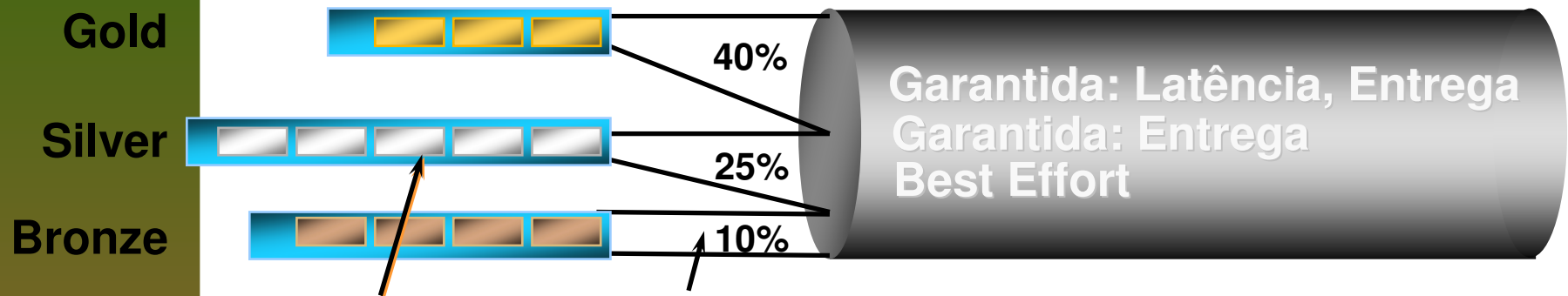
CBWFQ (1)



Passo 1: Define enfileiramento
Passo 2: Define banda

- Define classe, baseando-se em:
 - Precedência IP, DSCP, interface de entrada, grupo de QOS
- Permite especificar diretamente uma banda mínima por classe, diferindo de WFQ onde a banda mínima do fluxo é obtida indiretamente baseada nos pesos de todos os fluxos ativos

CBWFQ (2)

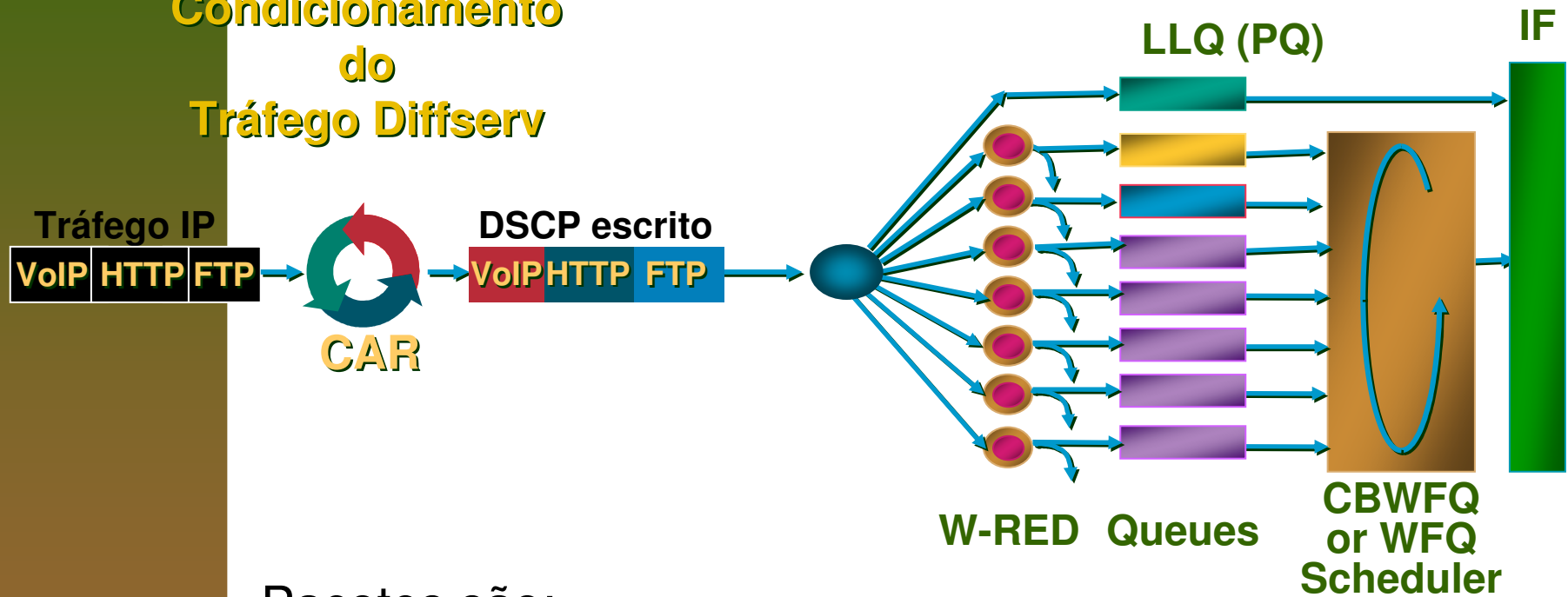


Passo 1: Define enfileiramento
Passo 2: Define banda

- Enfileiramento controla latência
- Capacidade não utilizada é repartida entre as outras classes
- Cada fila pode ser separadamente configurada para QoS
- Benefícios:
 - Pode maximizar transporte de tráfego pago
 - Garante que não há perda de serviço para cada classe

Juntando Tudo em Cisco

Condicionamento do Tráfego Diffserv



Pacotes são:

- “Coloridos” (DSCP escrito) na entrada
- Classificados antes de enfileirar
- Potencialmente descartados com WRED
- Colocados em filas com base no DSCP
- Escalonado com CBWFQ (e/ou LLQ)

Referências

- S. Keshav, An Engineering Approach to Computer Networking, Addison-Wesley, 1997
- Srinivas Vegesna, IP Quality of Service, Cisco Press, 2001

Referências

- RFC-2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (1998)
- RFC-2475, An Architecture for Differentiated Services (1998)
- RFC-2597, Assured Forwarding PHB Group (1999)
- RFC-2697, A Single Rate Three Color Marker (1999)
- RFC-3140, Per Hop Behavior Identification Codes (2001)
- RFC-3246, An Expedited Forwarding PHB (Per-Hop Behavior) (2002)
- RFC-3260, New Terminology and Clarifications for Diffserv (2002)