

Tópicos de Análise de Algoritmos

Algoritmos que executam para sempre:
cubo mágico × tetris

Epílogo do KT

Roteamento de pacotes na internet

Grafo gigante: nós são roteadores e arestas são links entre roteadores.

Roteador tem várias portas de entrada e várias portas de saída.

Pacotes têm um cabeçalho que indica por que porta entraram e por que porta devem sair.

Roteamento de pacotes na internet

Grafo gigante: nós são roteadores e arestas são links entre roteadores.

Roteador tem várias portas de entrada e várias portas de saída.

Pacotes têm um cabeçalho que indica por que porta entraram e por que porta devem sair.

Mais formalmente:

$I_1, \dots, I_n$ são as portas de entrada do roteador

$O_1, \dots, O_n$ são as portas de saída

Cada pacote p é de um tipo $(I[p], O[p])$.

Roteamento de pacotes na internet

Grafo gigante: nós são roteadores e arestas são links entre roteadores.

Roteador tem várias portas de entrada e várias portas de saída.

Pacotes têm um cabeçalho que indica por que porta entraram e por que porta devem sair.

Mais formalmente:

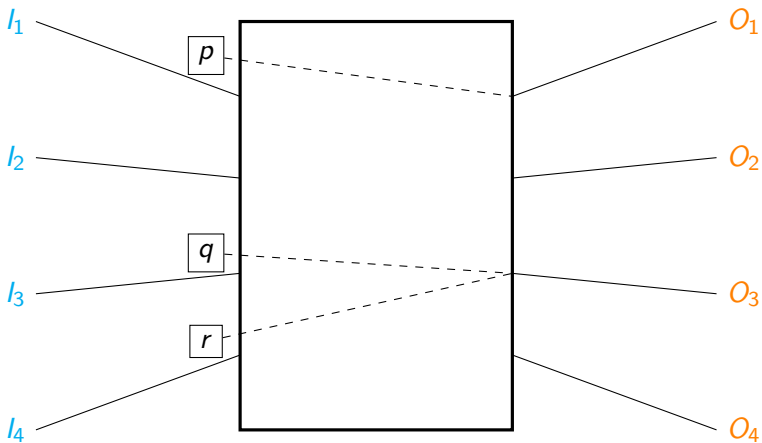
$I_1, \dots, I_n$ são as portas de entrada do roteador

$O_1, \dots, O_n$ são as portas de saída

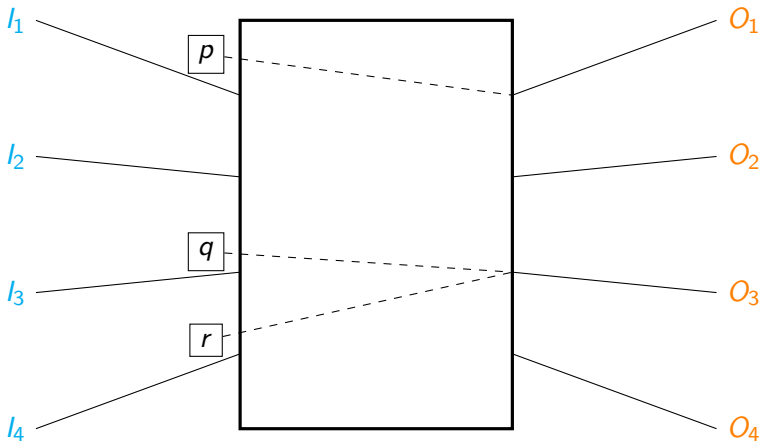
Cada pacote p é de um tipo $(I[p], O[p])$.

O tempo se propaga em passos: a cada passo, no máximo um pacote chega por cada porta de entrada do roteador, e no máximo um pacote parte de uma das portas de saída do roteador.

Roteador

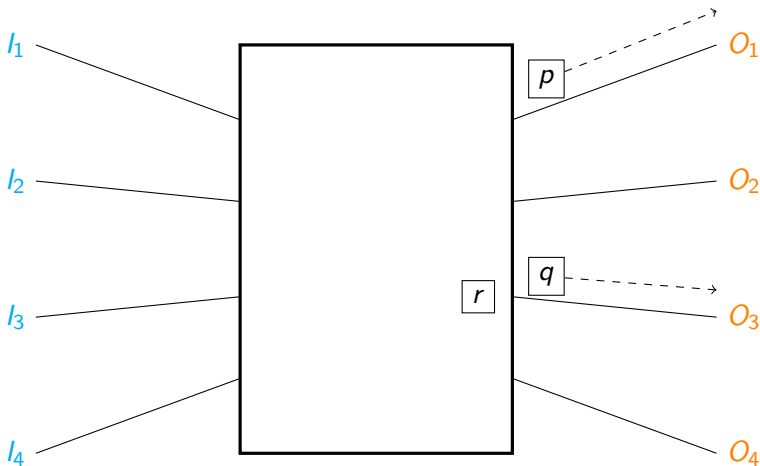


Fila na saída (pure output queueing)



Pacotes chegam e vão para filas nas portas de saída.
Um pacote parte de cada porta de saída.

Fila na saída (pure output queueing)



Pacotes chegam e vão para filas nas **portas de saída**.
Um pacote parte de cada **porta de saída**.

Funcionamento do roteador na prática

Na prática, leva alguns nanosegundos para transferir um pacote de uma porta de entrada para uma porta de saída.

Funcionamento do roteador na prática

Na prática, leva alguns nanosegundos para transferir um pacote de uma porta de entrada para uma porta de saída.

Fila de entrada/saída:

Há uma fila para cada porta de entrada e uma fila para cada porta de saída.

Numa rodada, no máximo um pacote é transferido a partir de cada porta de entrada, e no máximo um pacote é transferido para cada porta de saída.

Funcionamento do roteador na prática

Na prática, leva alguns nanosegundos para transferir um pacote de uma porta de entrada para uma porta de saída.

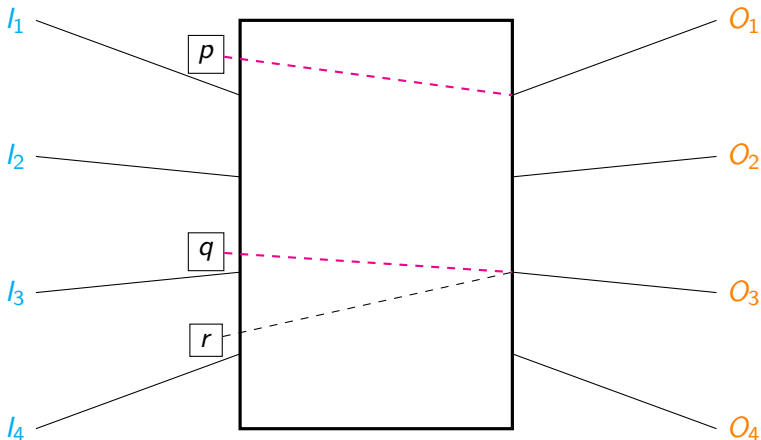
Fila de entrada/saída:

Há uma fila para cada porta de entrada e uma fila para cada porta de saída.

Numa rodada, no máximo um pacote é transferido a partir de cada porta de entrada, e no máximo um pacote é transferido para cada porta de saída.

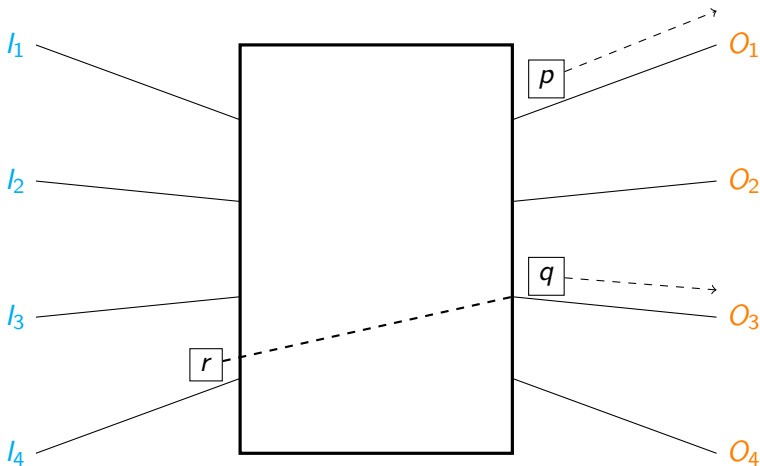
Ou seja, os pacotes transferidos de portas de entrada para portas de saída numa rodada formam um emparelhamento entre portas de entrada e portas de saída.

Filas na entrada e na saída (input/output queueing)



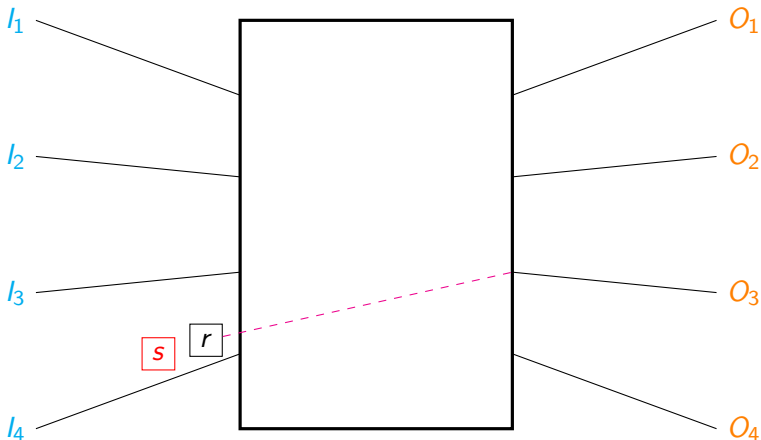
Pacotes chegam nas filas de entrada, um **emparelhamento** é transferido para as filas de saída e parte um pacote por **porta de saída**.

Fila na entrada e na saída (input/output queueing)



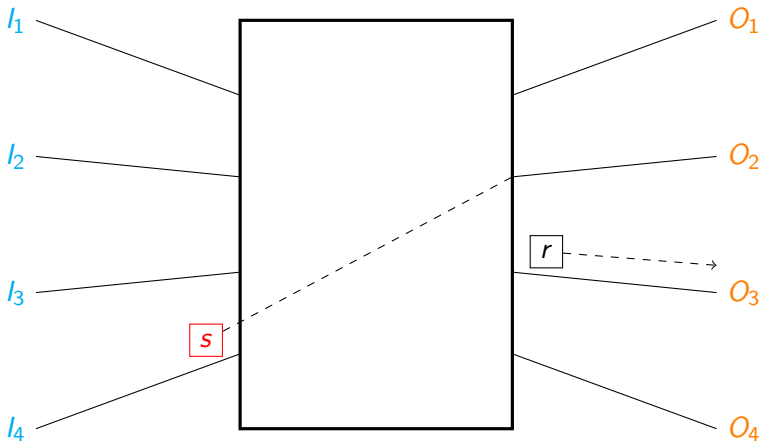
Pacotes chegam nas filas de entrada, um **emparelhamento** é transferido para as filas de saída e parte um pacote por **porta de saída**.

Fila na entrada e na saída (input/output queueing)



Pacotes chegam nas filas de entrada, um **emparelhamento** é transferido para as filas de saída e parte um pacote por **porta de saída**.

Fila na entrada e na saída (input/output queueing)



Pacotes chegam nas filas de entrada, um **emparelhamento** é transferido para as filas de saída e parte um pacote por **porta de saída**.

Mundo ideal × realidade

Mundo ideal: somente filas de saída

Realidade: filas de entrada e de saída

Como simular o funcionamento do mundo ideal com a realidade?

Mundo ideal $\times$ realidade

Mundo ideal: somente filas de saída

Realidade: filas de entrada e de saída

Como simular o funcionamento do mundo ideal com a realidade?

Poderíamos fazer n transferências por rodada, mas isso demora.

Mundo ideal $\times$ realidade

Mundo ideal: somente filas de saída

Realidade: filas de entrada e de saída

Como simular o funcionamento do mundo ideal com a realidade?

Poderíamos fazer n transferências por rodada, mas isso demora.

Vamos mostrar que duas transferências bastam!

Desde que bem escolhidas...

Mundo ideal $\times$ realidade

Mundo ideal: somente filas de saída

Realidade: filas de entrada e de saída

Como simular o funcionamento do mundo ideal com a realidade?

Poderíamos fazer n transferências por rodada, mas isso demora.

Vamos mostrar que duas transferências bastam!

Desde que bem escolhidas...

Rodada: chega no máximo um pacote por porta de entrada;
transferimos um emparelhamento das portas de entrada p/as de saída;
despachamos um pacote no máximo de cada porta de saída e...

Mundo ideal $\times$ realidade

Mundo ideal: somente filas de saída

Realidade: filas de entrada e de saída

Como simular o funcionamento do mundo ideal com a realidade?

Poderíamos fazer n transferências por rodada, mas isso demora.

Vamos mostrar que duas transferências bastam!

Desde que bem escolhidas...

Rodada: chega no máximo um pacote por porta de entrada;
transferimos um emparelhamento das portas de entrada p/as de saída;
despachamos um pacote no máximo de cada porta de saída e...
transferimos um emparelhamento das portas de entrada p/as de saída.

Simulação do esquema com apenas filas de saída

Rodada: chega no máximo um pacote por porta de entrada;
transferimos um emparelhamento das portas de entrada p /as de saída;
despachamos um pacote no máximo de cada porta de saída e...
transferimos um emparelhamento das portas de entrada p /as de saída.

Queremos que os pacotes saiam exatamente
como sairiam se tivéssemos apenas filas nas portas de saída.

Como escolher os emparelhamentos para isso dar certo?

Simulação do esquema com apenas filas de saída

Rodada: chega no máximo um pacote por porta de entrada;
transferimos um emparelhamento das portas de entrada p /as de saída;
despachamos um pacote no máximo de cada porta de saída e...
transferimos um emparelhamento das portas de entrada p /as de saída.

Queremos que os pacotes saiam exatamente
como sairiam se tivéssemos apenas filas nas portas de saída.

Como escolher os emparelhamentos para isso dar certo?

Sempre que o emparelhamento transfere um pacote do tipo (I, O) ,
deve ser o que partiria mais cedo na política com apenas filas de saída.

Simulação do esquema com apenas filas de saída

Rodada: chega no máximo um pacote por porta de entrada;
transferimos um emparelhamento das portas de entrada p /as de saída;
despachamos um pacote no máximo de cada porta de saída e...
transferimos um emparelhamento das portas de entrada p /as de saída.

Queremos que os pacotes saiam exatamente
como sairiam se tivéssemos apenas filas nas portas de saída.

Como escolher os emparelhamentos para isso dar certo?

Sempre que o emparelhamento transfere um pacote do tipo (I, O) ,
deve ser o que partiria mais cedo na política com apenas filas de saída.

$TL(p)$: rodada de partida do pacote p na política com filas de saída

Queremos garantir que cada pacote p parte na rodada $TL(p)$.

Ideia geral

Fila nas portas de entrada **ordenadas por instante de chegada**,
exceto que alguns pacotes podem entrar no meio da fila,
e alguns pacotes podem partir do meio da fila.

Ideia geral

Fila nas portas de entrada **ordenadas por instante de chegada**, exceto que alguns pacotes podem entrar no meio da fila, e alguns pacotes podem partir do meio da fila.

Podemos colocar um pacote que chega mais na frente na fila por ter que partir antes.

Podemos transferir para a saída um pacote que não esteja primeiro pois sua vez de partir está chegando.

Ideia geral

Fila nas portas de entrada **ordenadas por instante de chegada**,
exceto que alguns pacotes podem entrar no meio da fila,
e alguns pacotes podem partir do meio da fila.

Podemos colocar um pacote que chega mais na frente na fila
por ter que partir antes.

Podemos transferir para a saída um pacote que não esteja primeiro
pois sua vez de partir está chegando.

Queremos garantir que todo pacote p já foi transferido em $TL(p)$.

Ideia geral

Fila nas portas de entrada **ordenadas por instante de chegada**, exceto que alguns pacotes podem entrar no meio da fila, e alguns pacotes podem partir do meio da fila.

Podemos colocar um pacote que chega mais na frente na fila por ter que partir antes.

Podemos transferir para a saída um pacote que não esteja primeiro pois sua vez de partir está chegando.

Queremos garantir que todo pacote p já foi transferido em $TL(p)$.

Paralelo com área de check-in e embarque de um aeroporto:

Check-in: temos que passar em tempo da hora de embarque

Área de embarque: só temos que esperar o momento de embarcar

Input cushion e output cushion

Apenas pacotes ainda nas **filas de entrada** importam:
pacotes que já foram transferidos para o embarque
só esperam a rodada certa para partir.

Input cushion e output cushion

Apenas pacotes ainda nas **filas de entrada** importam:
pacotes que já foram transferidos para o embarque
só esperam a rodada certa para partir.

IC(p) : número de pacotes à frente de p na fila de entrada.

OC(p) : número de pacotes já transferidos que partem antes de p .

$$\text{slack}(p) = \text{OC}(p) - \text{IC}(p)$$

Input cushion e output cushion

Apenas pacotes ainda nas **filas de entrada** importam: pacotes que já foram transferidos para o embarque só esperam a rodada certa para partir.

IC(p) : número de pacotes à frente de p na fila de entrada.

OC(p) : número de pacotes já transferidos que partem antes de p .

$$\text{slack}(p) = \text{OC}(p) - \text{IC}(p)$$

Queremos escolher emparelhamentos que garantam o seguinte:

- ▶ **slack(p)** ≥ 0 para todo pacote p nas filas de entrada

Input cushion e output cushion

Apenas pacotes ainda nas **filas de entrada** importam:
pacotes que já foram transferidos para o embarque
só esperam a rodada certa para partir.

$IC(p)$: número de pacotes à frente de p na fila de entrada.

$OC(p)$: número de pacotes já transferidos que partem antes de p .

$$\text{slack}(p) = OC(p) - IC(p)$$

Queremos escolher emparelhamentos que garantam o seguinte:

- ▶ $\text{slack}(p) \geq 0$ para todo pacote p nas filas de entrada
- ▶ Toda rodada em que $IC(p) = OC(p) = 0$, o pacote p é transferido no primeiro emparelhamento da rodada.

Por que isso basta?

Se

- ▶ $\text{slack}(p) \geq 0$ para todo pacote p nas filas de entrada e
- ▶ toda rodada em que $\text{IC}(p) = \text{OC}(p) = 0$, o pacote p é transferido no primeiro emparelhamento da rodada, então todo pacote parte na rodada certa.

Por que isso basta?

Se

- ▶ $\text{slack}(p) \geq 0$ para todo pacote p nas filas de entrada e
- ▶ toda rodada em que $\text{IC}(p) = \text{OC}(p) = 0$, o pacote p é transferido no primeiro emparelhamento da rodada, então todo pacote parte na rodada certa.

Esboço da prova:

Na rodada $\text{TL}(p)$, ou p já está foi transferido ou $\text{OC}(p) = 0$.

Por que isso basta?

Se

- ▶ $\text{slack}(p) \geq 0$ para todo pacote p nas filas de entrada e
- ▶ toda rodada em que $\text{IC}(p) = \text{OC}(p) = 0$, o pacote p é transferido no primeiro emparelhamento da rodada, então todo pacote parte na rodada certa.

Esboço da prova:

Na rodada $\text{TL}(p)$, ou p já está foi transferido ou $\text{OC}(p) = 0$.
Então $\text{IC}(p) = 0$ pois $\text{slack}(p) \geq 0$,
e p será transferido no primeiro emparelhamento desta rodada. □

Por que isso basta?

Se

- ▶ $\text{slack}(p) \geq 0$ para todo pacote p nas filas de entrada e
- ▶ toda rodada em que $\text{IC}(p) = \text{OC}(p) = 0$, o pacote p é transferido no primeiro emparelhamento da rodada, então todo pacote parte na rodada certa.

Esboço da prova:

Na rodada $\text{TL}(p)$, ou p já está foi transferido ou $\text{OC}(p) = 0$.
Então $\text{IC}(p) = 0$ pois $\text{slack}(p) \geq 0$,
e p será transferido no primeiro emparelhamento desta rodada. □

A primeira propriedade acima é fácil de garantir.

A que exige cuidados é a segunda.

Como garantir as propriedades?

Quando um pacote p chega,
coloque-o o mais atrás na fila de modo que $\text{slack}(p) \geq 0$.

Como garantir as propriedades?

Quando um pacote p chega,
coloque-o o mais atrás na fila de modo que $\text{slack}(p) \geq 0$.

Queremos manter $\text{slack}(p) \geq 0$ de uma rodada para outra.

Como garantir as propriedades?

Quando um pacote p chega,
coloque-o o mais atrás na fila de modo que $\text{slack}(p) \geq 0$.

Queremos manter $\text{slack}(p) \geq 0$ de uma rodada para outra.

Quando um pacote p está na fila de entrada,
 $\text{slack}(p)$ pode mudar, diminuindo de 1:

- se um pacote que chega entra na sua frente.

- ($\text{IC}(p)$ aumentou de 1)

- se um pacote com TL menor parte da sua porta de saída.

- ($\text{OC}(p)$ diminuiu de 1)

Como garantir as propriedades?

Quando um pacote p chega,
coloque-o o mais atrás na fila de modo que $\text{slack}(p) \geq 0$.

Queremos manter $\text{slack}(p) \geq 0$ de uma rodada para outra.

Quando um pacote p está na fila de entrada,
 $\text{slack}(p)$ pode mudar, diminuindo de 1:

- se um pacote que chega entra na sua frente.

- ($\text{IC}(p)$ aumentou de 1)

- se um pacote com TL menor parte da sua porta de saída.

- ($\text{OC}(p)$ diminuiu de 1)

Basta garantir que cada emparelhamento aumenta $\text{slack}(p)$ de 1!

Como garantir que $\text{slack}(p)$ aumenta de 1?

Note que $\text{slack}(p)$ aumenta de 1 se
um pacote da frente de p é transferido para uma fila de saída ou

Como garantir que $\text{slack}(p)$ aumenta de 1?

Note que $\text{slack}(p)$ aumenta de 1 se

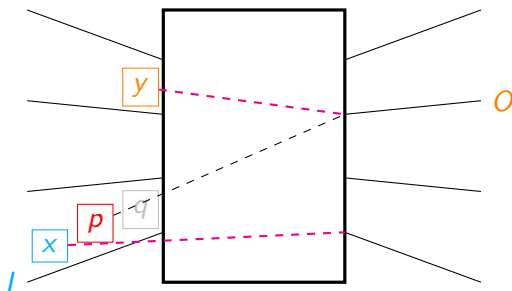
- um pacote da frente de p é transferido para uma fila de saída ou
- um pacote que parte antes de p de sua porta de saída é transferido.

Como garantir que $\text{slack}(p)$ aumenta de 1?

Note que $\text{slack}(p)$ aumenta de 1 se

um pacote da frente de p é transferido para uma fila de saída ou
um pacote que parte antes de p de sua porta de saída é transferido.

Para $\text{slack}(p)$ não aumentar de pelo menos 1,
o emparelhamento transfere um pacote x de trás de $p = (I, O)$
e um pacote y para a porta de saída de p que partirá depois de p .



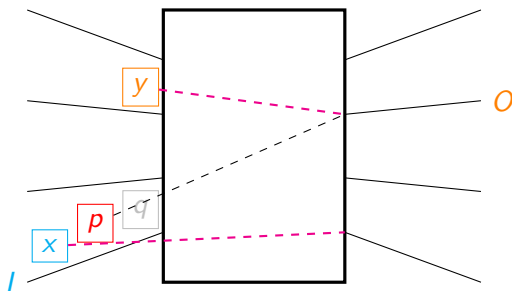
Como garantir que $\text{slack}(p)$ aumenta de 1?

Note que $\text{slack}(p)$ aumenta de 1 se

um pacote da frente de p é transferido para uma fila de saída ou
um pacote que parte antes de p de sua porta de saída é transferido.

Para $\text{slack}(p)$ não aumentar de pelo menos 1,

o emparelhamento transfere um pacote x de trás de $p = (I, O)$
e um pacote y para a porta de saída de p que partirá depois de p .



(I, O) é um par bloqueador! Escolha emparelhamentos estáveis!

Emparelhamento estável

Ordem de preferência para porta de entrada $/$:
porta de saída de pacote mais à frente na sua fila.

Emparelhamento estável

Ordem de preferência para porta de entrada *I*:
porta de saída de pacote mais à frente na sua fila.

Ordem de preferência para porta de saída *O*:
porta de entrada com pacote que parte mais cedo.

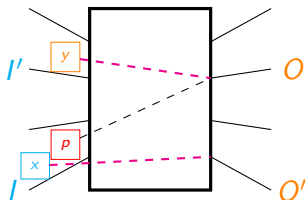
Emparelhamento estável

Ordem de preferência para porta de entrada I :
porta de saída de pacote mais à frente na sua fila.

Ordem de preferência para porta de saída O :
porta de entrada com pacote que parte mais cedo.

Par (I, O) é **bloqueador** se

pacote enviado por I está depois de p ,
pacote recebido por O partirá depois de p ,
onde p é o primeiro pacote do tipo (I, O) na fila de I



Emparelhamento estável

Ordem de preferência para porta de entrada I :
porta de saída de pacote mais à frente na sua fila.

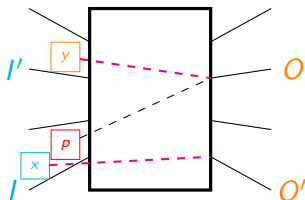
Ordem de preferência para porta de saída O :
porta de entrada com pacote que parte mais cedo.

Par (I, O) é **bloqueador** se

pacote enviado por I está depois de p ,

pacote recebido por O partirá depois de p ,

onde p é o primeiro pacote do tipo (I, O) na fila de I



I mandou pacote para a saída O' e O recebeu pacote da entrada I'
mas I prefere O a O' e O prefere I a I' .

Emparelhamento estável

Ordem de preferência para porta de entrada I :
porta de saída de pacote mais à frente na sua fila.

Ordem de preferência para porta de saída O :
porta de entrada com pacote que parte mais cedo.

Escolha emparelhamentos estáveis em relação a estas ordens!

Emparelhamento estável

Ordem de preferência para porta de entrada I :
porta de saída de pacote mais à frente na sua fila.

Ordem de preferência para porta de saída O :
porta de entrada com pacote que parte mais cedo.

Escolha emparelhamentos estáveis em relação a estas ordens!

Se o emparelhamento é estável, não há par bloqueador.

Emparelhamento estável

Ordem de preferência para porta de entrada I :
porta de saída de pacote mais à frente na sua fila.

Ordem de preferência para porta de saída O :
porta de entrada com pacote que parte mais cedo.

Escolha emparelhamentos estáveis em relação a estas ordens!

Se o emparelhamento é estável, não há par bloqueador.

Se não há par bloqueador,
 $\text{slack}(p)$ cresce de 1 a cada emparelhamento.

Emparelhamento estável

Ordem de preferência para porta de entrada I :
porta de saída de pacote mais à frente na sua fila.

Ordem de preferência para porta de saída O :
porta de entrada com pacote que parte mais cedo.

Escolha emparelhamentos estáveis em relação a estas ordens!

Se o emparelhamento é estável, não há par bloqueador.

Se não há par bloqueador,
 $\text{slack}(p)$ cresce de 1 a cada emparelhamento.

Logo $\text{slack}(p) \geq 0$, ou seja, as duas propriedades valem
e cada pacote p parte em $\text{TL}(p)$, como desejado.