Přenos dat, počítačové sítě a protokoly

Architektura přepínačů

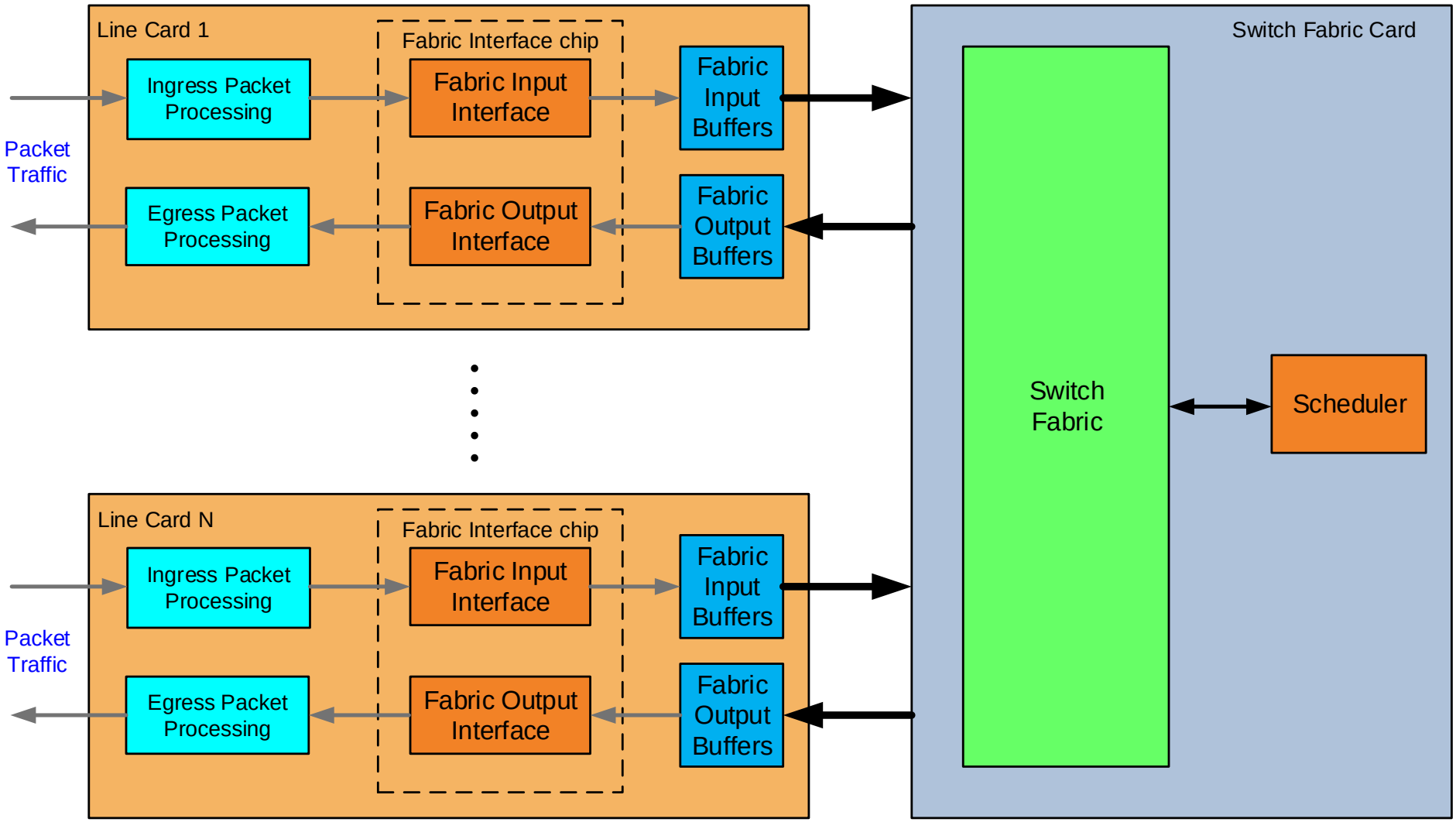
Ing. Petr Matoušek, Ph.D., M.A.

Fakulta informačních technologií VUT v Brně

matousp@fit.vutbr.cz

Obsah přednášky

Obecná architektura přepínače



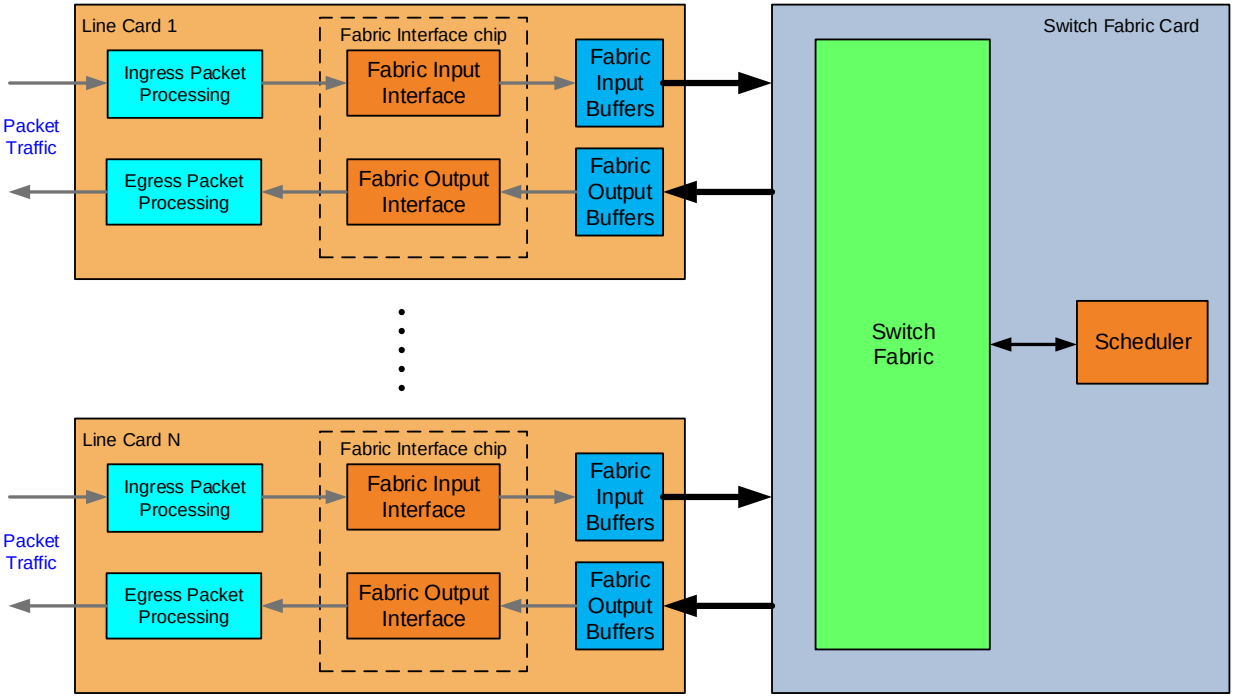
Obecná architektura přepínače

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Základní části přepínače

- *Vstupní rozhraní (fabric input interface)*
  - Propojuje vstupní modul a přepínací logiku
  - Rozděluje příchozí pakety na buňky pevné délky pro přepínání
  - Odesílá data na přepínací logiku na pokyn plánovače
- *Vstupní buffer (fabric input buffer)*
  - Ukládá příchozí data, pokud je není možné poslat do přepínací logiky
- *Přepínací logika (switch fabric)*
  - Dynamicky propojuje síťové rozhraní a přenáší mezi nimi data.
  - Vytváří propojení pomocí přepínacích obvodů či sběrnic.
  - Implementována na základní desce
- *Plánovač (scheduler)*
  - Plánuje přepínání dat ze vstupu na výstup
- *Výstupní rozhraní (fabric output interface)*
- *Výstupní buffer (fabric output buffer)*

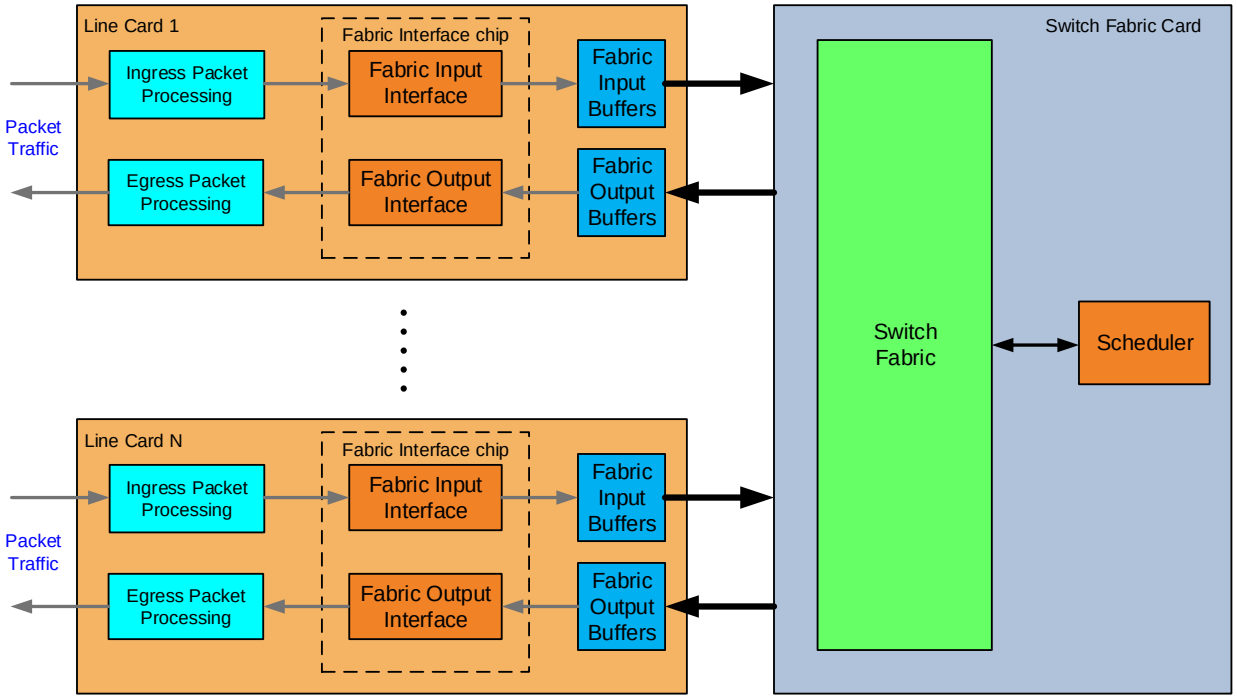
Obecná architektura přepínače



Požadavky na činnost přepínače

- Maximální přenos dat přepínací logikou
- Paralelní přenosy mezi různými síťovými rozhraními
- Spravedlivé přidělování přenosového pásma
- Zachování pořadí paketů

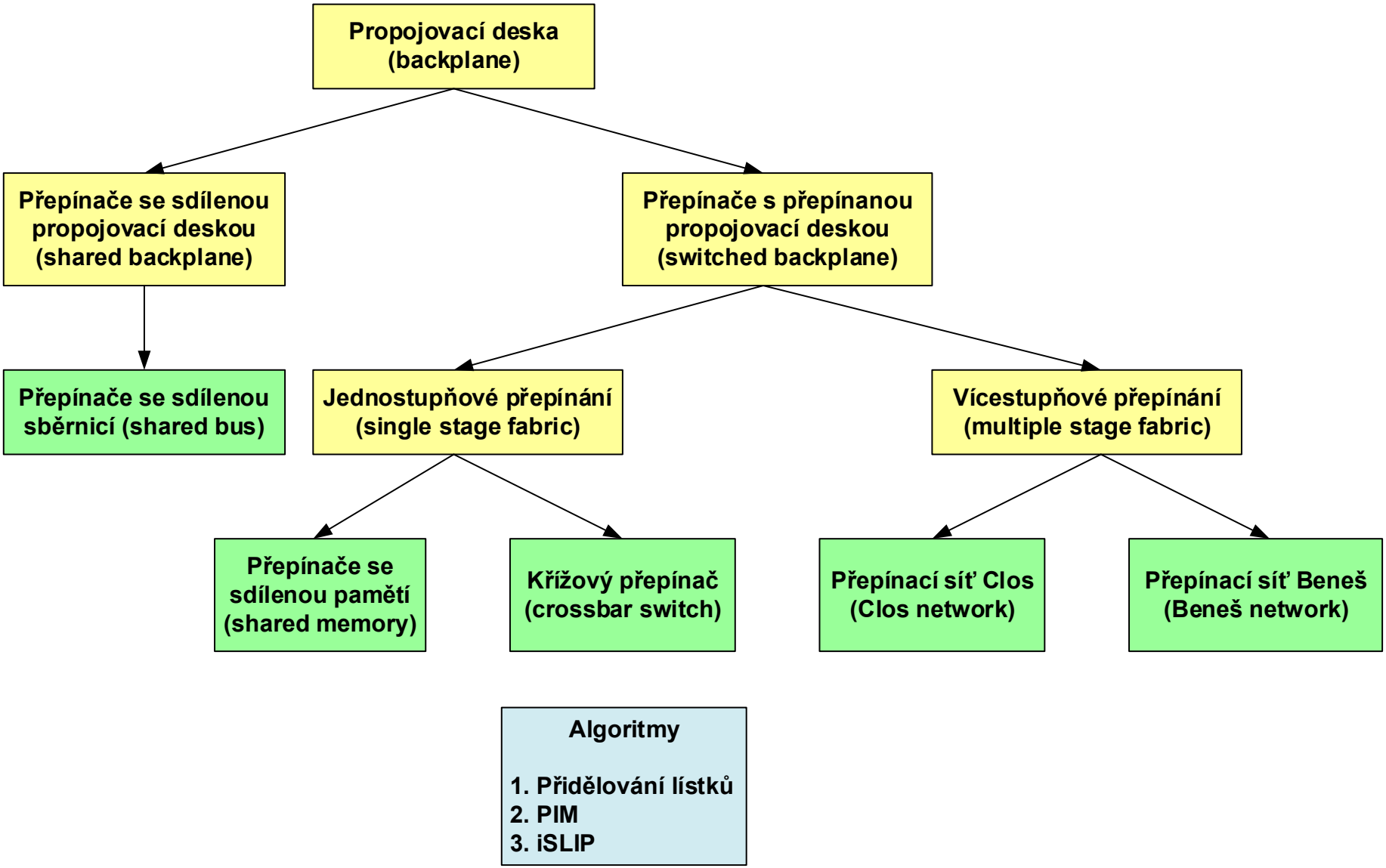
Obecná architektura přepínače



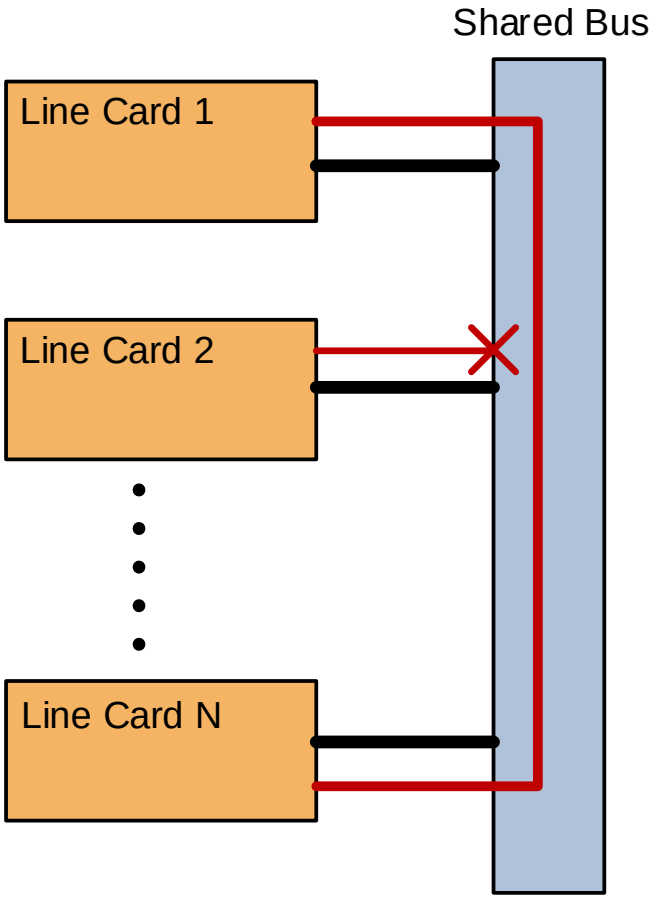
Metriky pro měření činnosti přepínače

- *Propustnost:* množství dat přenesných za jednotku času (bps či pps)
- *Latence:* doba přenosu paketu ze vstupního na výstupní rozhraní (s)
- *Počet dostupných cest* v přepínací logice, které propojují každou dvojici vstupních a výstupních portů (blokující vs. neblokující přepínání).

Klasifikace přepínačů [1]



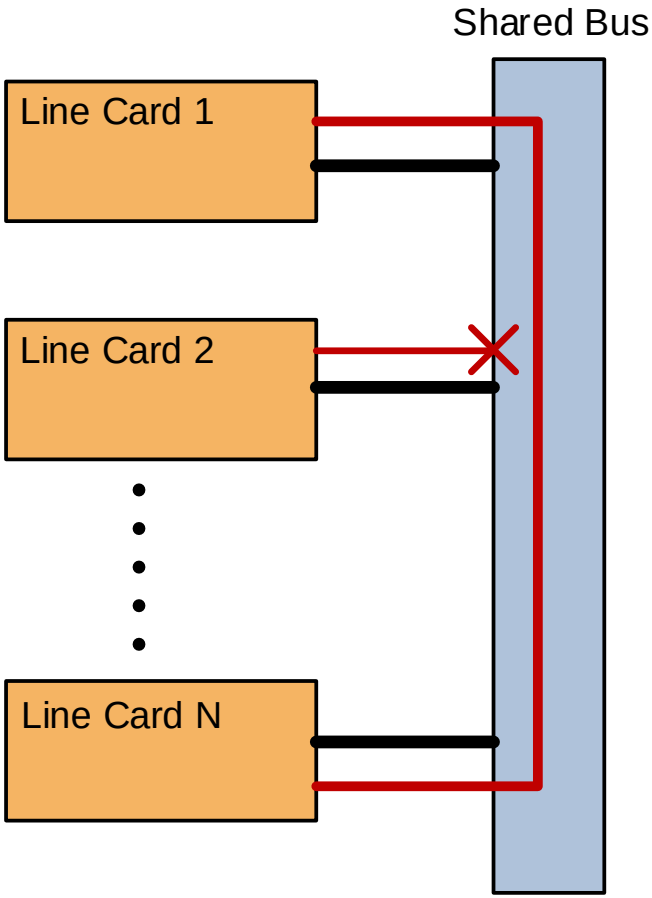
Přepínače se sdílenou sběrnicí



- Sdílené přenosové médiu
- Bus protocol $\Rightarrow$ řízení přenosů
- Potřebná propustnost sběrnice: $R \times N$
- Potřebná šířka sběrnice: $w = \frac{R \times N}{r}$
  - R: šířka pásma na portu
  - N: počet portů
  - r: taktovací frekvence sběrnice
- Vhodné pro zařízení s propustností 1 Gb/s

- Nativní implementace přenosu broadcast a multicast
- Pouze jeden port může v daný okamžik komunikovat
- S počtem portů roste šířka sběrnice

Přepínače se sdílenou sběrnicí

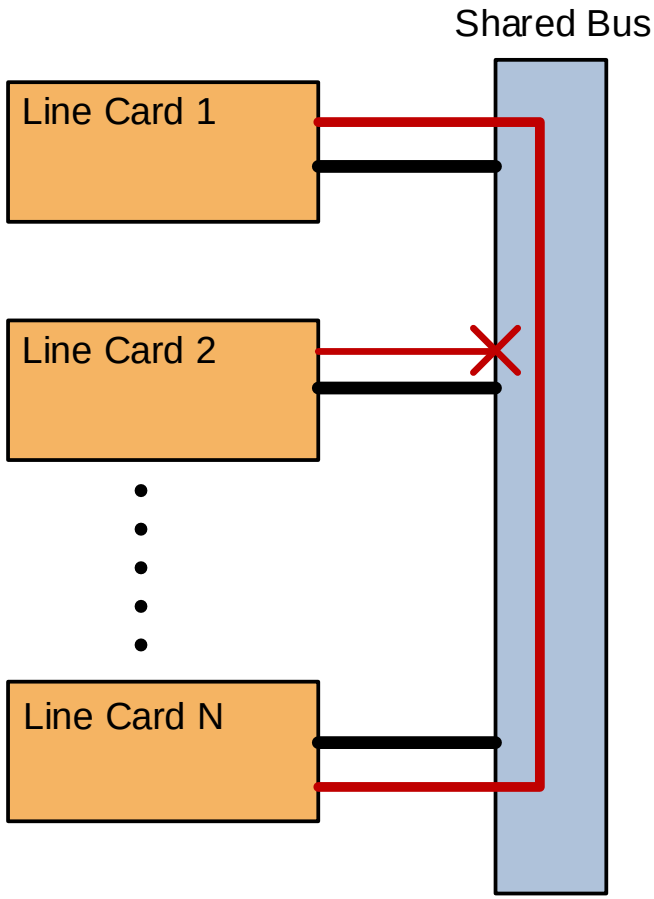


- Sdílené přenosové médiu
- Bus protocol $\Rightarrow$ řízení přenosů
- Potřebná propustnost sběrnice: $R \times N$
- Potřebná šířka sběrnice: $w = \frac{R \times N}{r}$
  - R: šířka pásma na portu
  - N: počet portů
  - r: taktovací frekvence sběrnice
- Vhodné pro zařízení s propustností 1 Gb/s

Úkol:

- Přepínač má 16 portů o rychlosti 100 Mb/s. Jaká je potřebná propustnost sběrnice a jaká musí být šířka sběrnice, aby zvládl požadovaný přenos? Uvažujte taktovací frekvenci sběrnice 40 MHz.

Přepínače se sdílenou sběrnicí



- Sdílené přenosové médiu
- Potřebná propustnost sběrnice: $R \times N$
- Potřebná šířka sběrnice: $w = \frac{R \times N}{r}$
  - R: šířka pásma na portu
  - N: počet portů
  - r: taktovací frekvence sběrnice

$\Rightarrow$ Co se stane, když přidáme porty?

$\Rightarrow$ Co se stane, když se zvýší rychlost na portech?

Řešení:

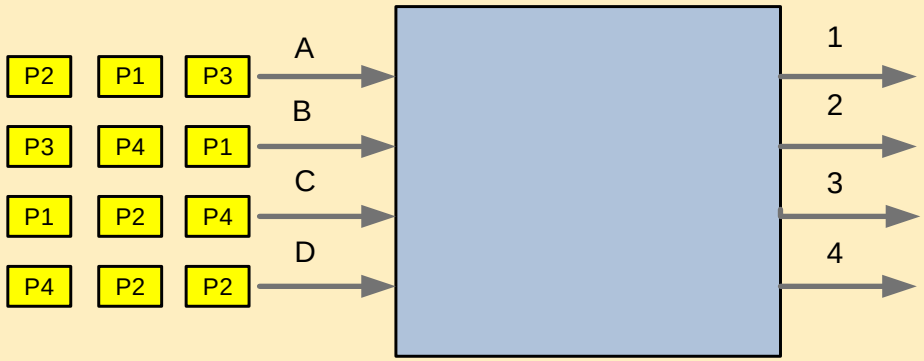
Přepínač má 16 portů o rychlosti 100 Mb/s, taktovací frekvence je sběrnice 40 MHz.

- Propustnost $R \times N = 1{,}6$ Gb/s ; šířka sběrnice $w = \frac{R \times N}{r} = \frac{100.10^6 \times 16}{40.10^6} = 40$ bitů

Přepínaná propojovací deska (switched backplane)

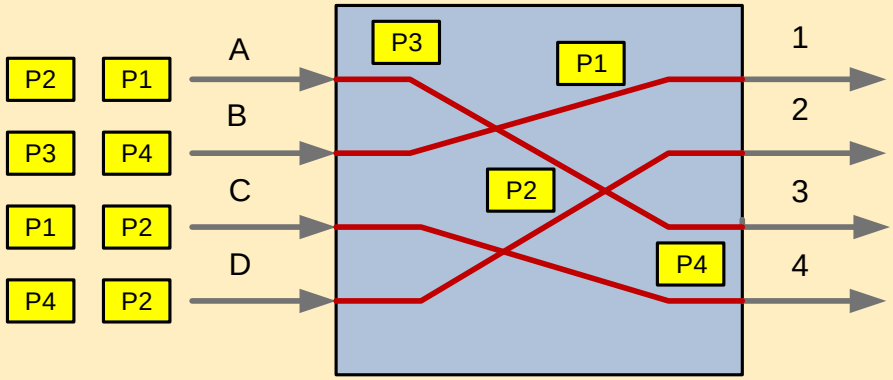
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Vlastnosti



- Paralelní přenos paketů
- Buňky pevné délky
- Potřeba plánování: plánovač (scheduler)

Proces plánování přenosu

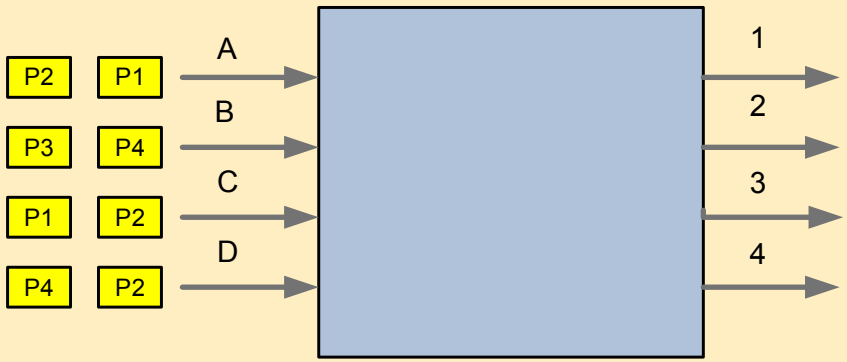


1. Detekce buněk na vstupu
2. Plánování přenosu
3. Přenos buněk

Přepínaná propojovací deska (switched backplane)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Příklady architektur s přepínanou propojovací deskou

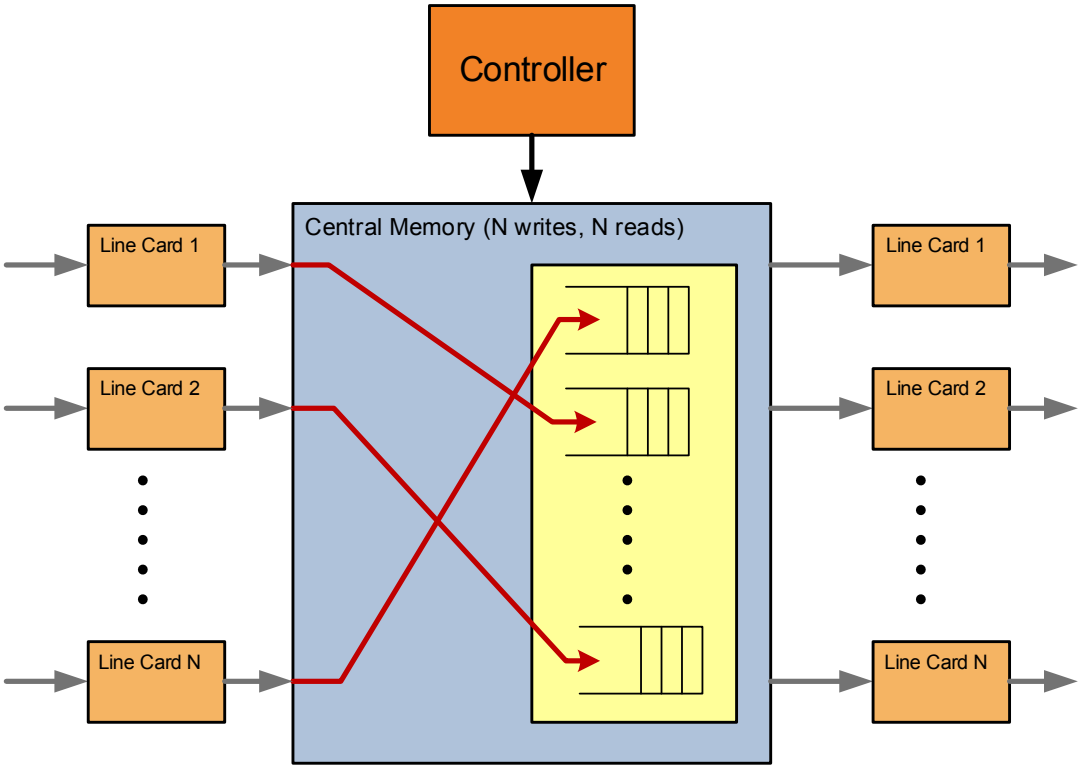


- *Jednostupňové:*
  - se sdílenou pamětí
  - křížový přepínač
- *Vícestupňové:*
  - Closova síť
  - Benešova síť
  - Torus

Plánovač (Scheduler)

- *Činnost plánovače*
  - Problém párování portů
  - Vnitřní blokování a virtuální fronty
- *Algoritmy pro plánování*
  - Přidělování lístků (take-a-ticket)
  - Algoritmus PIM (Parallel Iterative Matching)
  - Algoritmus iSLIP (Iterative round-robin matching with SLIP)

Přepínač se sdílenou pamětí (shared memory)

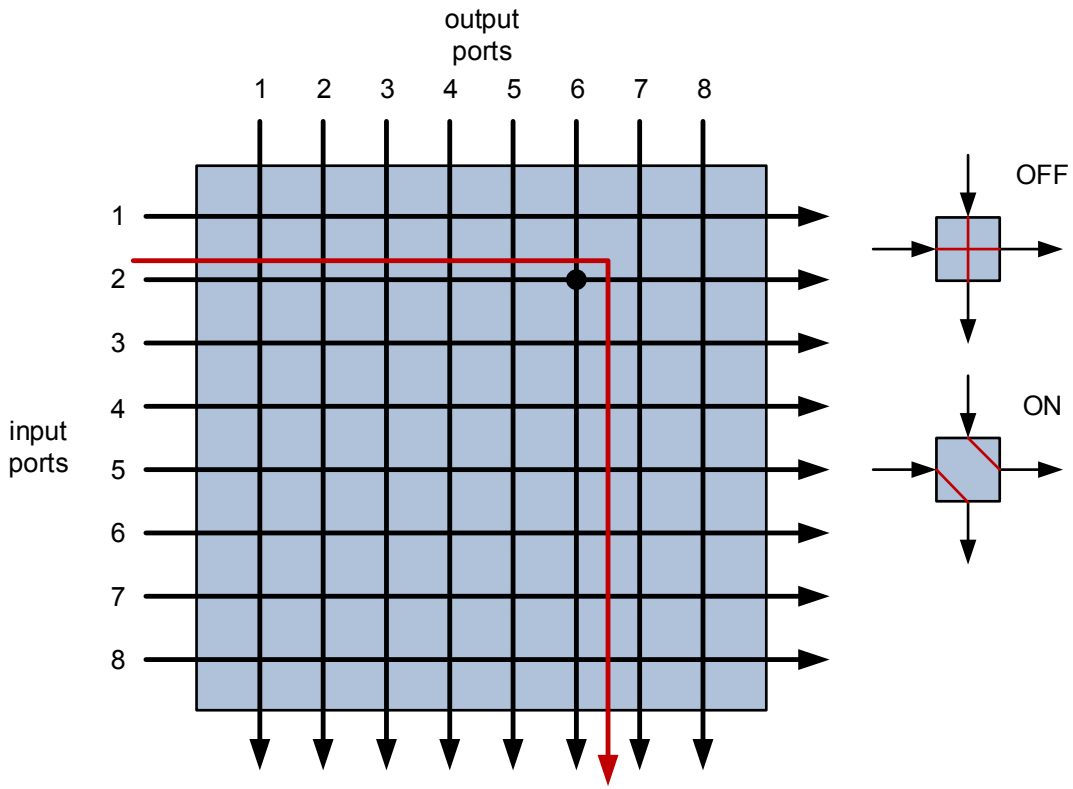


- Centrální sdílená paměť
- Fronty pro výstupní porty
  - Oblasti pevné velikosti
  - Oblasti proměnné velikosti
- Rychlost přístupu do paměti
  - $BW = 2 \times N \times R\ [b/s]$
  - R: rychlost síťového rozhraní
  - N: počet síťových karet
- Doba přenosu dat o velikosti C
  - $t = \frac{C}{BW}\ [s]$

Úkol

- Jaká je potřebná doba zápisu do paměti u přepínače se sdílenou pamětí, který má 32 portů o rychlosti 1 Gb/s a velikost ukládané buňky je 40 bytů? Jak se tato doba změní pro rychlosti portů 10 Gb/s a 40 Gb/s?
- $t_1 = \frac{c}{2 \times R \times N} = \frac{40 \times 8}{2 \times 32 \times 10^9} = 5.10^{-9}\ s = 5\ ns;\ t_{10} = 0,5\ ns;\ \ t_{40} = 0,125\ ns.$
- Přístupová doba pamětí SDRAM je 5 – 10 ns, u pamětí DRAM cca 50 ns.

Křížový přepínač (crossbar)



- N^2 propojení (crosspoints)
  - Dvoustavové chování (on/off)
  - Implementace např. tranzistory
- Interně neblokující architektura
  - Paralelní přenosy
- Nativní podpora multicastu
- Jednoduchá implementace
- Centrální plánovač
  - Plánuje konfigurace přepínače
  - Nastavení pomocí řídící linky

Další vlastnosti křížového přepínače

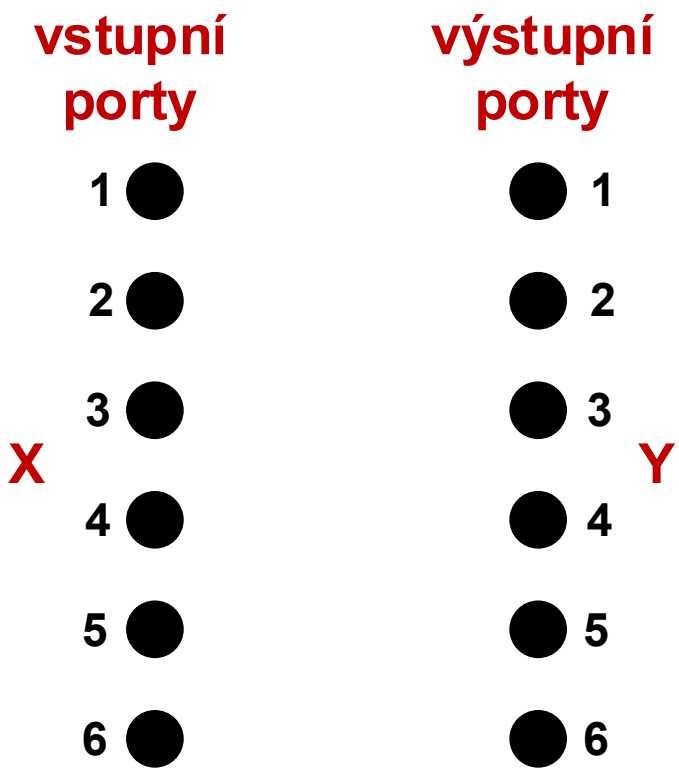
- Náročná rozšiřitelnost: kvadratická složitost $\mathcal{O}(N^2)$
- Možnost kolizí: více vstupů požaduje jeden výstup
- Obtížná garance kvality služeb při souběžných požadavcích
- Chybí redundance: pouze jedna cesta mezi vstupem a výstupem

Problém párování (matching)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Algoritmický problém párování

- Bipartitní graf $G = (X \cup Y, E)$
- Uzly: množina vstupních portů X a množina výstupních portů Y
- Hrany: množina propojení E (požadavky na přenos dat)

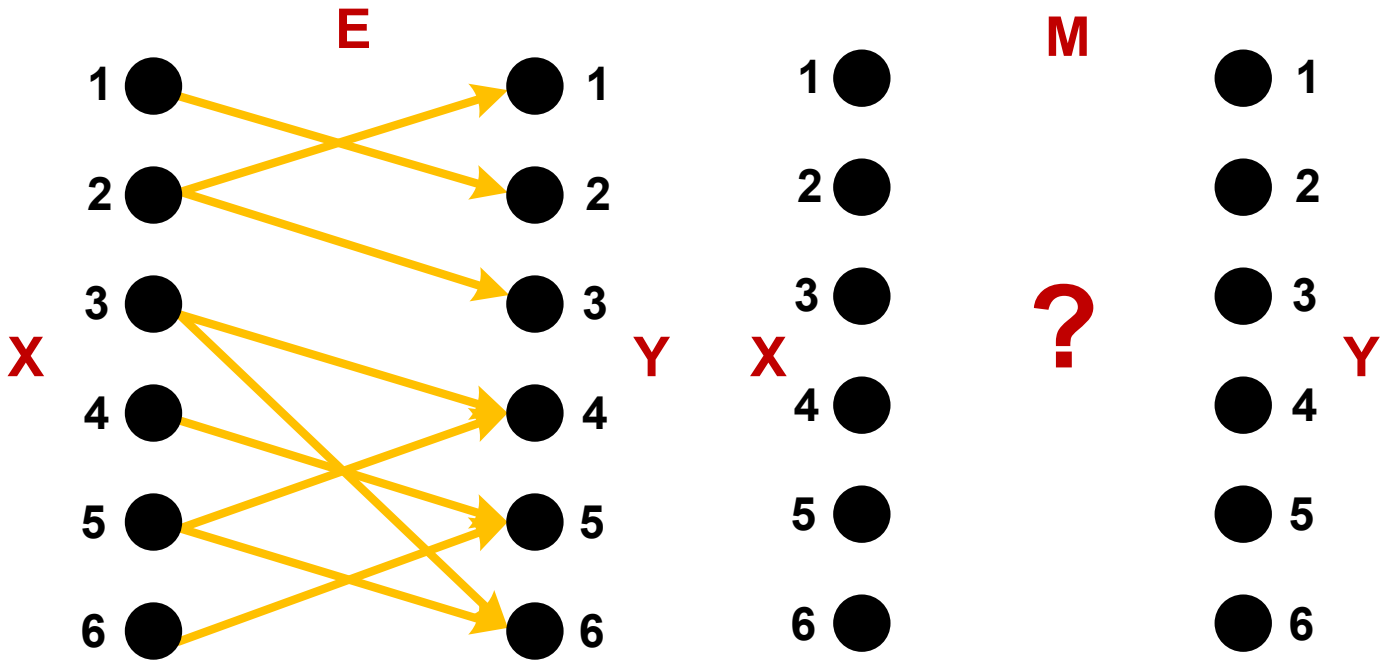


Problém párování (matching)

Algoritmický problém párování

- Bipartitní graf $G = (X \cup Y, E)$
- Uzly: množina vstupních portů X a množina výstupních portů Y
- Hrany: množina propojení E (požadavky na přenos dat)

- Párování vstupních portů X na výstupní porty Y
- Hledáne takové párování $M \subseteq E$, kde žádné dvě hrany z M nemají společný vrchol

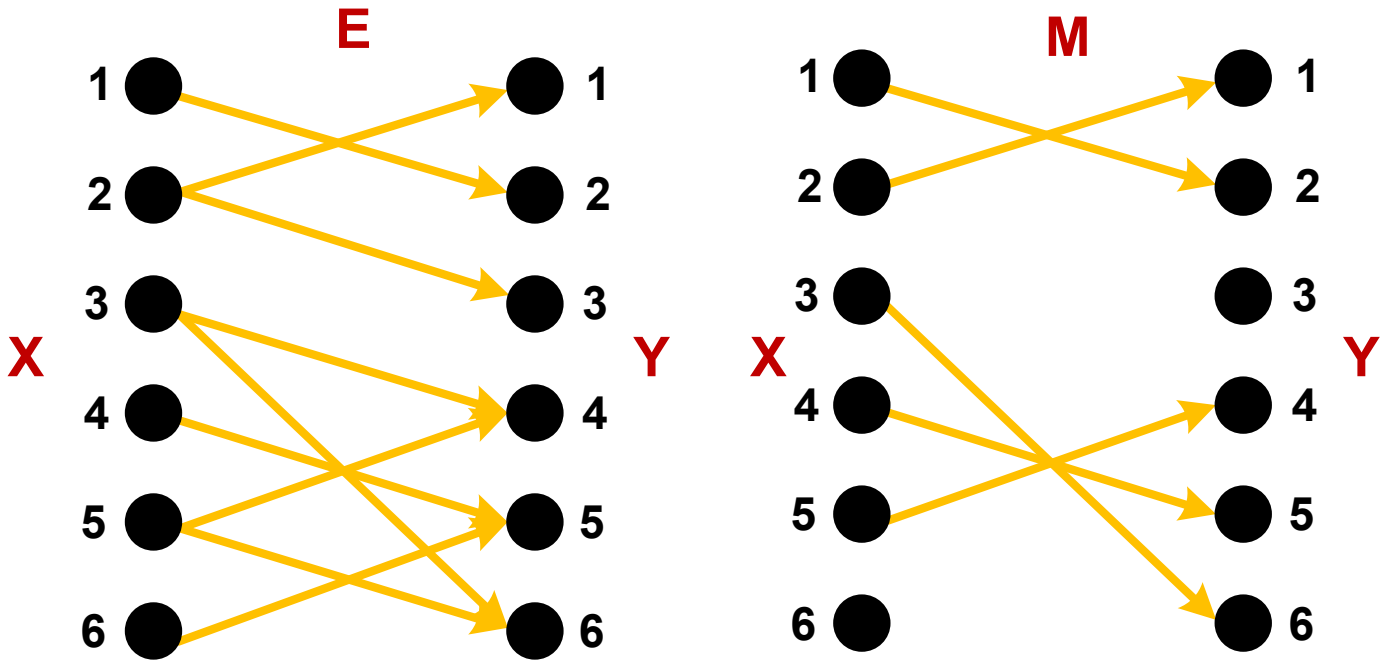


Problém párování (matching)

Algoritmický problém párování

- Bipartitní graf $G = (X \cup Y, E)$
- Uzly: množina vstupních portů X a množina výstupních portů Y
- Hrany: množina propojení E (požadavky na přenos dat)

- Párování vstupních portů X na výstupní porty Y
- Hledáne takové párování $M \subseteq E$, kde žádné dvě hrany z M nemají společný vrchol



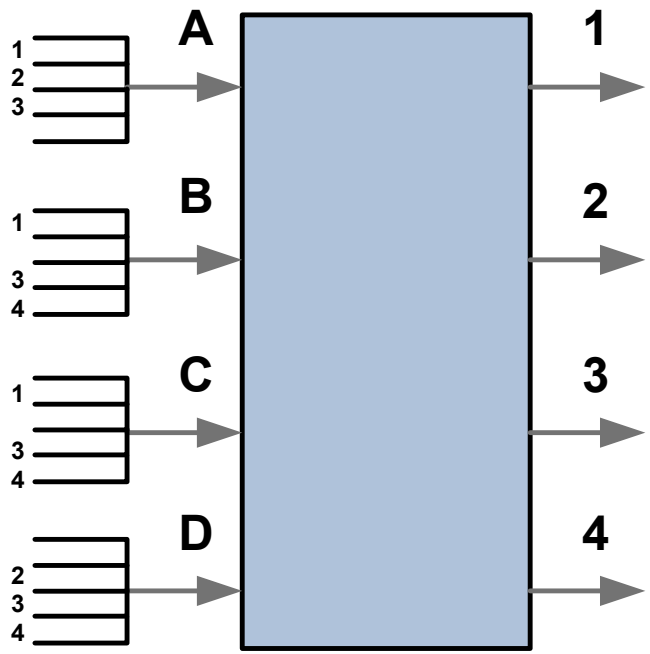
Problém párování (matching)

Největší vs. maximální párování

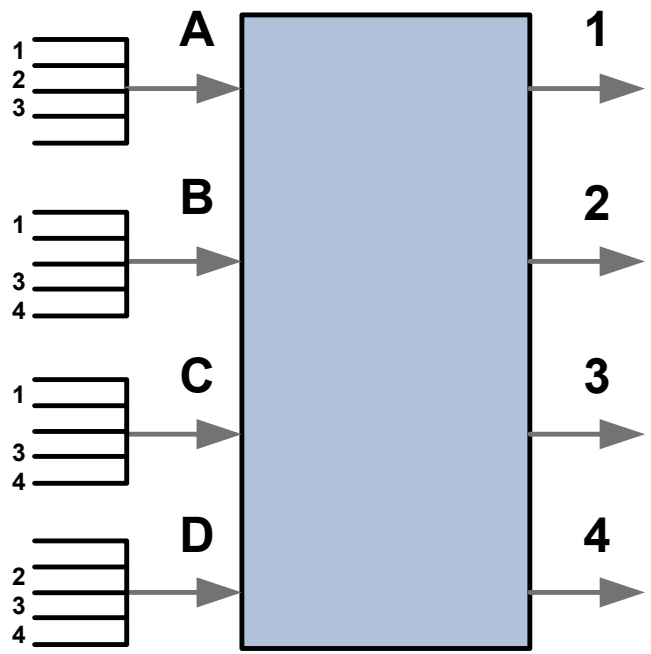
- *Největší párování (maximum matching)*: párování, které má největší počet hran
- *Maximální párování (maximal matching)*: párování, kdy nelze přidat další hranu

Požadavek na přenos:
$\{(A,1),(A,2),(A,3),(B,1),(B,3),(B,4),(C,1),(C,3),(C,4),(D,2),(D,3),(D,4)\}$

Největší párování M_1



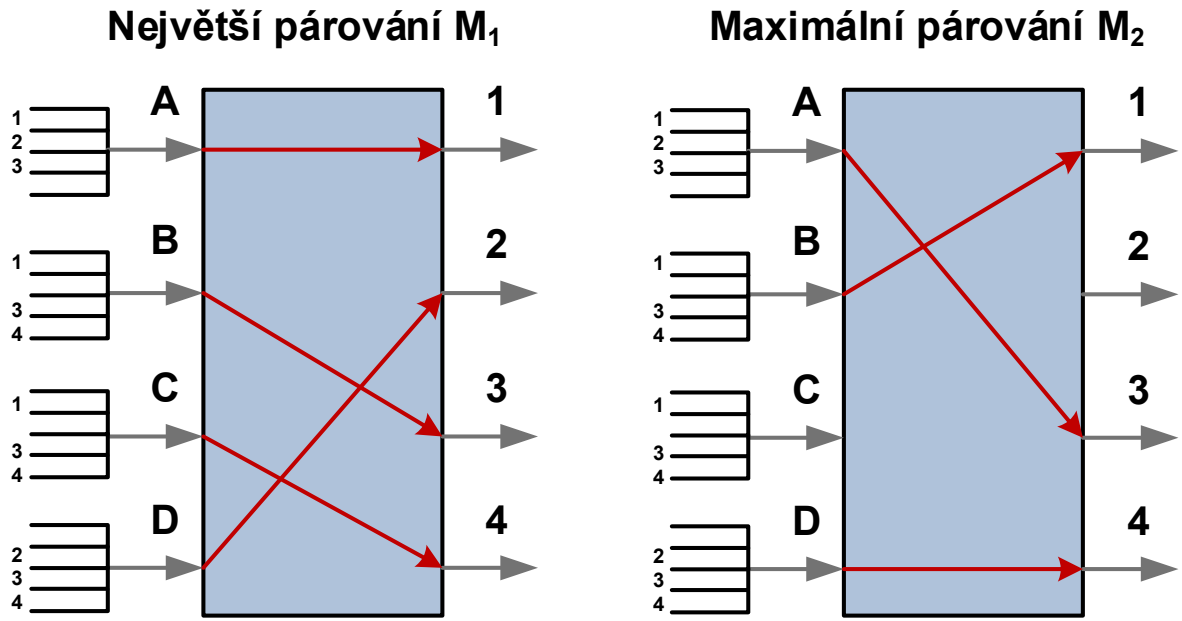
Maximální párování M_2



Problém párování (matching)

Největší vs. maximální párování

- *Největší párování (maximum matching)*: obsahuje největší možný počet hran
  - Globální maximum s největší propustností
  - Složitost vyhledání $\mathcal{O}(N^{\frac{5}{2}})$ či $\mathcal{O}(N+E)$
  - Nebezpečí vyhladovění
- *Maximální párování (maximal matching)*
  - Nelze přidat další hranu, aniž bychom zvýšili stupeň některého z uzlů na dva.
  - Lokální maximum



Plánovací algoritmus přidělování lístků

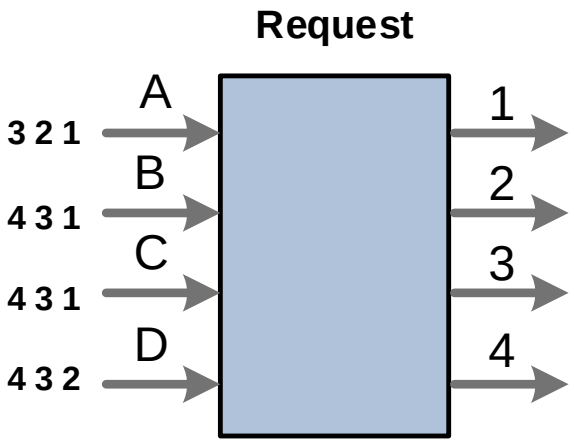
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

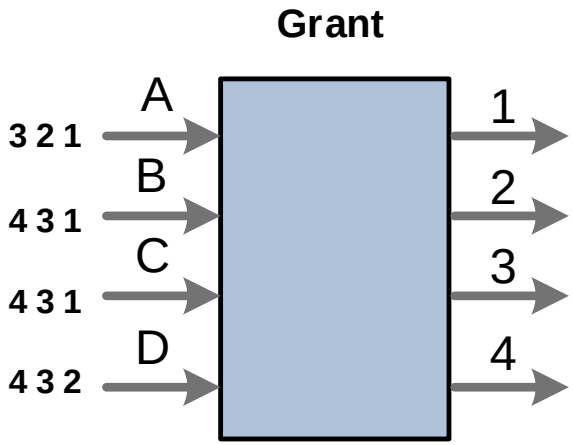
Princip algoritmu

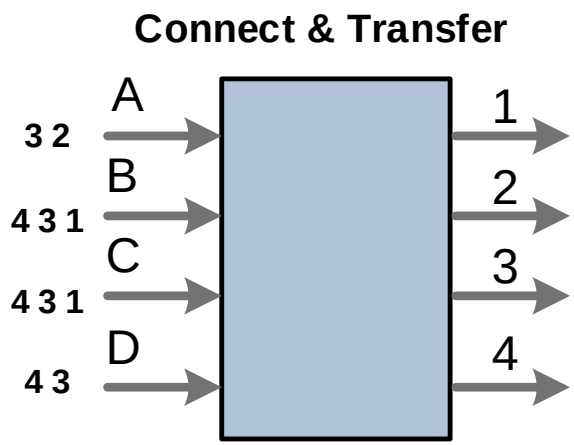
- Výstupní port Q obsluhuje frontu požadavků na propojení.
- Požadavek na vstupu P dostane od Q číslo T_{QX} na obsloužení portu Q s pořadím X
- Výstupní porty jsou přiděleny žádostem s nižším T_{QX}

Fáze algoritmu

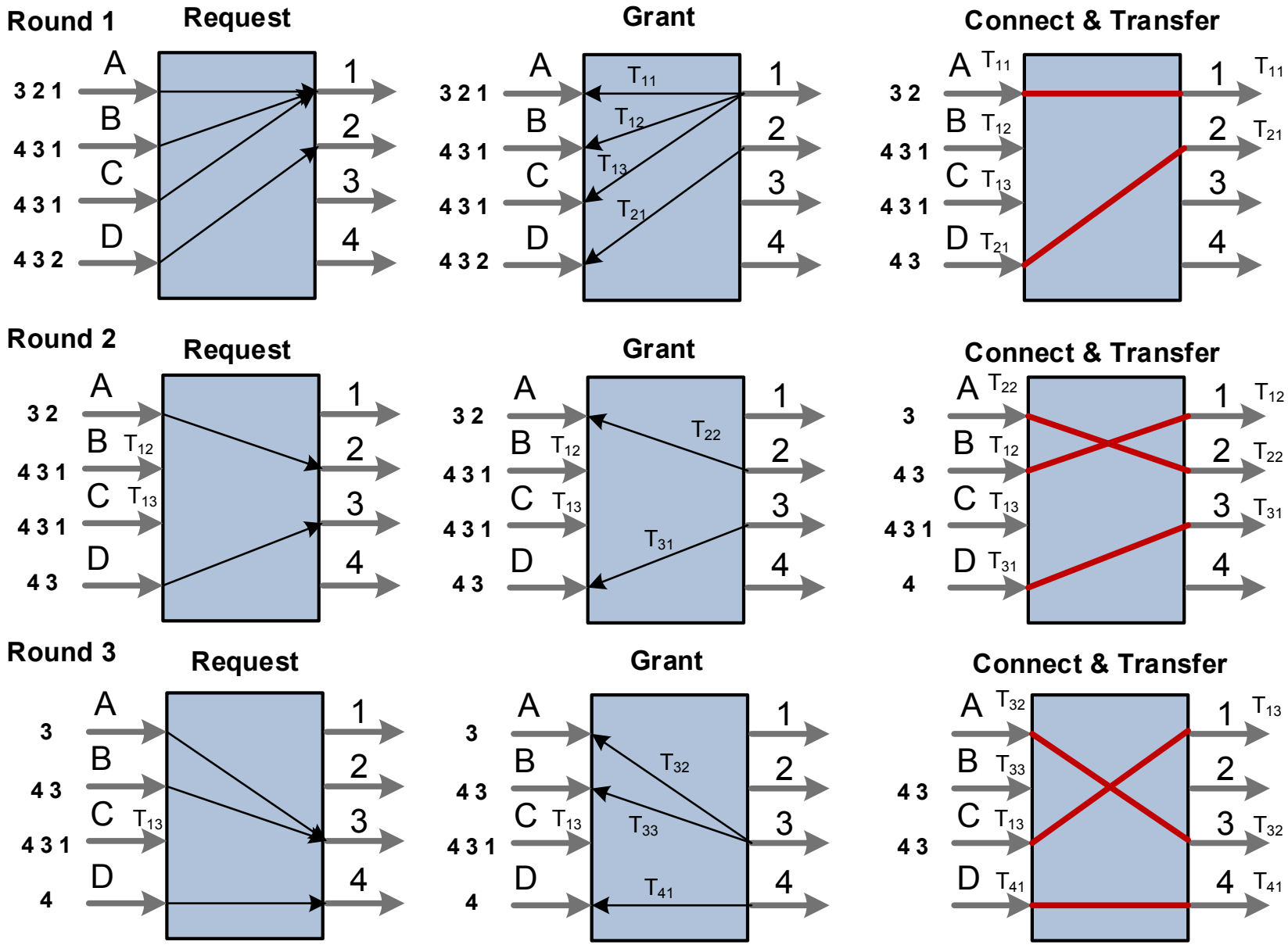
1. Žádost o lístek (Request)
2. Přidělení lístku (Grant)
3. Propojení vstupů a výstupů, přenos (Connect & Transfer)







Plánovací algoritmus přidělování lístků: Příklad



Plánovací algoritmus přidělování lístků: Příklad

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

- Přehled přenesených paketů

	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4	Time Slot 5	Time Slot 6
Input A	1	2	3			
Input B		1		3	4	
Input C			1		3	4
Input D	2	3	4			

Zhodnocení algoritmu

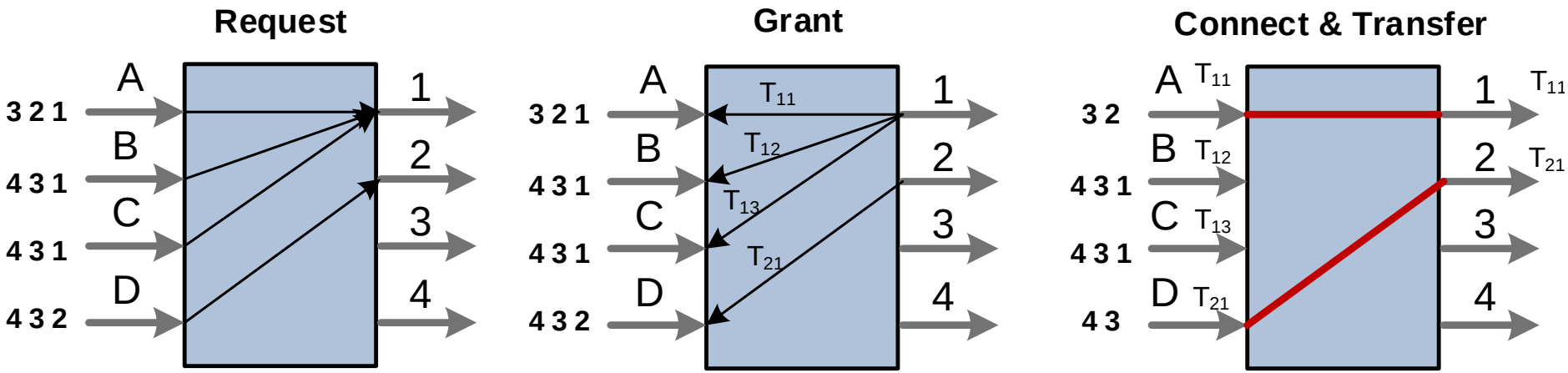
- Umožňuje asynchronní zpracování
- Přenos rámců proměnné délky
- Blokování na začátku fronty (HOL, Head of Line Blocking)
- Nezávislé přidělování lístků: problém u multicastu

Blokování na začátku fronty (HOL, Head of line)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Příčiny blokování

- Pakety na vstupních portech čekají ve frontě FIFO
- První buňka v pořadí blokuje další požadavky
- Blokované buňky čekají, i když je cílový port dostupný



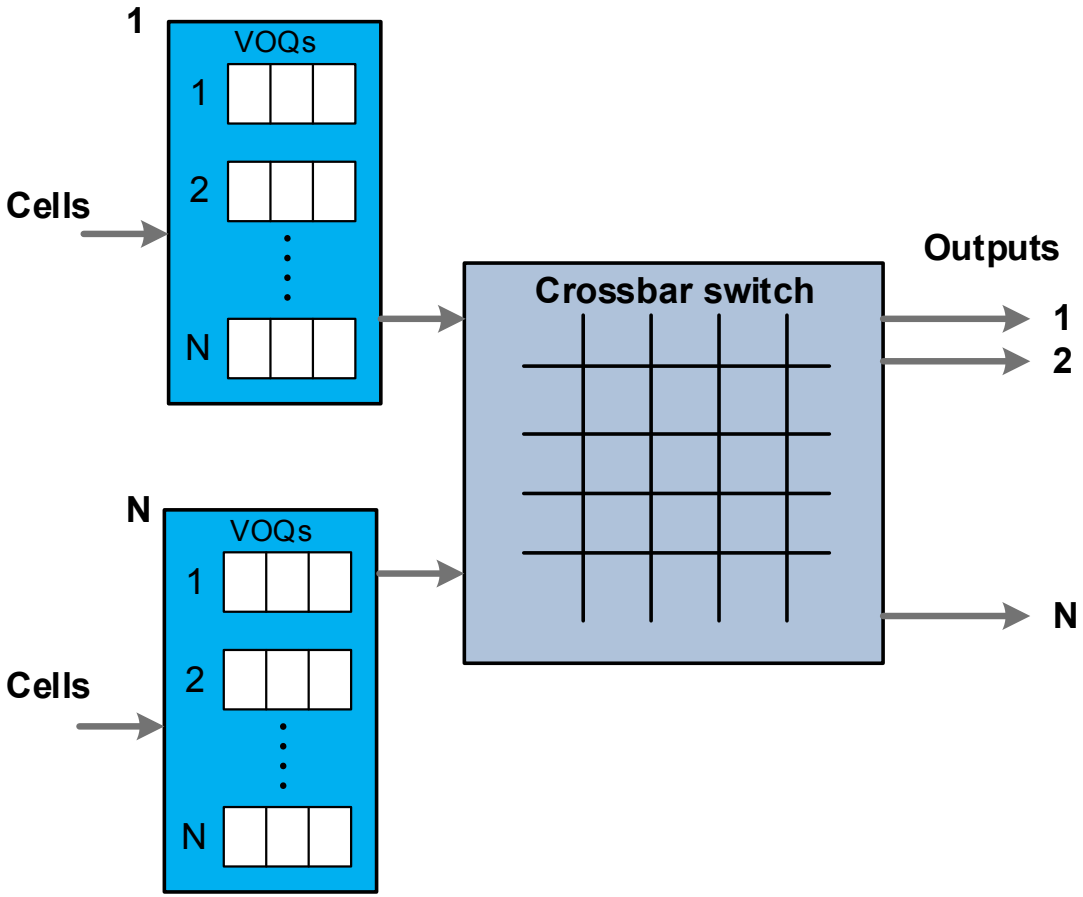
Co s tím?

- Rozdělení vstupní fronty na více virutálních front podle cílových portů

Blokování na začátku fronty (HOL, Head of line)

Virtuální výstupní fronty VOQ (Virtual Output Queues)

- Pro každý výstupní port vytvořena na daném vstupu jedna fronta
- Celkově potřebujeme N^2 front

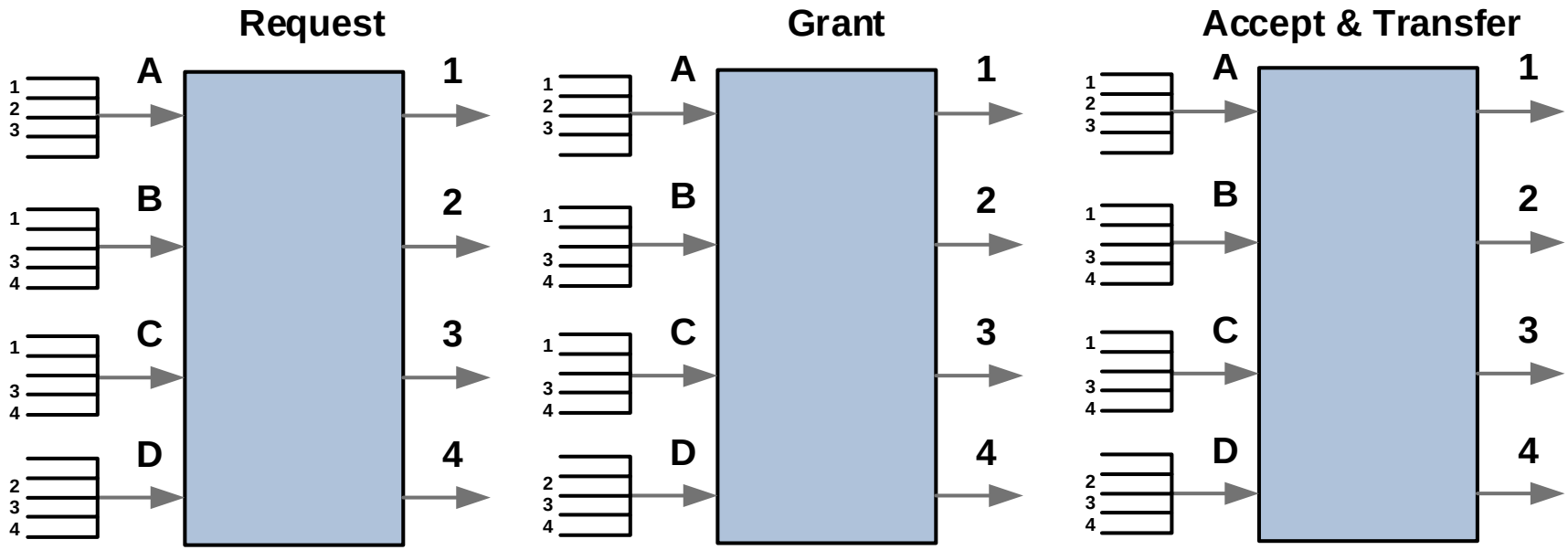


Algoritmus PIM (Parallel Iterative Matching)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Princip algoritmu

- Pracuje podobně jako algoritmus přidělování lístků, využívá virtuální fronty
- Hledá maximální párování pomocí *náhodné volby* požadavku ze vstupního portu
- Soutěžení o porty:
  - výstupní porty: více žádostí na jeden výstup
  - vstupní porty: více povolení pro data na jednom portu

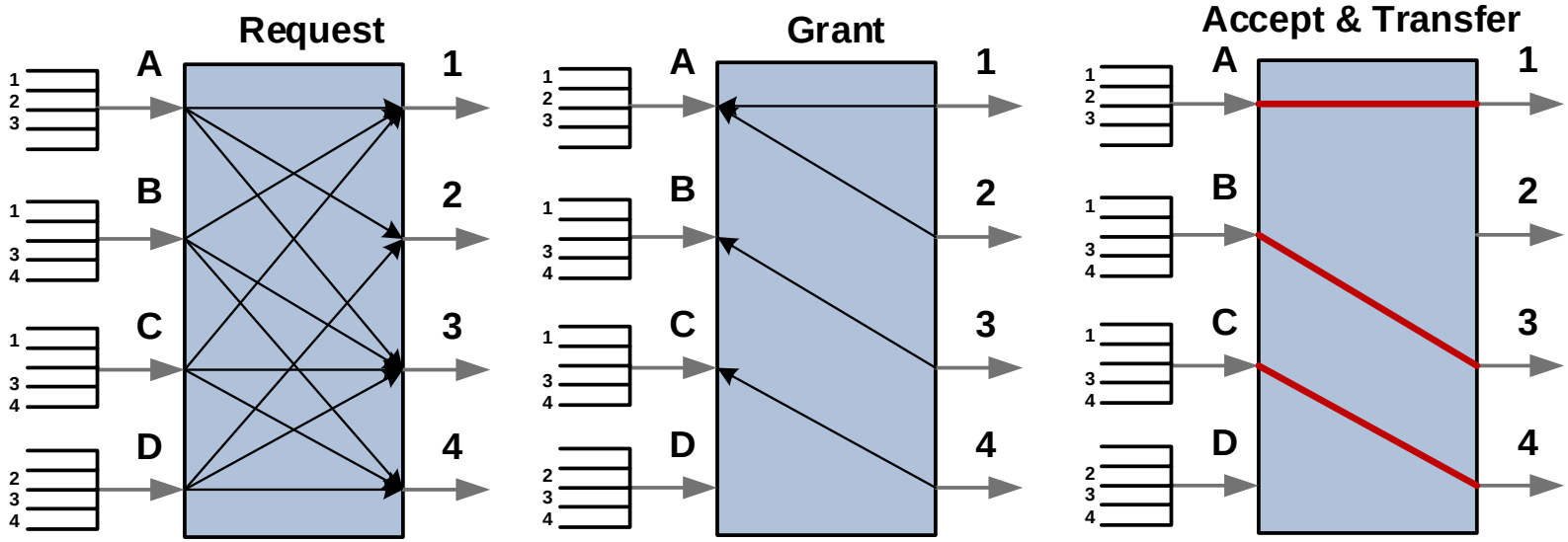


Algoritmus PIM (Parallel Iterative Matching)

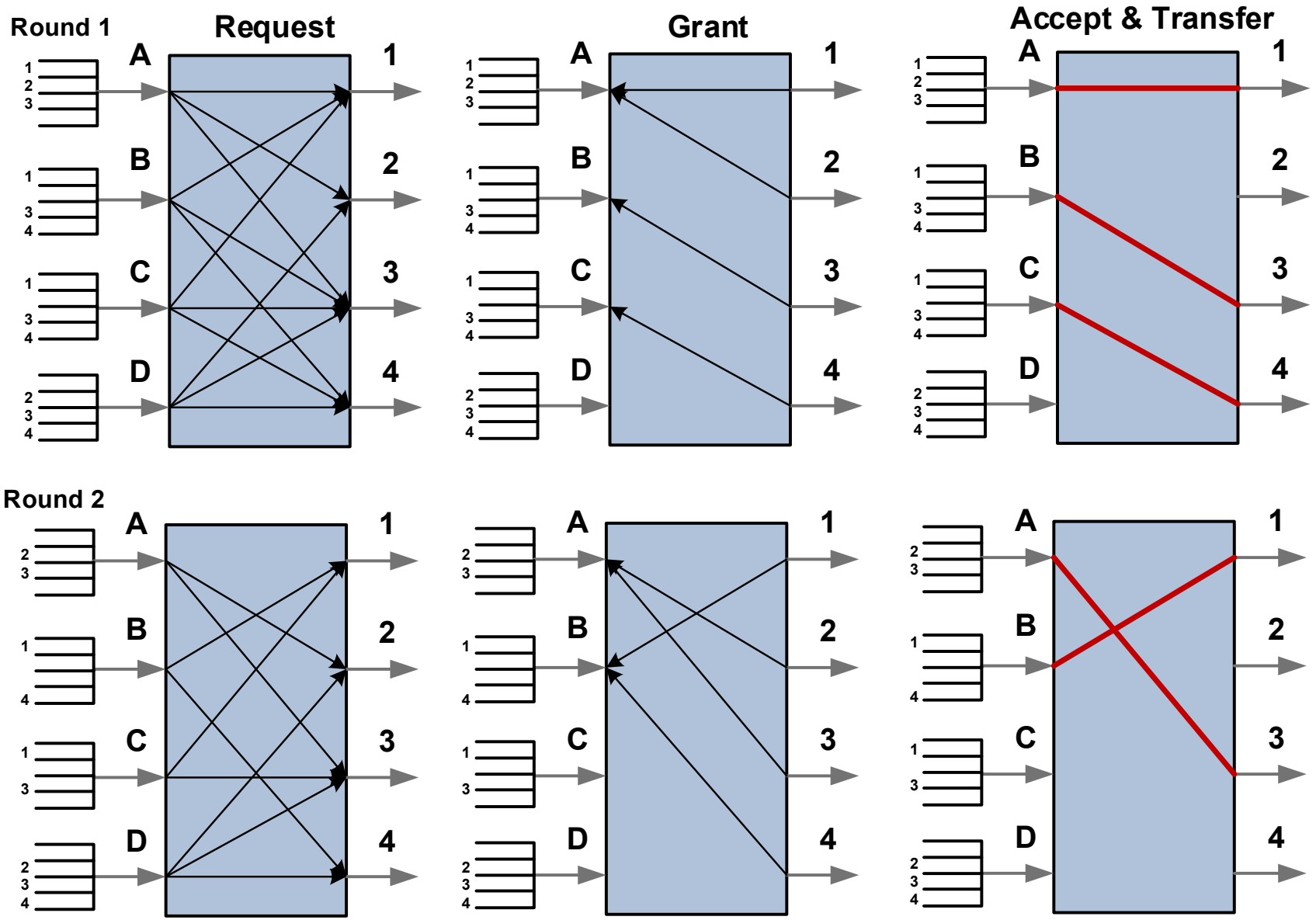
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Fáze algoritmu

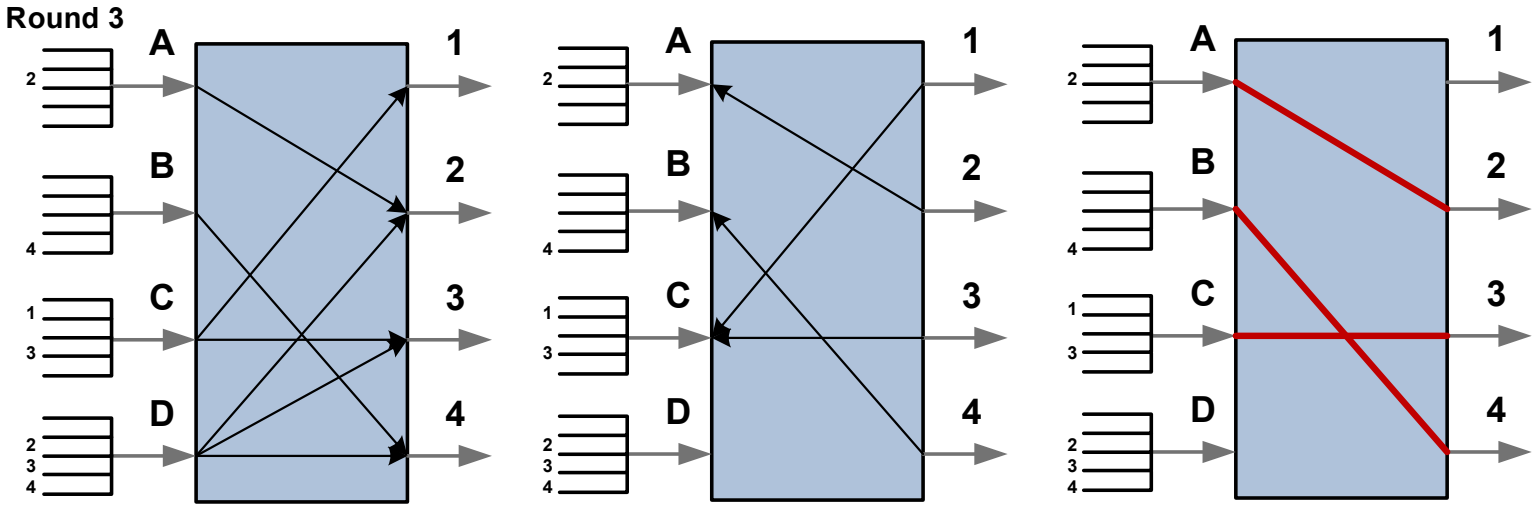
1. *Žádost o port (Request)*
   - Vstupní port P posílá žádosti ze všech front na všechny žádané porty.
2. *Udělení portu (Grant)*
   - Výstupní port Q přidělí právo přenosu.
   - V případě více žádostí o port Q, vybere *náhodně* jednu žádost.
3. *Přijetí a přenos (Accept & Transfer)*
   - Vstupní port P zpracuje udělená povolení k přenosu.
   - V případě udělení více povolení, *náhodně* vybere jedno. Dojde k přenosu dat.



Algoritmus PIM: Příklad



Algoritmus PIM: Příklad

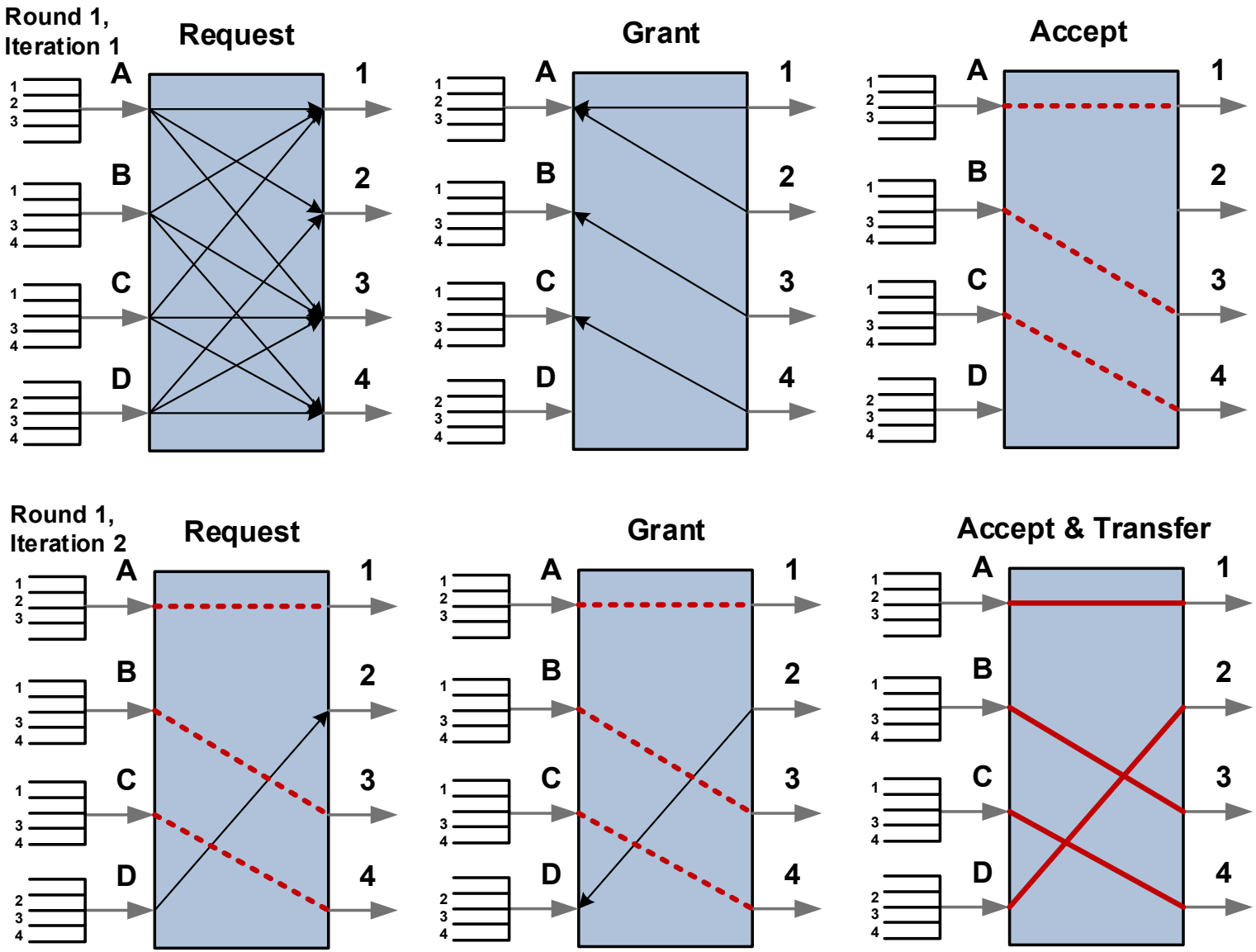


- Přehled přenesených paketů

	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4	Time Slot 5	Time Slot 6
Input A	1	3	2			
Input B	3	1	4			
Input C	4		3	1		
Input D				2	3	4

- 6 časových slotů, 4 porty $\Rightarrow$ 24 možností přenosu
- Použito 12 přenosů z 24 $\Rightarrow$ propustnost 50%

Algoritmus PIM: Optimalizace pomocí více iterací



Algoritmus PIM: Optimalizace pomocí více iterací

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Přehled přenesených paketů

	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4
Input A	1	3	2	
Input B	3	1	4	
Input C	4		3	1
Input D	2	4		3

- 4 časové sloty, 4 porty
  $\Rightarrow$ 16 možností přenosu
- Použito 12 přenosů z 16
  $\Rightarrow$ propustnost 75%
- Problém generování náhodných čísel

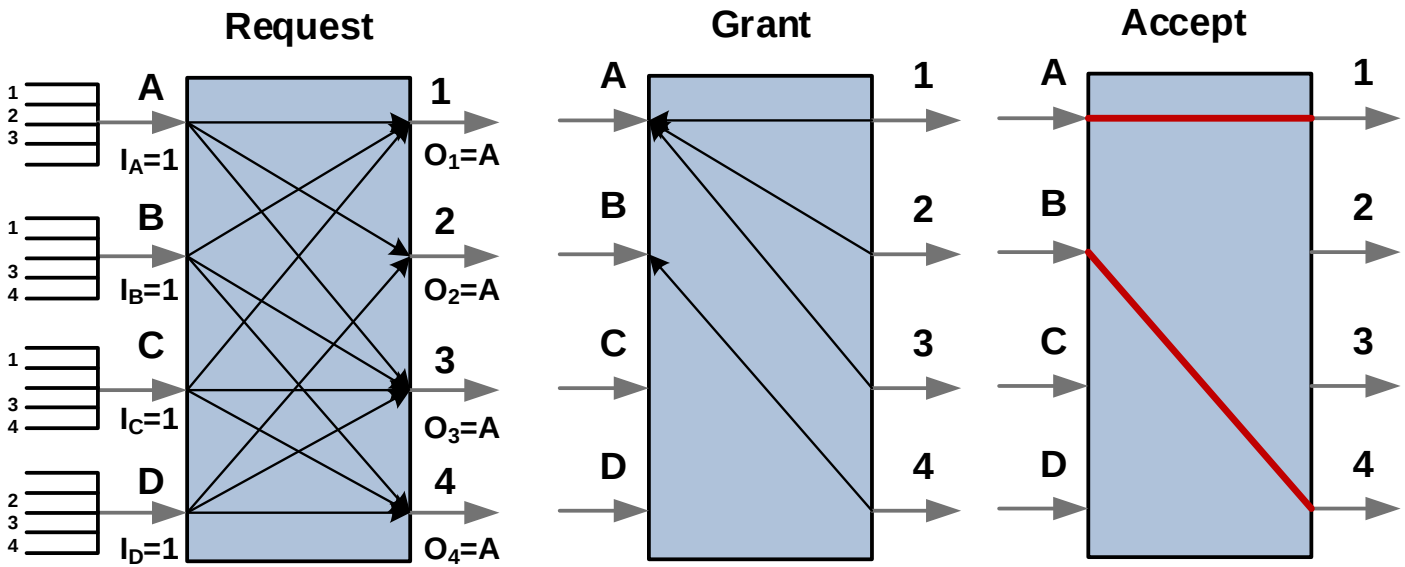
Zhodnocení algoritmu PIM

- *Úspěšnost nalezení největšího párování*
  - Nejhorší případ: N iterací (všechny vstupy na jeden výstup)
  - Nejlepší případ: 1 iterace (každý vstup na jiný výstup)
  - Průměr: $\mathcal{O}(log_2 N)$
- *Počet iterací vs. doba párování*
  - Obecně variabilní počet iterací, v praxi pevně dané
  - Experimentální výsledky pro velké N: jedna iterace: 63%, dvě iterace: 75%

Algoritmus iSLIP

Princip algoritmu

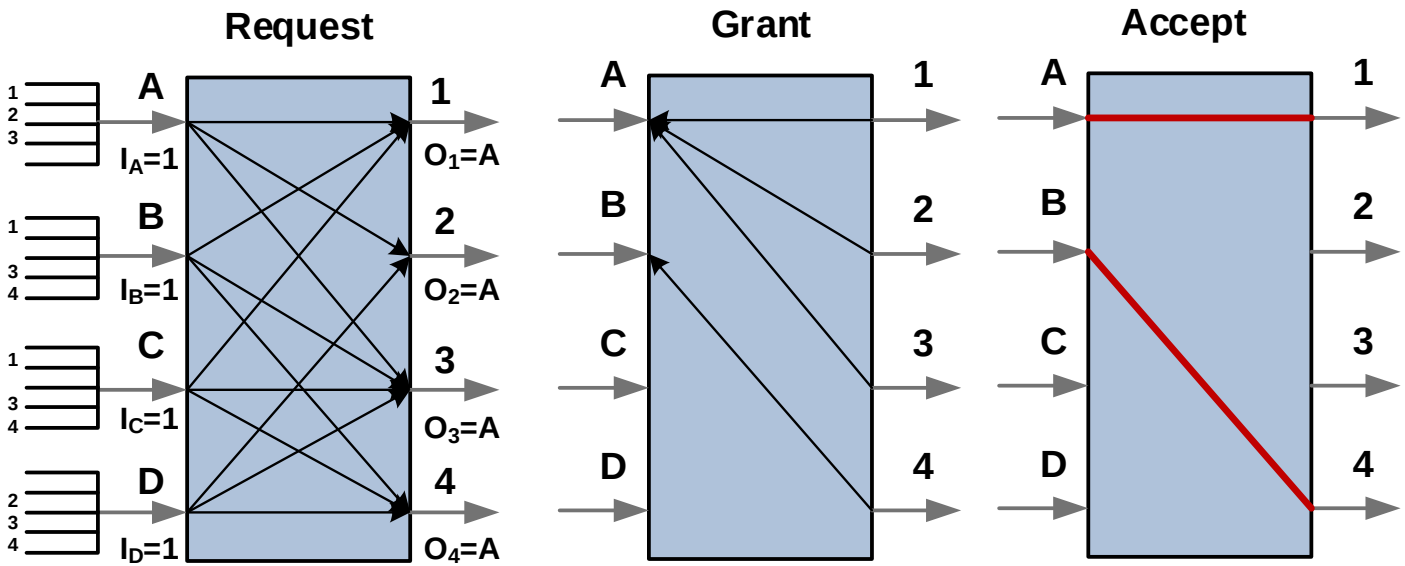
- Iterativní algoritmus pro hledání maximálního párování
- Při soupeření o port používá algoritmus *rotujících ukazatelů*:
  - I_i: ukazatel na vstupním portu i (přijetí žádosti)
  - O_j: ukazatel na výstupním portu j (udělení povolení k přenosu)
- Přidělení probíhá iterativně. Po každé iteraci se ukazatele inkrementují.
  - $I_i' = (I_i + 1)\ mod\ N$
  - $O_j' = (O_j + 1)\ mod\ N$
  - Ukazatele se inkrementují, až když je žádost o port potvrzena.



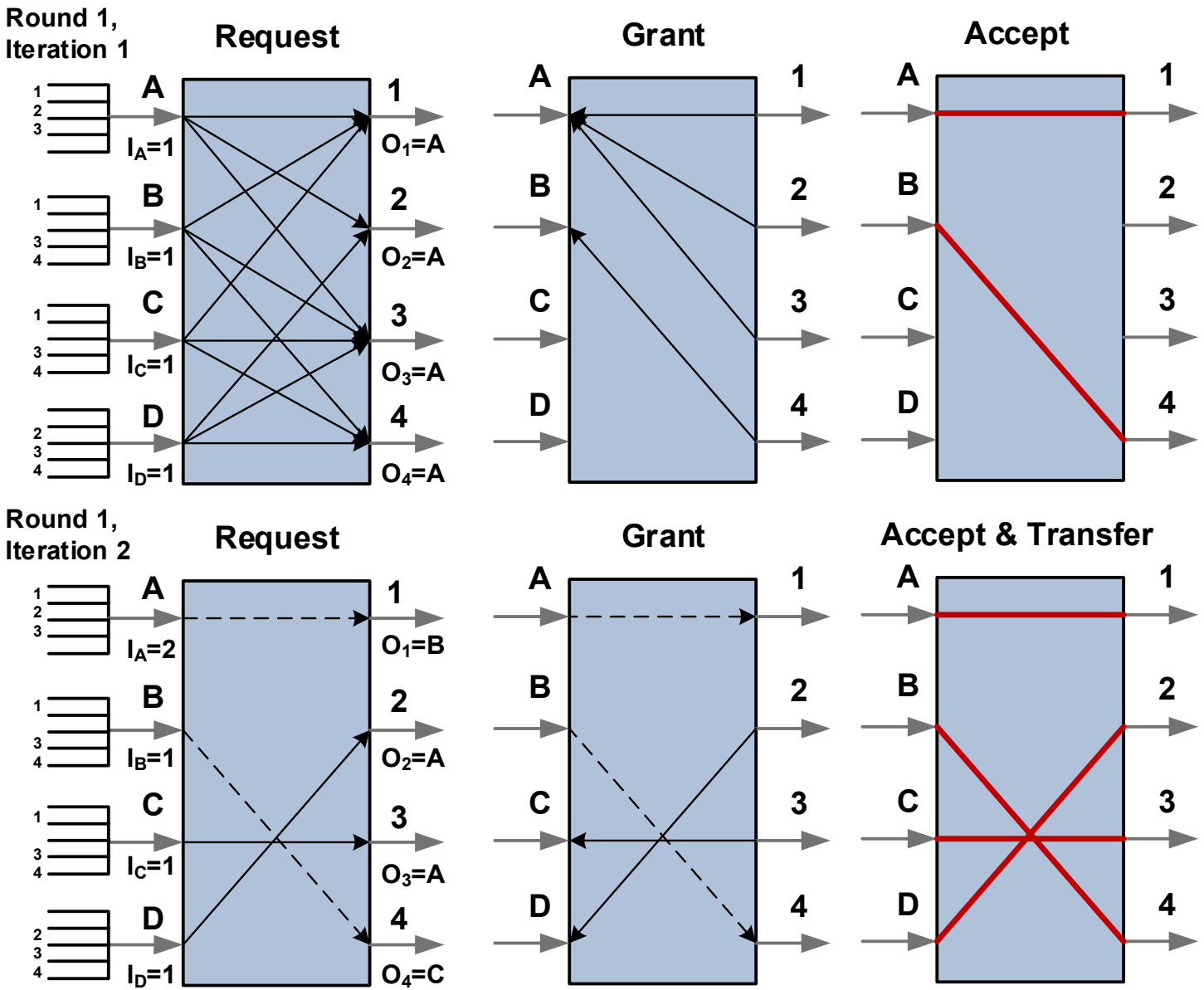
Algoritmus iSLIP

Fáze algoritmu

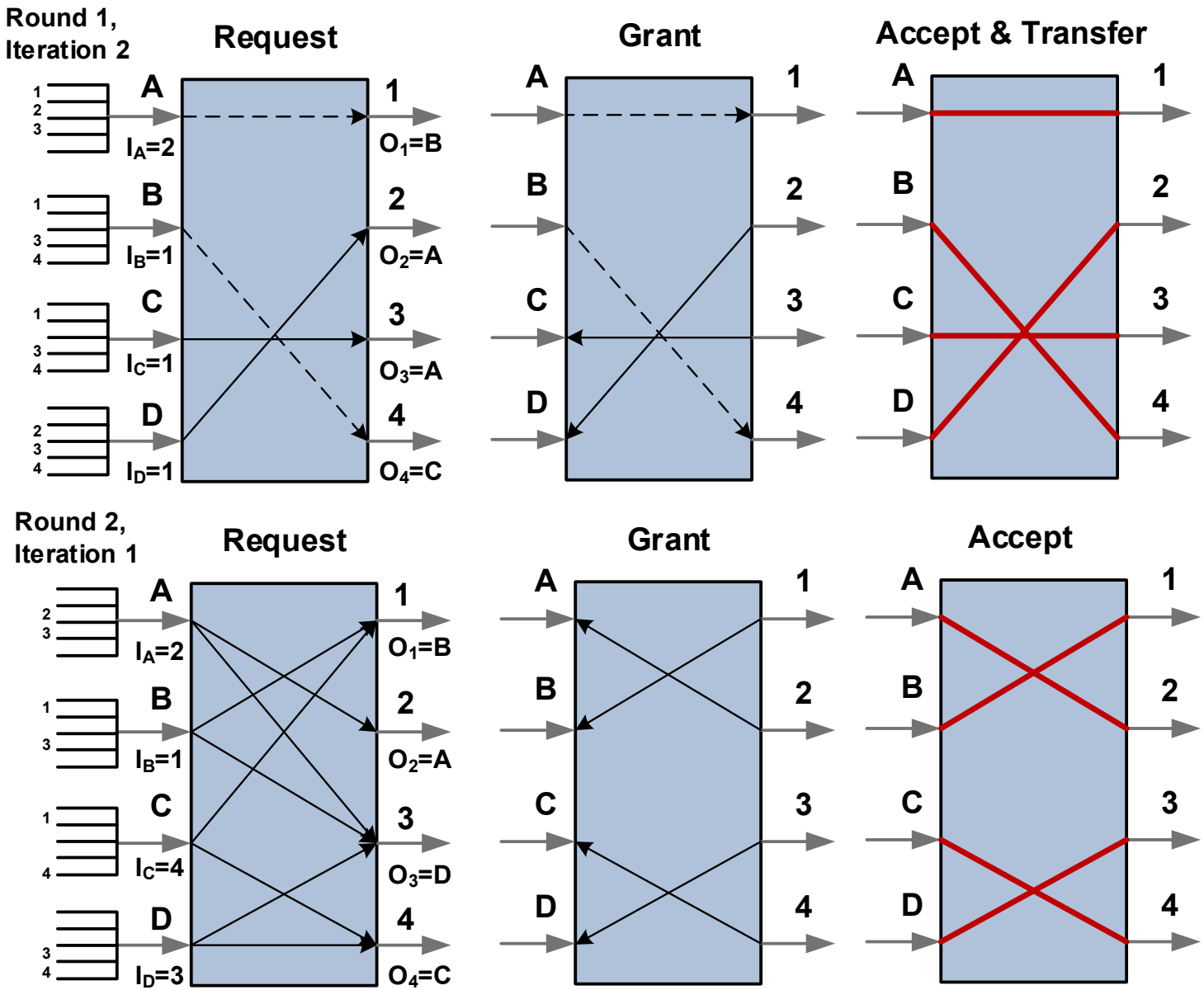
1. *Žádost o port (Request)*
   - Každý port pošle žádost na každý výstupní port
2. *Udělení portu (Grant)*
   - Výstupní port Y vybere žádost, která má číslo portu větší nebo rovno ukazateli O_j; v případě více žádostí se vybere žádost s nejmenší hodnotou
3. *Přijetí volby (Accept)*
   - Vstupní port X vybere povolení od portu, které je větší nebo rovno ukazateli I_j; v případě více povolení se vybere povolení s nejmenší hodnotou



Algoritmus iSLIP: Příklad



Algoritmus iSLIP: Příklad



Algoritmus iSLIP: Příklad

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

- Přehled přenesených paketů

	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4
Input A	1	2	3	
Input B	4	1		3
Input C	3	4	1	
Input D	2	3	4	

Zhodnocení algoritmu iSLIP

- 4 časové sloty, 4 porty $\Rightarrow$ 16 možností přenosu
- Použito 12 přenosů z 16 $\Rightarrow$ propustnost 75%
- Deterministická obsluha (vs. náhodná u PIM)

Vícestupňové přepínání

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Vlastnosti jednostupňového přepínání

- Využívá se převážně křížového přepínače (crossbar)
- Problémy s kvadratickým nárůstem portů
- Vnitřní blokování, blokování HOL
- Součástí přepínání je hledání maximálního párování

Je možné zvětšit počet portů, aniž by se dramaticky rozšířilo přepínací pole?

Ano $\Rightarrow$ s použitím vícestupňového přepínání.

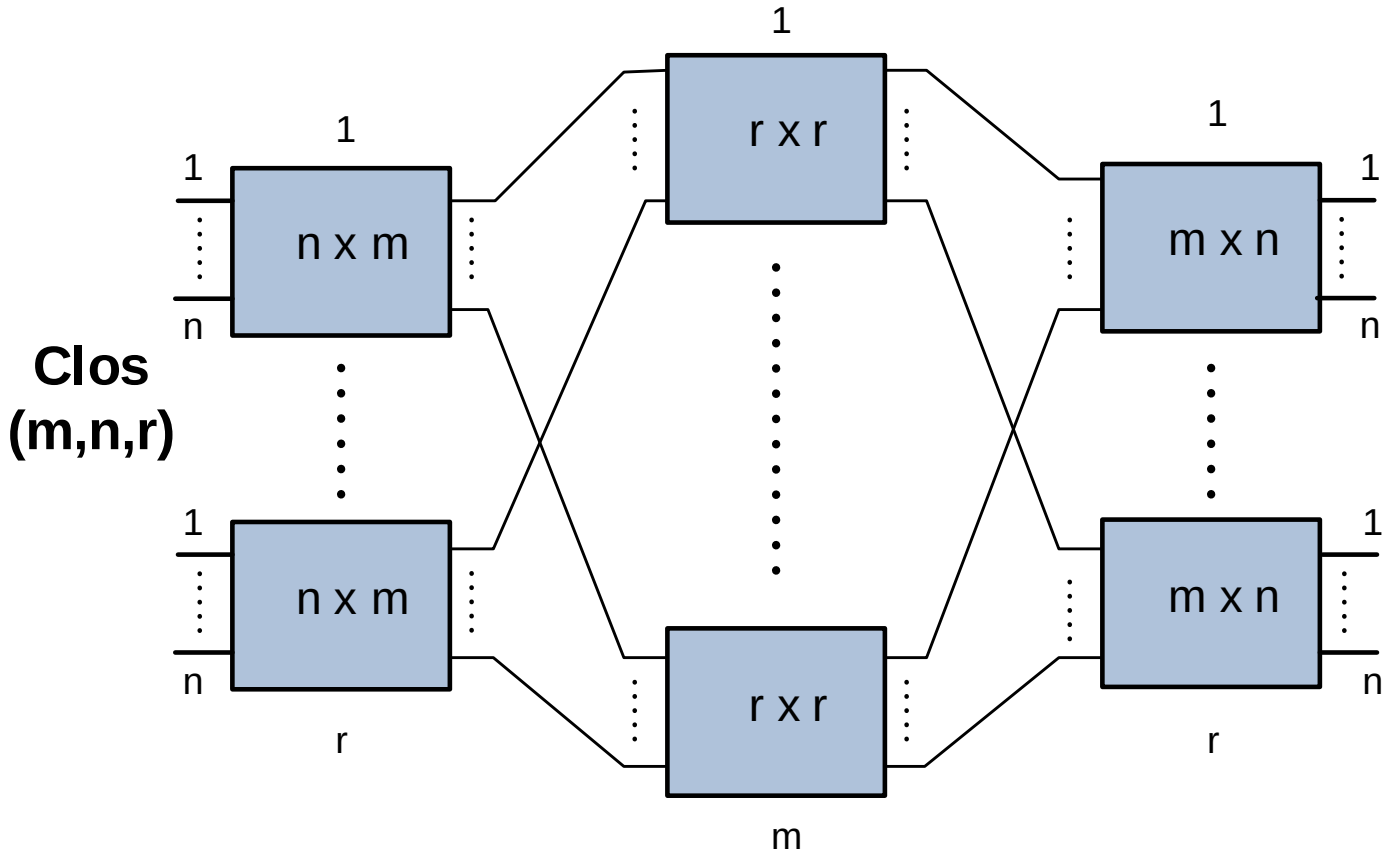
Vlastnosti vícestupňového přepínání

- Sítě typu Clos a Beneš
  - *vstup* $\rightarrow$ *vnitřní přepínací obvody* $\rightarrow$ *výstup*
- Vícestupňového obvody nemají vnitřního blokování.
- Součástí přepínání je vytvoření bezkonfliktního propojení.

Přepínací síť Clos(m,n,r)

Třístupňová přepínací síť Clos(m,n,r)

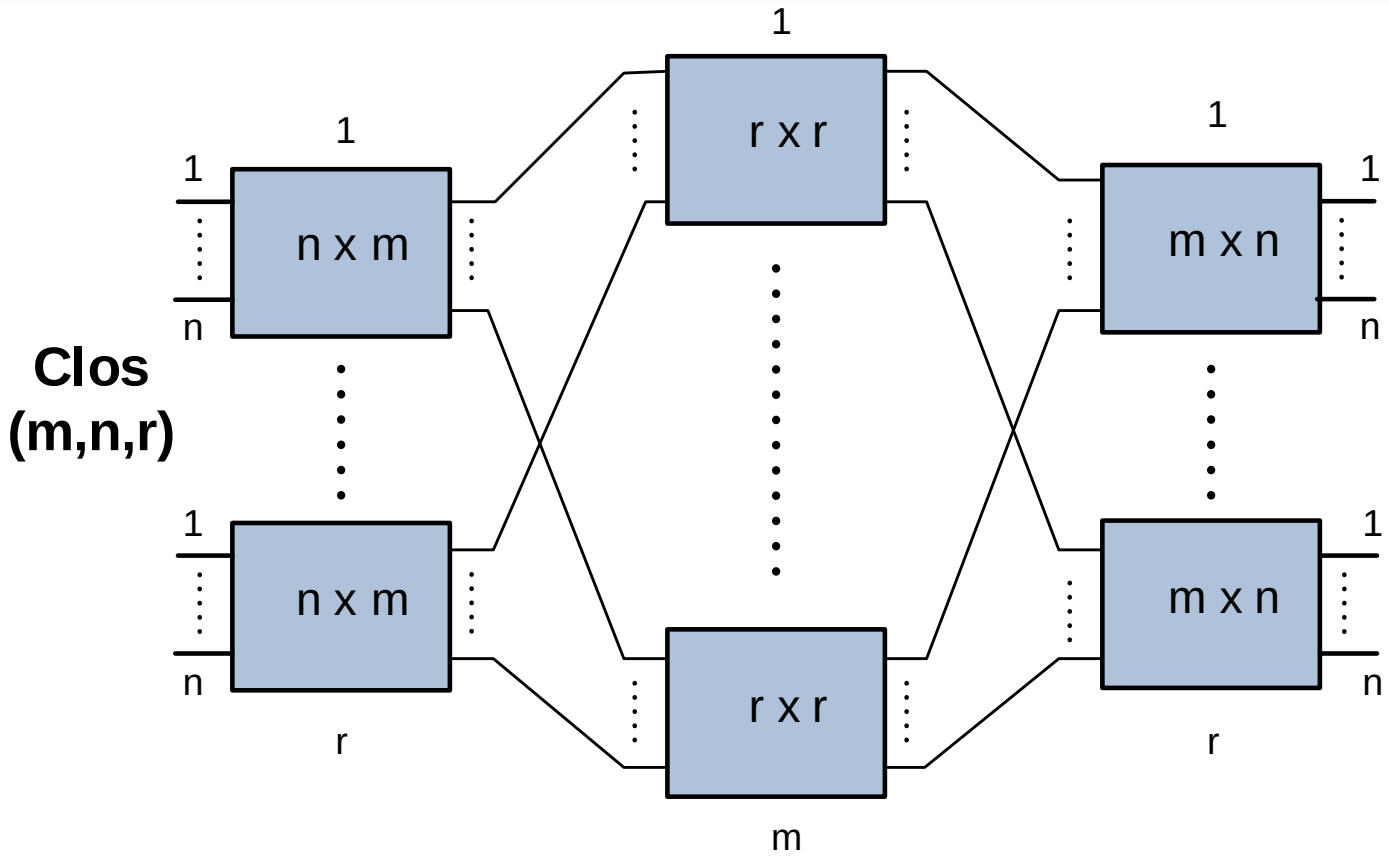
- r vstupních křížových přepínačů s n vstupy $\Rightarrow$ přepínač má $N = n \times r$ portů
- m vnitřních křížových přepínačů s r vstupy $\Rightarrow$ propojují každý V/V blok
- Mezi každým vstupním a výstupním portem existuje m různých cest.



Přepínací síť Clos(m,n,r)

Closův teorém (neblokující síť)

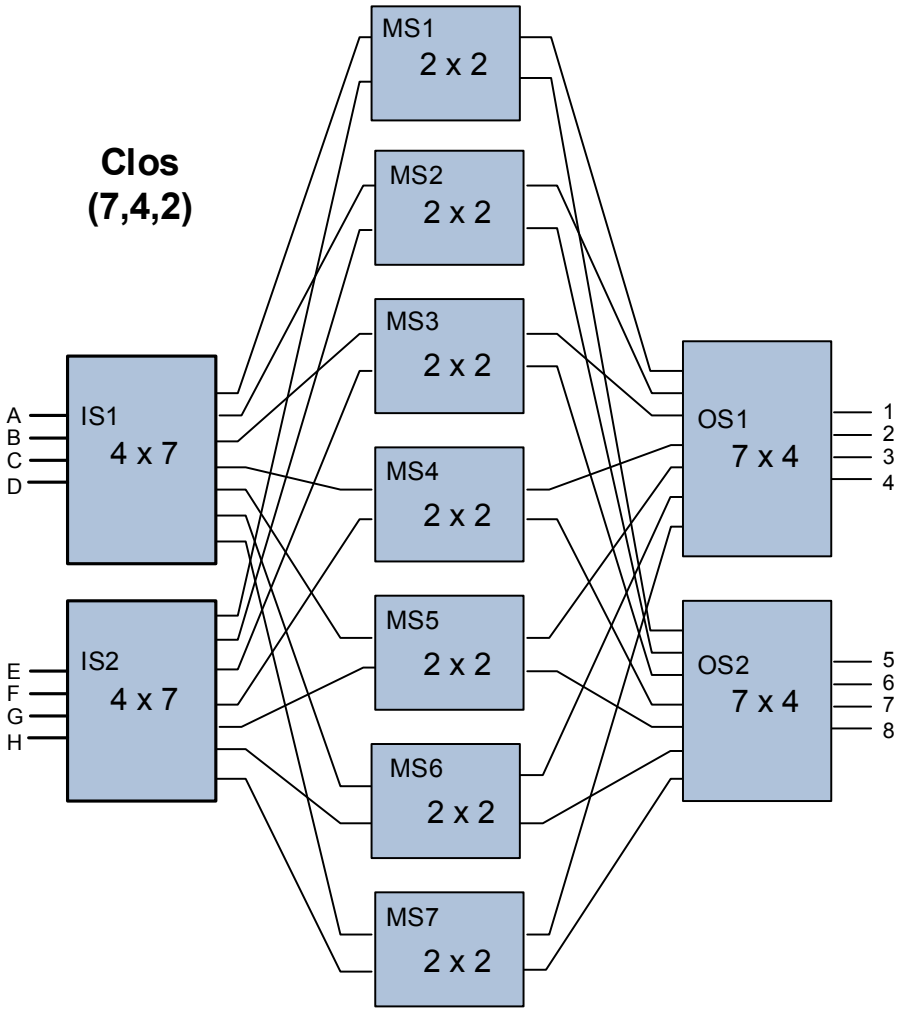
- *Pokud $m \geq 2n-1$, pak lze přidat nové propojení vstupu a výstupu bez přeskládání.*
- Tehdy je přepínač pro jakoukoliv konfiguraci vstupů a výstupů neblokující.
- Nevýhoda $\Rightarrow$ síť vyžaduje velký počet vnitřních bloků.

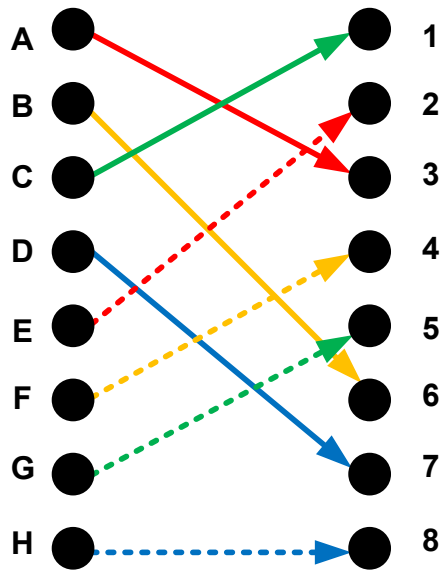


Přepínací síť Clos(m,n,r)

Příklad neblokující sítě Clos(7,4,2): osmiportový přepínač

- Platí Closův teorém, tj. $m \geq 2n - 1$





Přepínací síť Clos(m,n,r): Vlastnosti

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Vlastnosti sítě Clos(m,n,r)

- Přepínač s N vstupními porty potřebuje síť $(m, n, \lceil N/n \rceil)$.
- Pokud $n = \sqrt{N/2}$, pak minimální počet propojení je $5.76 \times N \times \sqrt{N}$.
- Počet propojení je stále lepší než křížový přepínač, tj. než N^2.

Neblokující chování

- Pro neblokující chování musí platit Closův teorém, tj. $m \geq 2n - 1$.
- Například přepínač o 256 portech vyžaduje síť (31,16,16) $\rightarrow$ drahé.

Modifikované řešení: síť Clos(m,n,r), kde $m \geq n$

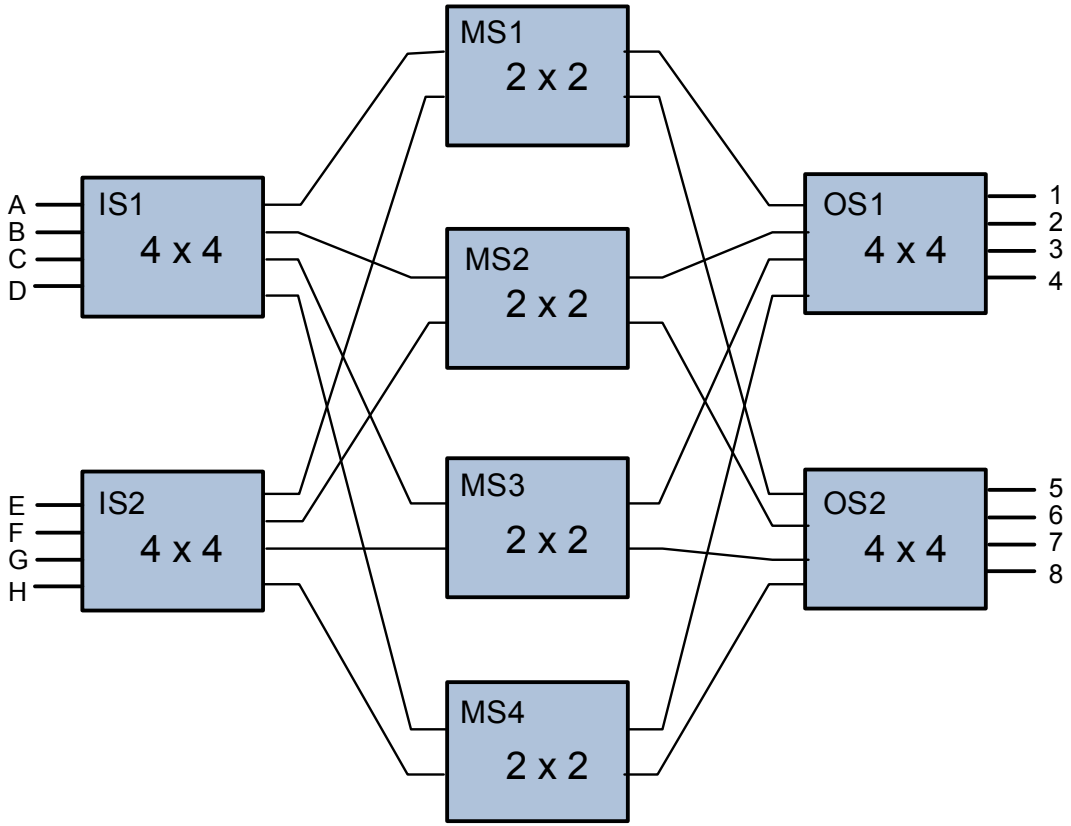
- Redukován počet prostředních přepínačů $\rightarrow$ levnější.
- Síť není neblokující pro libovolnou konfiguraci vstupů. a výstupů.
- Vlastnost modifikované sítě: *síť je neblokující po přeskládání*.
  - Výpočet přeskládání musí být rychlejší než přenos dat.
  - Používají se algoritmy barvení hran v bipartitním multigrafu.

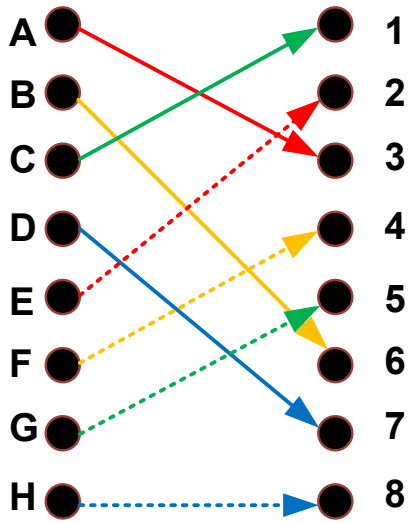
Modifikovaná přepínací síť Clos(m,n,r)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Příklad sítě Clos(4,4,2): osmiportový přepínač

- Síť Clos(4,4,2) je neblokující po přeskládání, protože $m \geq n$.
- Výpočetní složitost přeskládání je $\mathcal{O}(N.log\ D)$, kde D je počet barev.

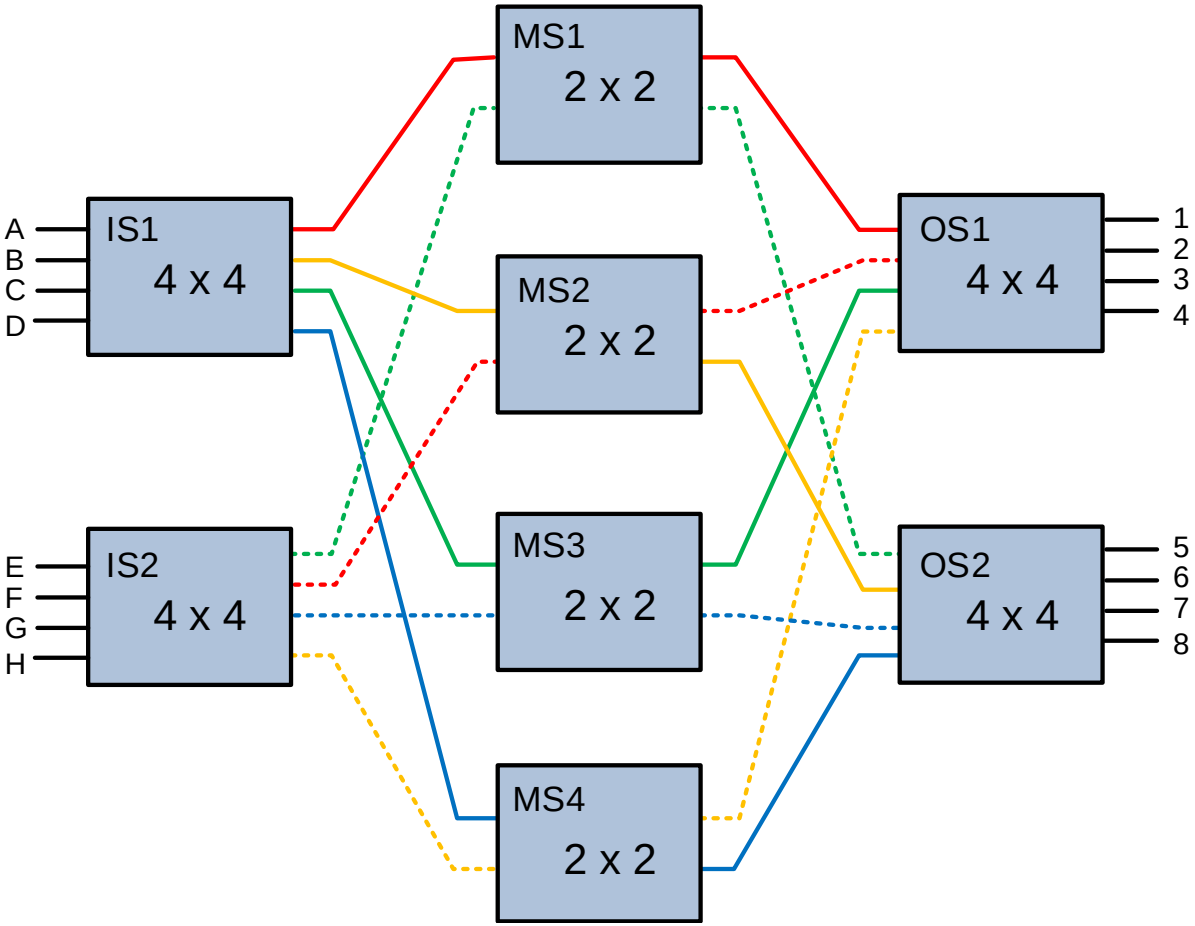


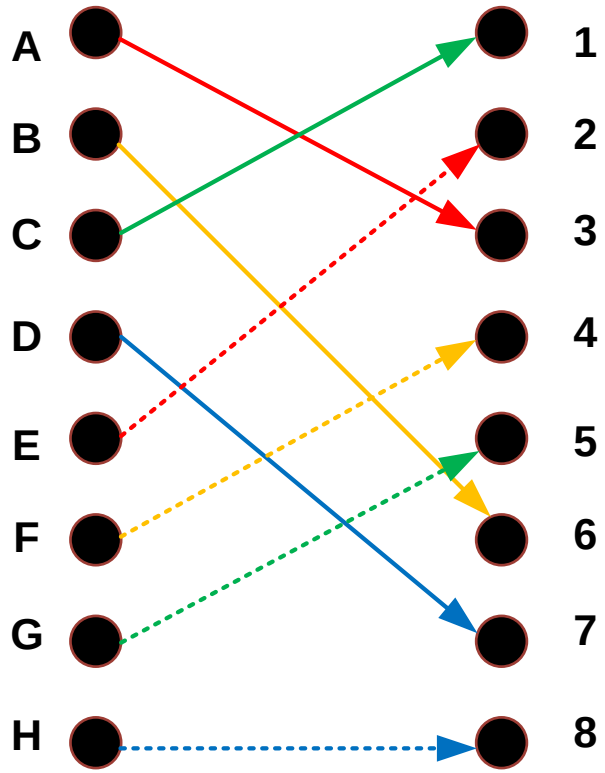


Modifikovaná přepínací síť Clos(m,n,r)

Příklad sítě Clos(4,4,2): osmiportový přepínač

- Příklad neblokující konfigurace po přeskládání.



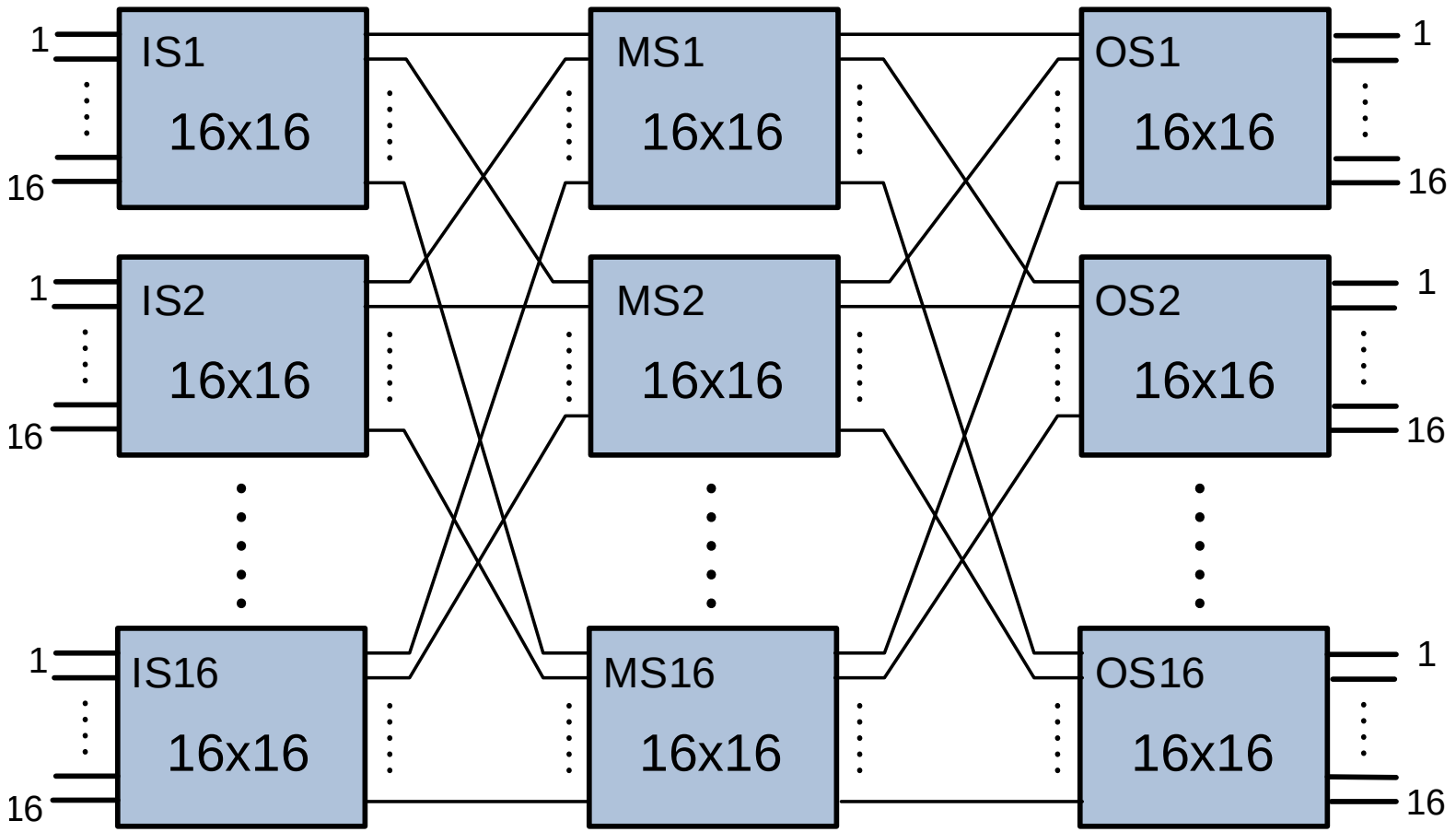


Modifikovaná přepínací síť Clos(m,n,r)

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Použití sítě Clos(16,16,16) v přepínači Juniper, T-series

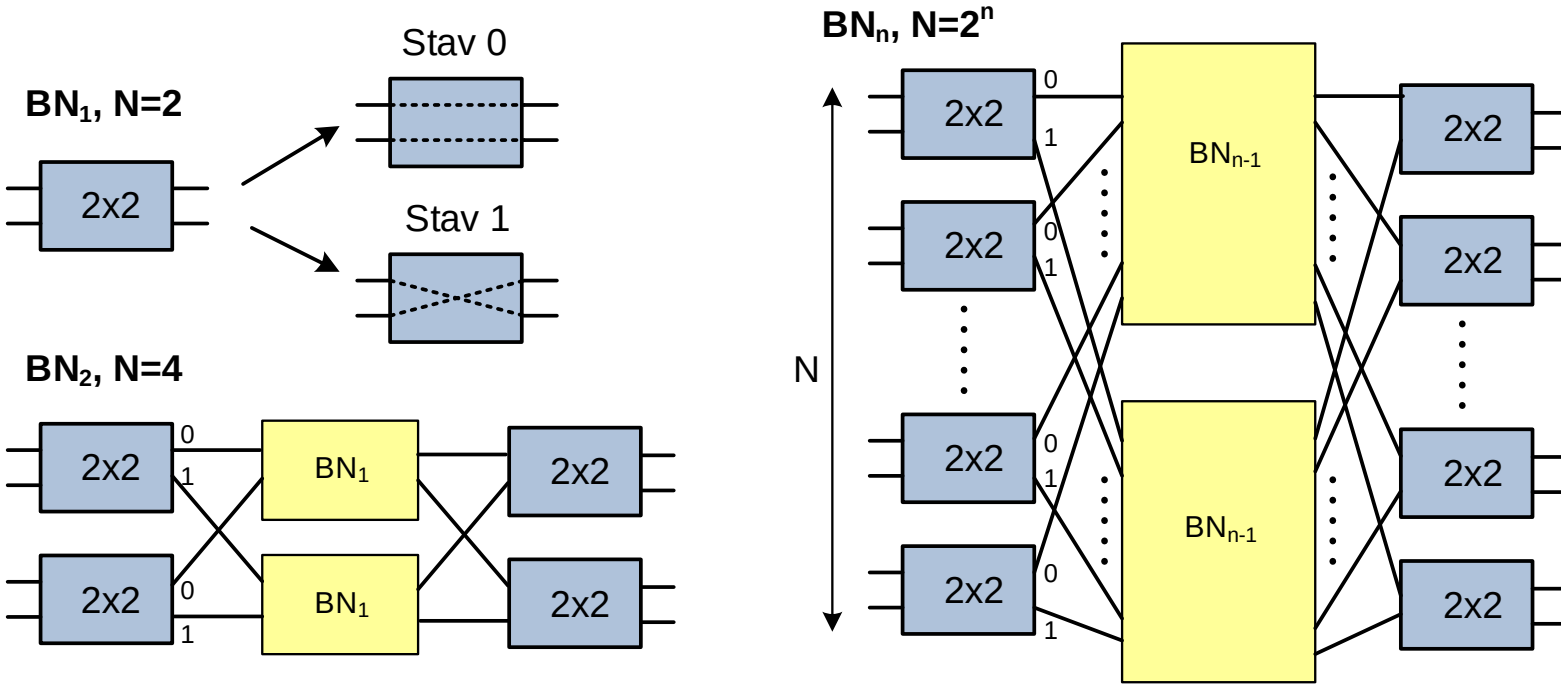
- Třístupňový přepínací modul o velikosti 256 portů (celkem 48 přepínacích bloků)
- Využito u přepínačů typu Juniper či Cisco ASR 9000



Přepínací síť Beneš BN_n

Rekurzivní přepínací síť Beneš BN_n

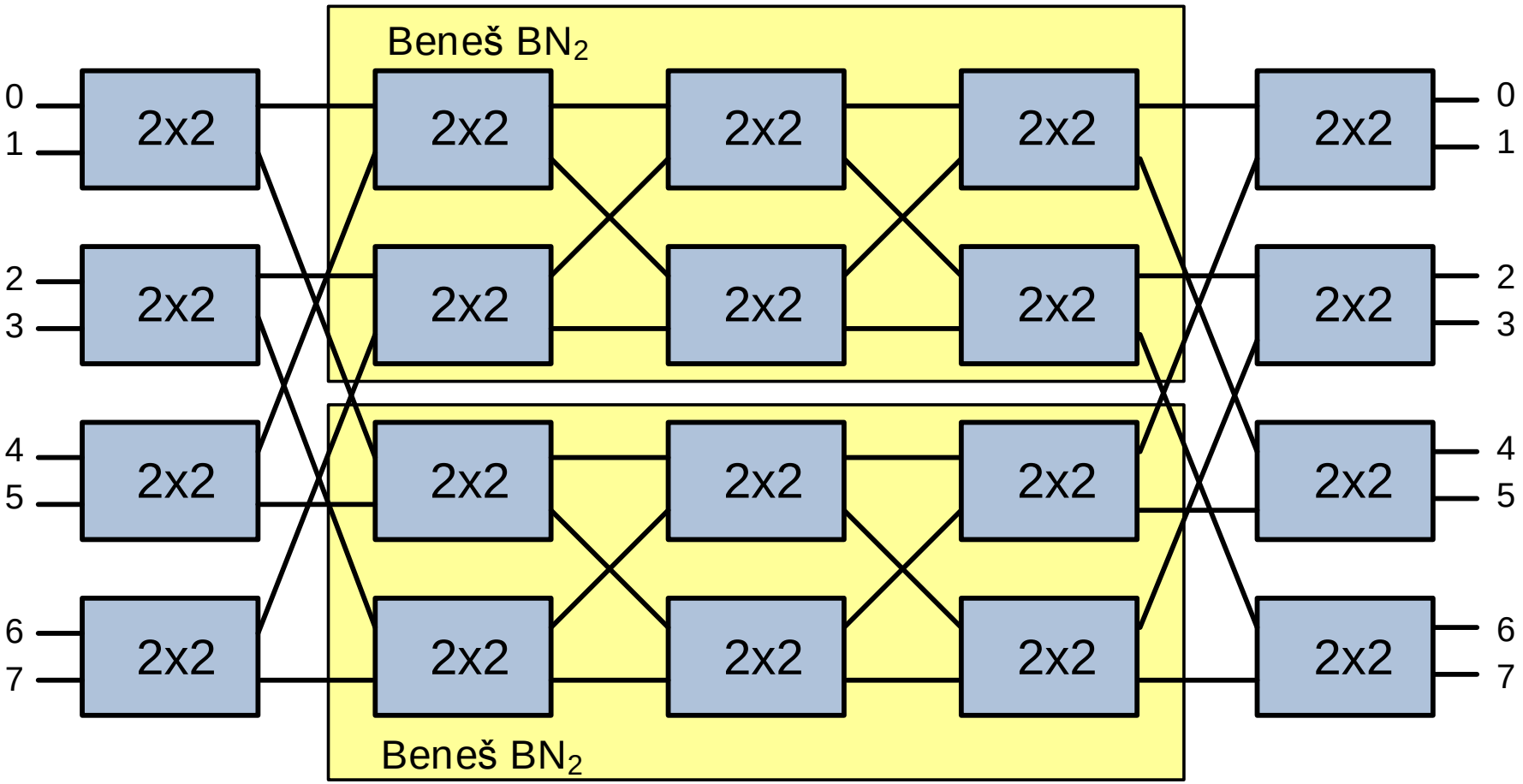
- Základem je modifikovaná síť Clos(2,2,1) s přeskládáním
- *Rekurzivní konstrukce sítě* BN_n
  - Počet vstupů (portů): $N = 2^n$
  - Blok $N/2$-vstupních a $N/2$-výstupních přepínačů 2×2
  - Prostřední část rekurzivních bloků $BN_{(n-1)}$
  - Počet bloků (stupňů): $2.\log_2(N) - 1$



Přepínací síť Beneš BN_n

Příklad: přepínací síť Beneš BN_3

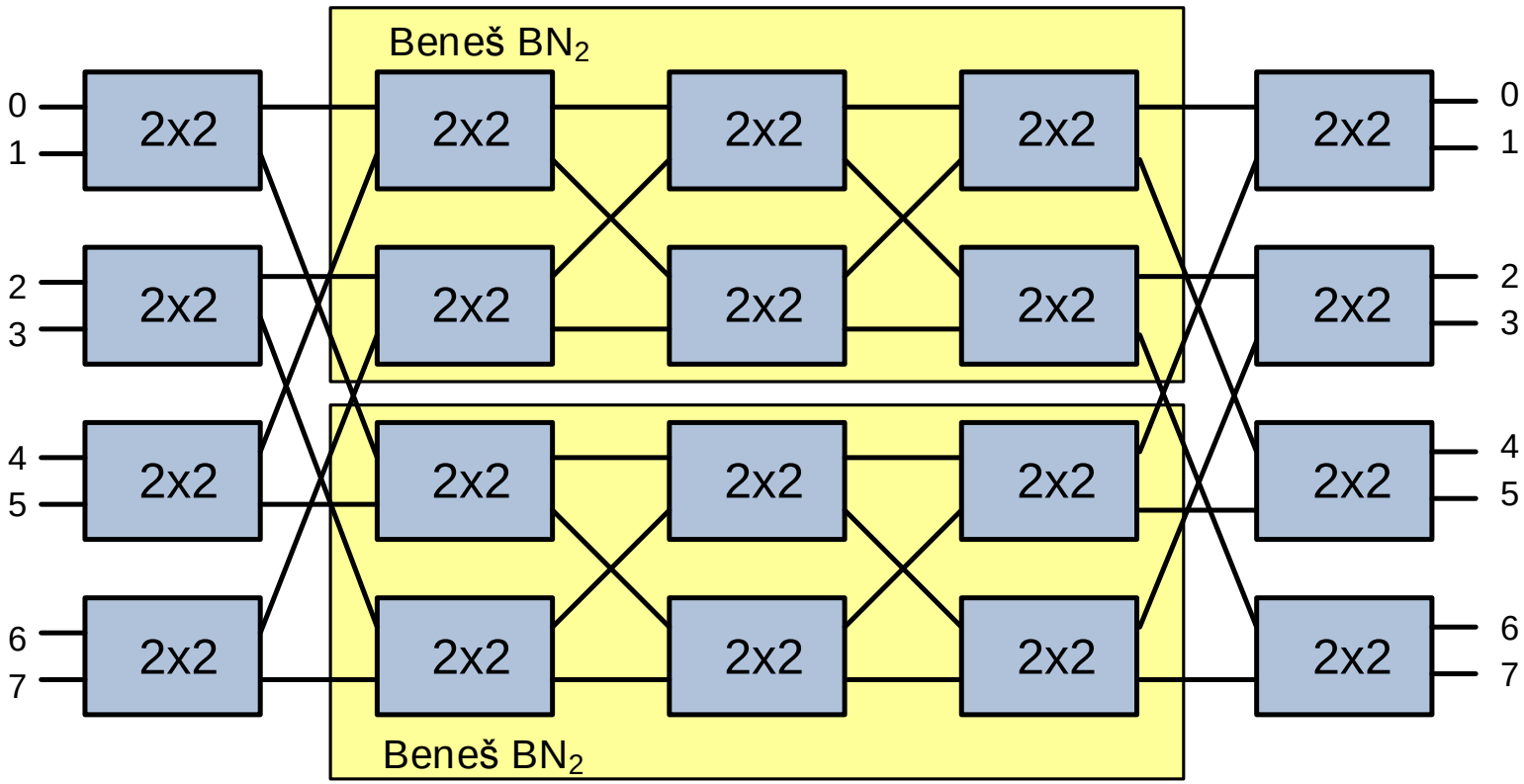
- N = 8 vstupů, 5 stupňů
- Použito např. u směrovače Cisco CSR-1.

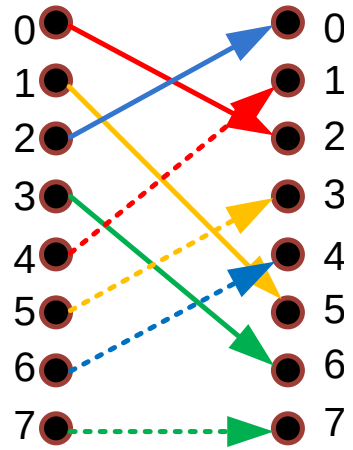


Přepínací síť Beneš BN_n

Propojování sítě

- Hledáme propojení – počet propojení je N!
- Časová složitost algoritmů je $\mathcal{O}(\sqrt{N})$ kroků.
- Používá se jednoduchý algoritmus využívající hierarchii sítě.

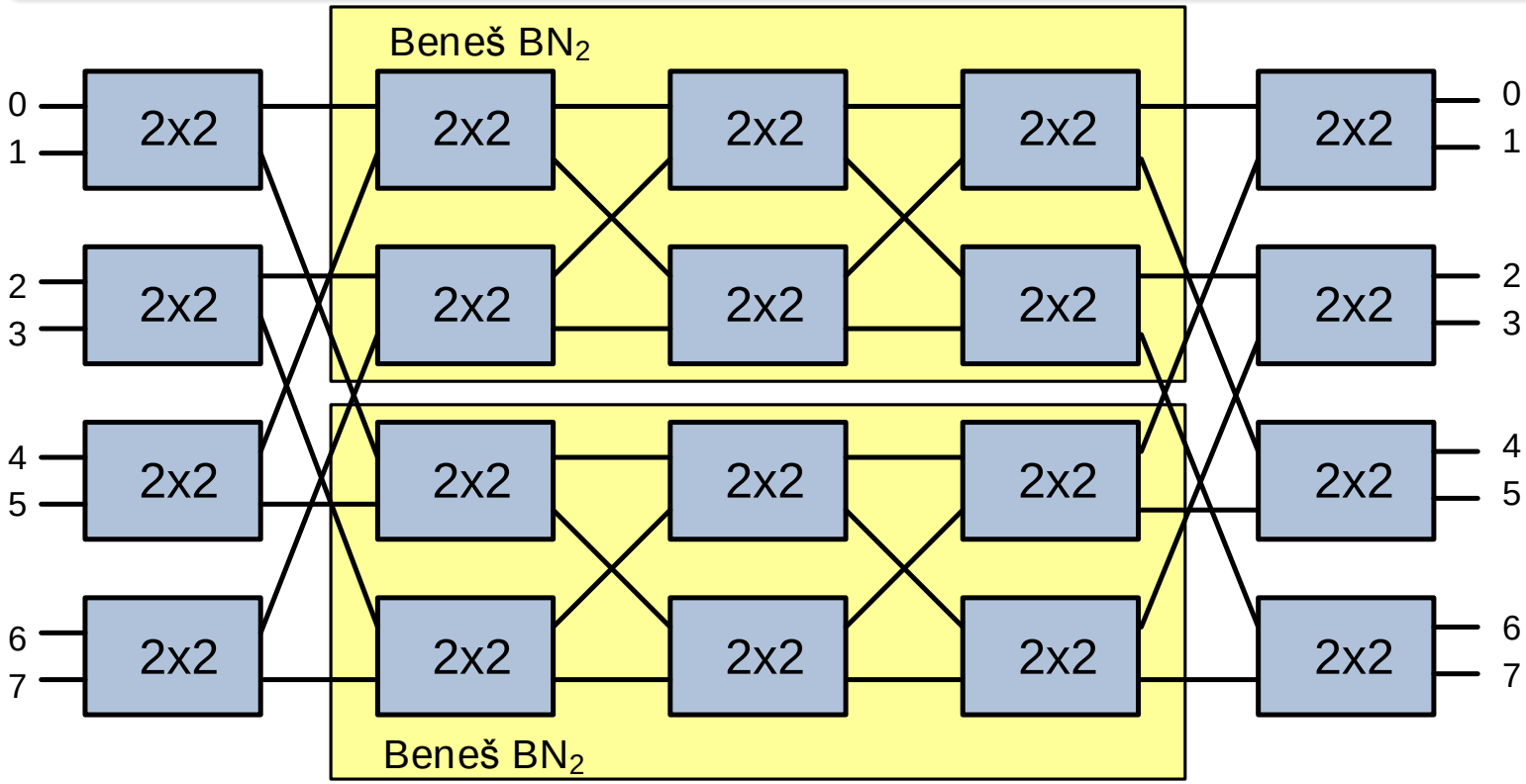


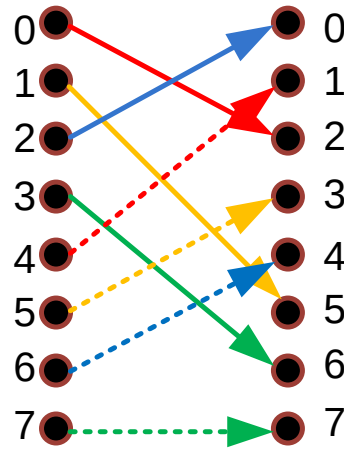


Přepínací síť Beneš BN_n

Algoritmus hledání propojení

- Horní podsíť BN_i slouží pro dopředné směrování, spodní podsíť BN_i pro zpětné.
- Postupuje se od vstupů k výstupům a zpět, vždy pro sousední porty.
- Nejprve se propojí vstupní a výstupní bloky, poté rekurzivně síť BN_{i-1}.

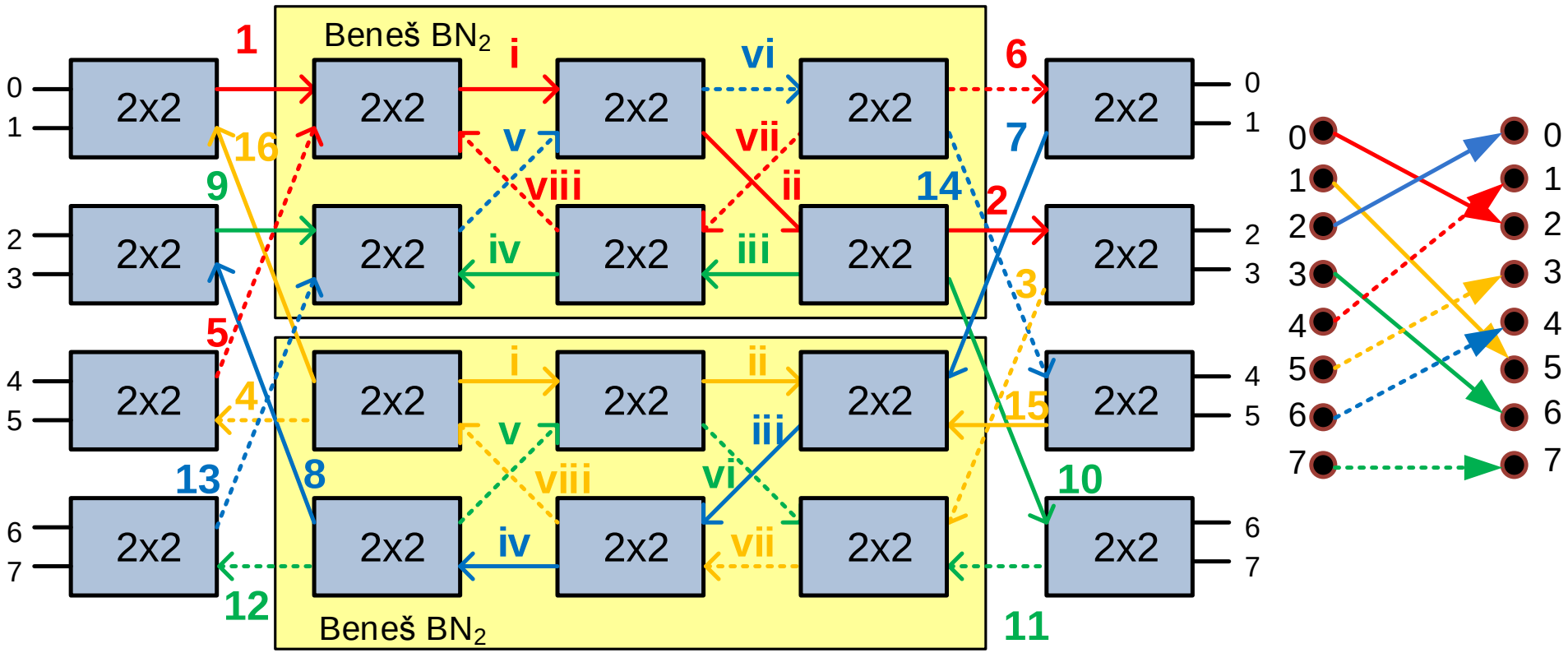




Přepínací síť Beneš BN_n

Algoritmus hledání propojení

- Horní podsíť BN_i slouží pro dopředné směrování, spodní podsíť BN_i pro zpětné.
- Postupuje se od vstupů k výstupům a zpět, vždy pro sousední porty.
- Nejprve se propojí vstupní a výstupní bloky, poté rekurzivně síť BN_{i-1}.



Otázky a úkoly k opakování I

- Popište základní části přepínače a jejich funkci.
- Popište vlastnosti a chování přepínače se sdílenou pamětí.
- Jaké jsou výhody a nevýhody křížového přepínače?
- Co je to blokování HOL a jak ho lze eliminovat?
- Popište princip a použití algoritmu PIM.
- Popište princip a použití algoritmu iSLIP.
- Porovnejte algoritmy přidělování lístků, PIM a iSLIP z hlediska efektivity, implementace a spravedlivosti.
- Uvažujte čtyřportový křížový přepínač se vstupy A až D a výstupy 1 až 4. Na vstupu přepínače je fronta paketů s těmito cílovými porty: A $\rightarrow$ (1,3,3), B $\rightarrow$ (2,3,4), C $\rightarrow$ (3,2,1), D $\rightarrow$ (4,3,1). Zapište, jak budou vypadat přenosy v časových slotech 1-6 při použití planování za použití algoritmu (i) přidělování lístků, (ii) PIM, (iii) iSLIP.

Otázky a úkoly k opakování II

- Popište princip a chování přepínací sítě Clos.
- Uvažujte přepínač o velikosti 128 vstupních portů. Kolik vnitřních propojení by vyžadoval při implementaci pomocí (i) křížového přepínače, (ii) neblokující sítě Clos, (iii) neblokující sítě Clos po přeskládání?
- Navrhněte architekturu 16-portového přepínače pomocí přepínací sítě Clos tak, aby bylo přepínání (i) neblokující, (ii) neblokující s přeskládání.
- Zakreslete schéma přepínací sítě Clos (2,3,4).
- Popište architekturu a vlastnosti přepínací sítě Beneš.
- Zakreslete nastavení sítě BN_3 pro následující konfiguraci vstupů a výstupů: $\{(0,2),(1,6),(2,5),(3,0),(4,7),(5,1),(6,4),(7,3)\}$
- Zakreslete nastavení sítě BN_4 pro následující konfiguraci vstupů a výstupů: $\{(0,0),(1,4),(2,8),(3,12),(4,1),(5,5),(6,9),(7,7),(8,2),(9,6),(10,10),(11,14),(12,3),(13,7),(14,11),(15,15)\}$

Použitá literatura

[1] D. Medhi and K. Ramasamy.
Network Routing. Algorithms, Protocols, and Architectures.
Elsevier, Inc., 2007.

[2] George Varghese.
Network Algorithmics.
Elsevier, Inc., 2005.

[3] Nick McKeown.
The iSLIP Scheduling Algorithm for Input-queued Switches.
IEEE/ACM Trans. Netw., 7(2):188–201, April 1999.

[4] Thomas E. Anderson, Susan S. Owicki, James B. Saxe, and Charles P. Thacker.
High speed switch scheduling for local area networks.
In *Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS V, pages 98–110, New York, NY, USA, 1992. ACM.

[5] William J. Dally.
Scalable Switching Fabrics for Internet Routers.
In *Standford University and Avici Systems*, 1999.

[6] Charles Clos.
A Study of Nonblocking Switching Networks.
Bell System Technical Journal, 32:406–424, 1952.

[7] Jiří Demel.
Grafy a jejich aplikace.
Academia, 2002.