

Přednáška #9: Permutační směrování v mřížkových a hyperkubických sítích

Předpoklady

- Permutace:
 - **úplné**: každý uzel vyšle a přijme **právě 1 paket**.
 - **neúplné**: každý uzel vyšle nebo přijme **nejvýše 1 paket**.
- Uvažujeme pouze **ortogonální** a **hyperkubické** sítě.
- Uvažujeme většinou **store-and-forward** (SF) sítě.
- Algoritmy pro efektivní permutační směrování ve **WH** sítích jsou zpravidla **komplikovanější**.
- V některých případech předpokládáme, že kromě standardních vstupních a výstupních front mají směrovače **pomocné** fronty pro β paketů. Pak definujeme, že permutace je **paměťově optimální**, jestliže $\beta = O(1)$.

Ortogonální síť

Lemma 1. Předpokládejme *plně duplexní všeportovou SF* mřížku $M(n)$ s $\beta = O(n)$ a *komunikační problém 1-mnoha* takový, že

1. každý uzel je *zdrojem jakéhokoli* počtu paketů,
2. každý uzel je *cílem nejvýše 1* paketu.

Pak řízení toku pomocí strategie *Farthest-First (FF)* lze tuto komunikaci provést v nejvýše $n - 1$ krocích.

Důkaz

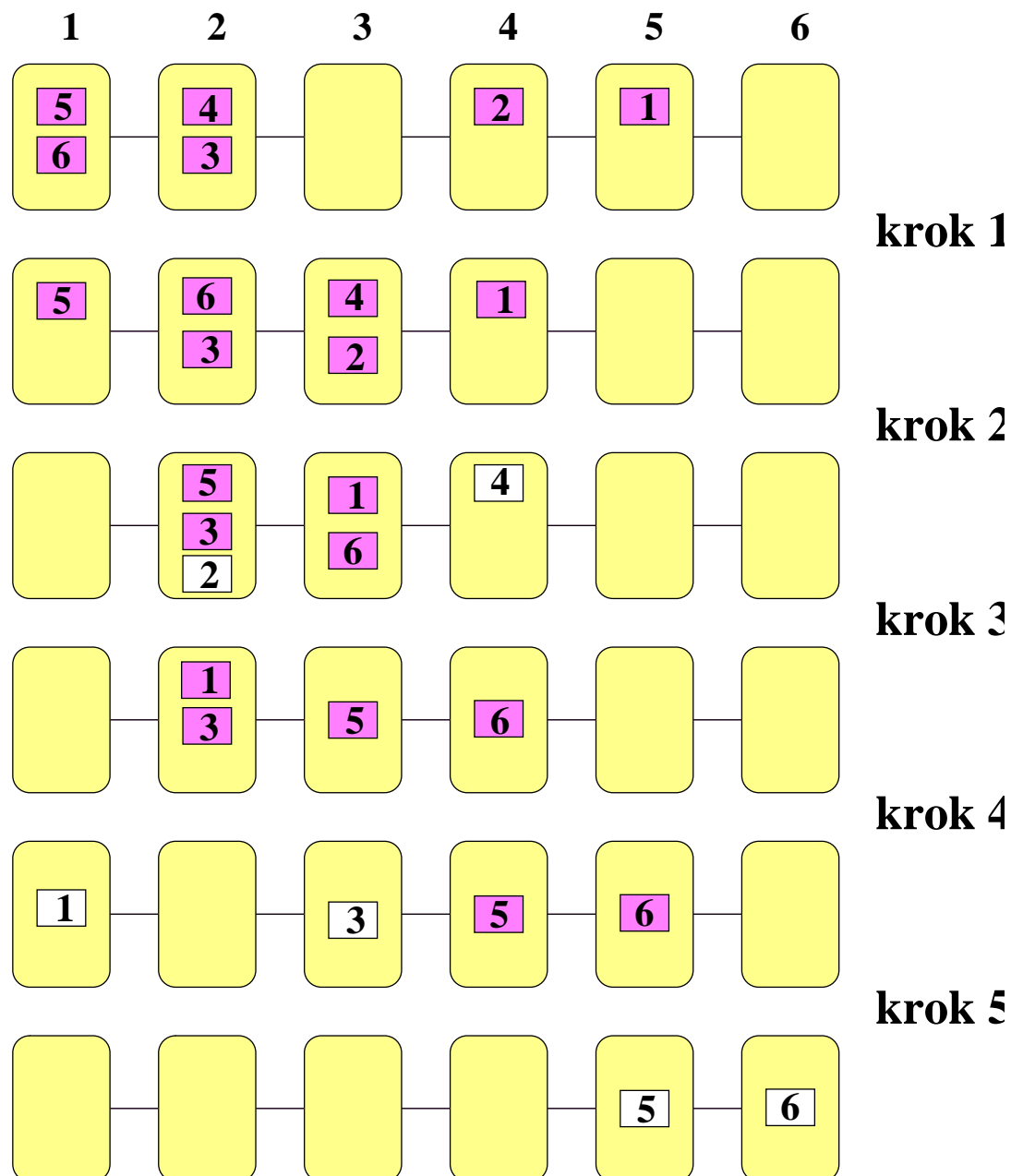
Indukcí podle *délky* cest, po kterých se pakety pohybují.

- Strategie FF: paket s nejdelší cestou před sebou má nejvyšší prioritu.
- Pro dané $1 \leq i \leq n - 1$, definujme $Z_i =$ podmřížka i uzlů nejvíce vpravo v $M(n)$.
 - *i-prioritní* pakety = *pakety směřující do* Z_i .
 - $k = \#$ *i-prioritních* paketů.
 - $p_j =$ zprava j -tý *i-prioritní* paket, $1 \leq j \leq k$.

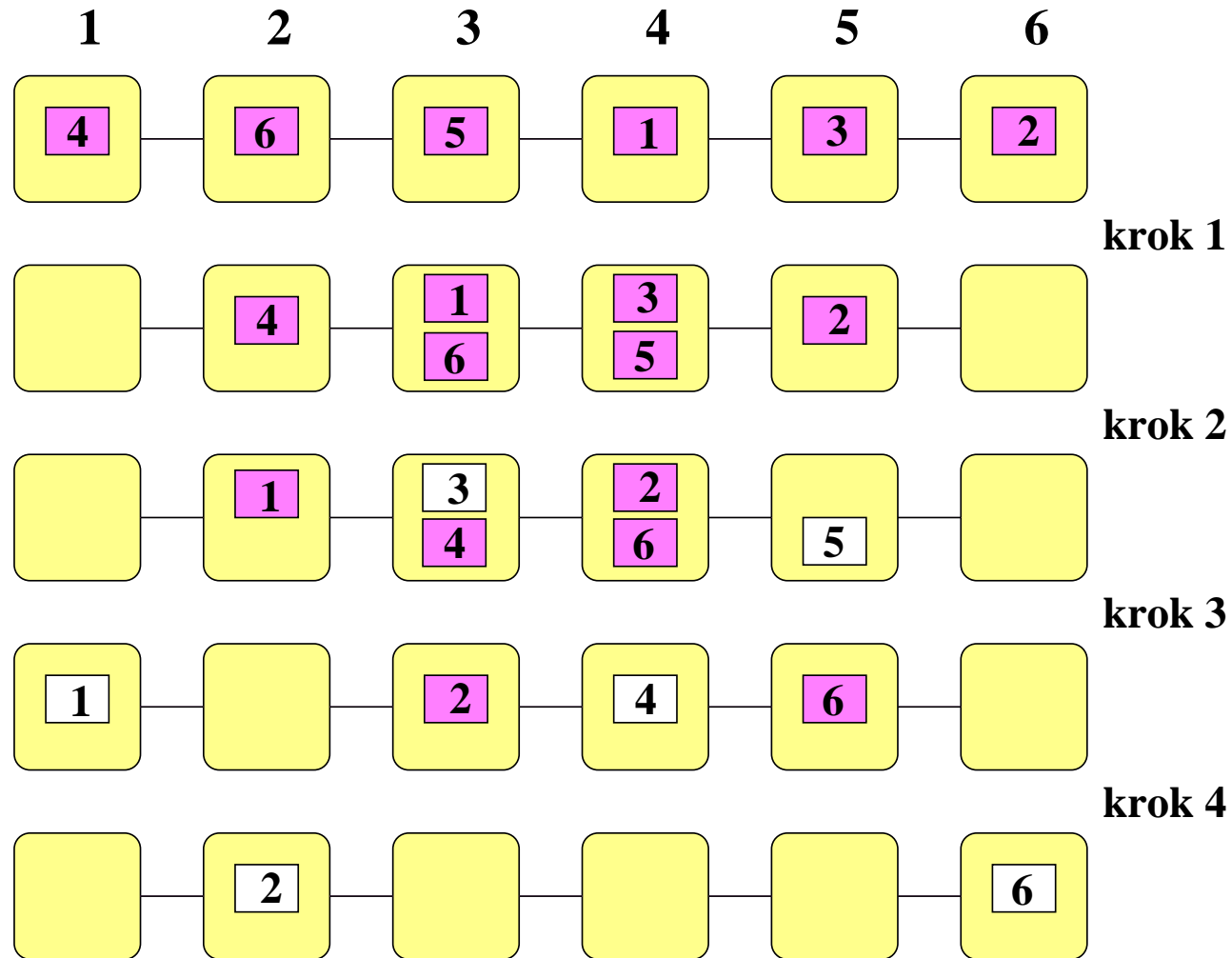
- Pakety pohybující se v opačném směru spolu nikdy nekolidují.
- i -prioritní pakety pokračují v pohybu bez přerušení, jakmile se jednou dají do pohybu.
- Konflikt více i -prioritních paketů v 1 uzlu je řešen pomocí FF.
- Strategie FF nikdy nezpozdí i -prioritní paket kvůli neprioritnímu paketu.
- Paket p_1 se dá do pohybu v kroku 1.
 - p_1 dosáhne bezkonfliktně Z_i v nejvýše $n - i$ krocích.
 - Po kroku 1 nemůže p_1 potkat, a tudíž zpozdit, jakýkoli jiný i -prioritní paket.
- p_1 může zpozdit v kroku 1 paket $p_2 \implies p_2$ se dá do pohybu nejpozději v kroku 2.
 - p_2 dosáhne bezkonfliktně Z_i v nejvýše $n - i + 1$ krocích.
 - Po kroku 2, p_2 nemůže zpozdit jakýkoli jiný i -prioritní paket.
- Indukce: poslední i -prioritní paket p_k se dá do pohybu nejpozději v kroku k .
- V nejhorším případě:
 - $k = i$,
 - p_i je v kroku $i - 1$ stále v nejlevějším uzlu,
 - k dosažení svého cíle, i -tého uzlu zprava, musí překonat $n - i$ hran

$\implies p_i$ potřebuje pro dosažení cíle nejvýše $(i - 1) + (n - i) = n - 1$ kroků.

■ Protože tento argument platí pro jakékoli i , $1 \leq i \leq n - 1$, lemma je dokázána.



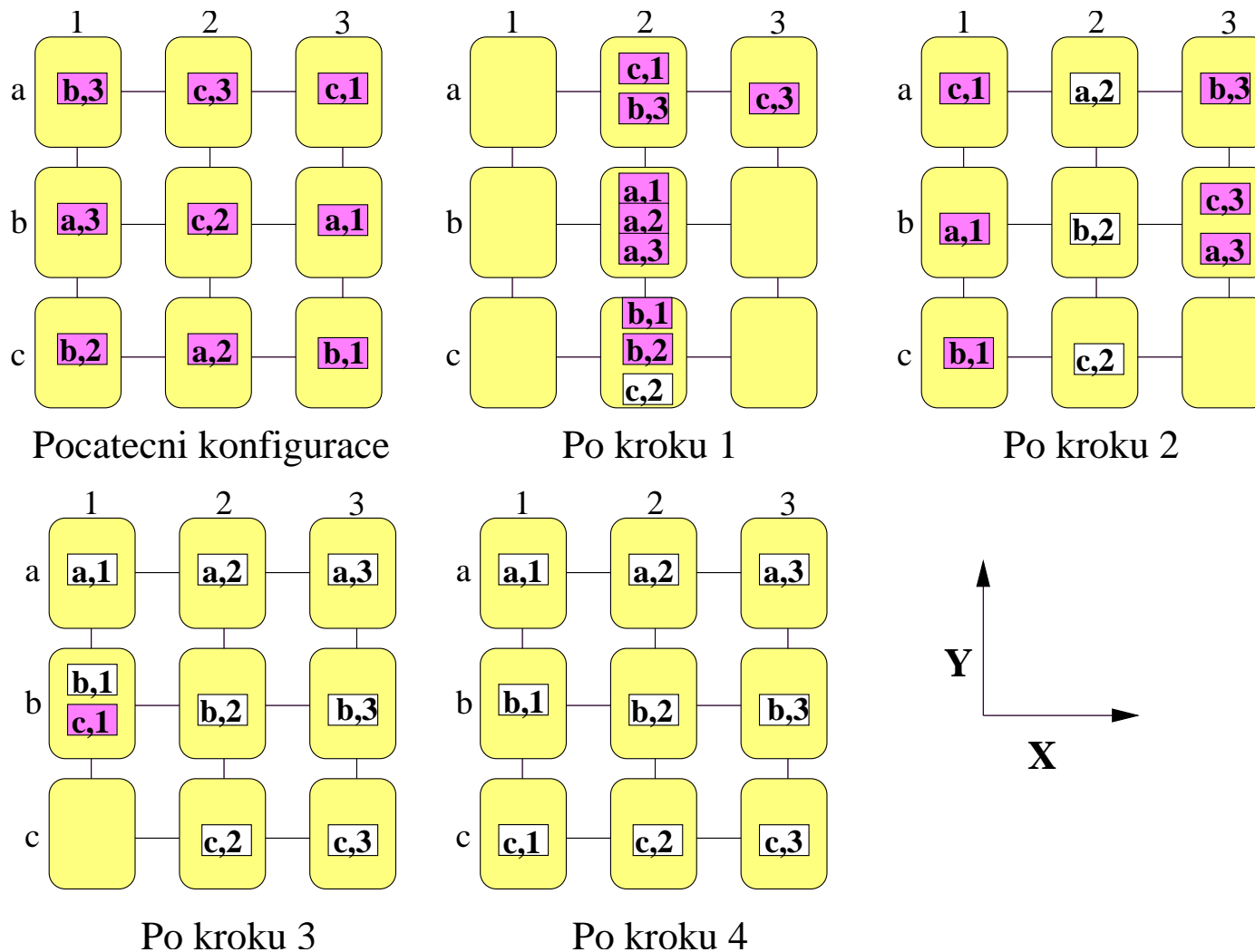
Všechny pakety se dají do pohybu v kroku 1 a jejich pohyb je bezkolizní.



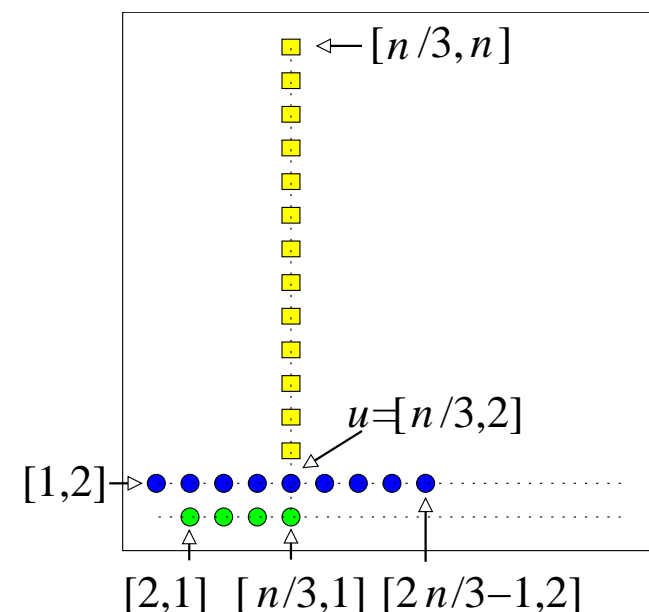
Poznámka: Téměř identické s EOTSort řazením na 1-D mřížce.

Věta 2. Uvažujme libovolnou permutaci π na všeportové plně-duplexní SF 2-D mřížce $M(n, n)$. Pak pro provedení permutace π pomocí XY směrování je potřeba nejvýše $2n - 2$ kroků, což je optimální, ale směrovače potřebují fronty na $\beta = \max(2, 2n/3 - 1)$ paketů.

Příklad úplné permutace na $M(3, 3)$



- **Důkaz časové složitosti:** Všechny pakety se pohybují ke svým cílovým sloupcům
 - \implies na horizontálních kanálech nejsou kolize
 - \implies každý paket dosáhne svůj cílový sloupec v **nejvýše $n - 1$ krocích**.
- **Strategie FF** ve sloupcích (Lemma 1)
 - \implies každý paket dosáhne svůj cílový **řádek** v **nejvýše $n - 1$ krocích**.
- **Důkaz** $\beta = \max(2, 2n/3 - 1)$. Triviálně $\beta \geq 2$. Uvažujme uzel $u = [n/3, 2]$:
 - max. # soupeřících paketů = # cílových uzlů = $n - 2$.
 - všech $n - 3$ paketů může do u dorazit v $n/3 - 1$ krocích
 - $\implies u$ musí dočasně uložit $1 + n - 3 - (n/3 - 1) = 2n/3 - 1$ paketů. ♣



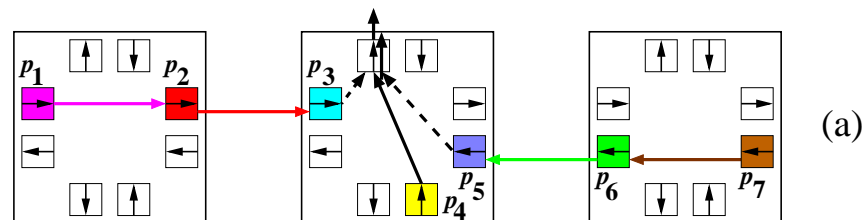
Vícerozměrné mřížky

Důsledek 3. *Jakákoli permutace π na všeportové plně-duplexní SF k -D mřížce $M(n, \dots, n)$ s dimenzně uspořádaným směřováním potřebuje nejvýše $k(n - 1)$ kroků a $\beta = \max(2k - 2, n - 2 - \frac{n-3}{2k-1})$.*

Existuje **časově i paměťově optimální** permutační směřování na 2-D mřížkách?

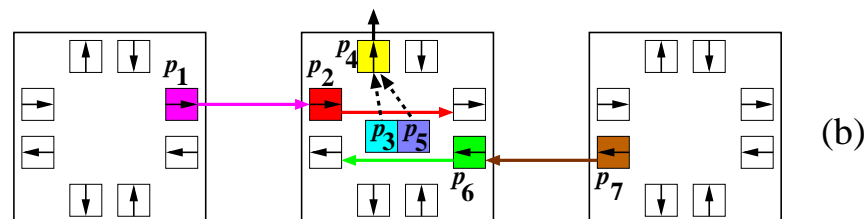
1. $\beta = 0$:

pakety p_1, p_2, p_6, p_7 se zpozdí



2. $\beta > 0$:

pakety p_1, p_2, p_6, p_7 se nezpozdí



Metody minimalizace paměťových požadavků pro permutační směřování

Tento problém minimalizace je značně obtížný. Obecně existují 3 obvyklé postupy:

- **randomizace**,
- permutační směřování založené na **seřazení** paketů,
- permutace s předvýpočtem: **off-line** permutační směřování s předpočítáním optimálních tras.

- Nejjednodušší postup pro zmenšení maximální velikosti fronty:

převedení permutace s nejhorším chováním na 2 náhodné permutace .

- Uvažujme permutaci π na p procesorech.

Algorithm VALIANTRANDOMIZEDPERM (**In:** permutation $\pi : \{1, \dots, p\} \rightarrow \{1, \dots, p\}$)

for all $i := 1, \dots, p$ **do_in_parallel**

{ **Fáze 1:** Vygeneruj náhodný mezilehlý uzel $g(i)$.

Fáze 2: Pošli paket z i do $g(i)$ použitím základního minimálního směrování.

Fáze 3: Pošli paket z $g(i)$ do $\pi(i)$ použitím základního minimálního směrování. }

- 2 vysílače si vybírají své cíle **nezávisle** \implies 1 uzel může být cílem více než 1 paketu.

Věta 4. *Jakoukoli permutaci v $M(n, n)$ lze pomocí randomizace realizovat v $2n + o(n)$ krocích s $\beta = O(\log n)$ frontami s pravděpodobností aspoň $1 - O(1/n^2)$.*

Náznak důkazu:

Algorithm MESHRANDPERM (**In:** permutation $\pi : \{1, \dots, n^2\} \rightarrow \{1, \dots, n^2\}$)

{ **Fáze 0:** Rozděl každý sloupec do $\log n$ **segmentů** velikosti $n / \log n$.

Fáze 1: Pošli každý paket *náhodnému* cíli uvnitř jeho segmentu.

Fáze 2: Pošli každý paket uvnitř jeho současného řádku do jeho *správného* sloupce.

Fáze 3: Pošli každý paket uvnitř jeho *správného* sloupce do jeho *správného* řádku. }

Při kolizi na kanálech použijí směrovače opět **strategii FF**.

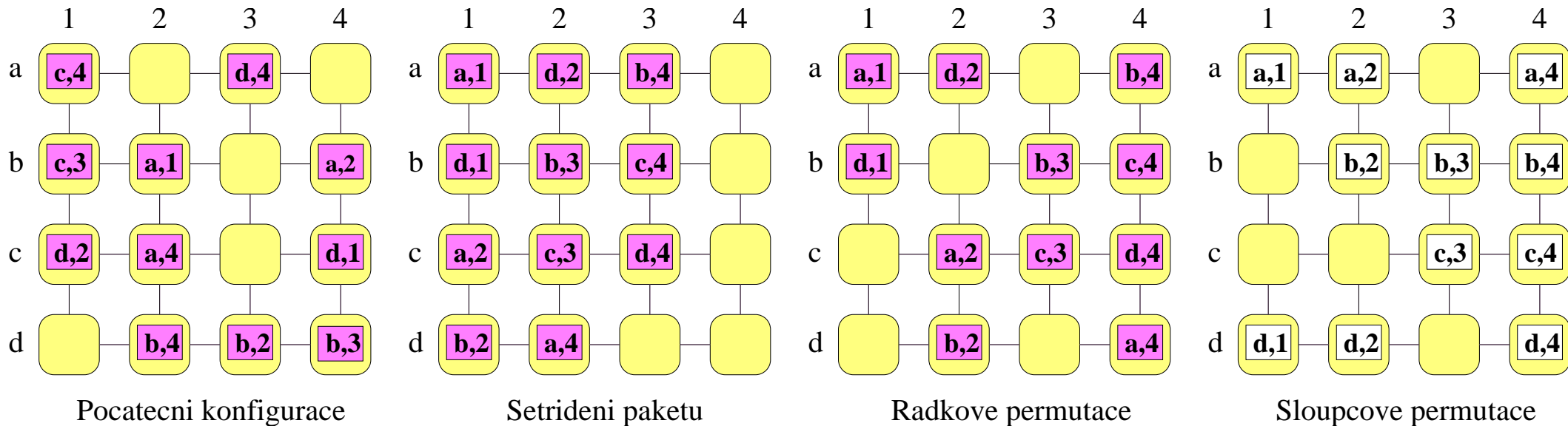


Věta 5. *Permutační směrování na všeportové SF $M(n, n)$ s plně-duplexními kanály může být provedeno*

v $O(T_{\text{sort}}(M(n, n)) + 3n)$ krocích s $\beta = 0$,

kde $T_{\text{sort}}(M(n, n))$ je paralelní čas hadovitého seřazení n^2 čísel na $M(n, n)$.

Z přednášky #8 (Paralelní třídící algoritmy) víme, že $T_{\text{sort}}(M(n, n)) = 3n + o(n)$.



Algorithm SORTINGBASEDMESHPERM (**In:** permutation $\pi : \{1, \dots, n^2\} \rightarrow \{1, \dots, n^2\}$)

Fáze 1: Seřad' pakety *hadovitě do sloupců* podle adresy jejich *cílového sloupce* .

(* $T_{\text{sort}}(M(n, n))$ kroků *)

Fáze 2: Otoč každý 2. sloupec.

(* $n - 1$ kroků *)

(* Nyní máme pakety seřazeny lexikograficky do sloupců. *)

(* Počet paketů s *týmž cílovým sloupcem* je $\leq n$. *)

(* Tudíž, v každém řádku, \exists *nejvýše 1* paket určený pro daný sloupec. *)

Fáze 3: Paralelně ve \forall řádkách, zpermutuj pakety do *správných sloupců*. (* $n - 1$ kroků *)

(* Nyní v každém sloupci, \exists *nejvýš 1* paket určený pro 1 daný řádek. *)

Fáze 4: Paralelně ve \forall sloupcích, zpermutuj pakety do *správných řádků*. (* $n - 1$ kroků *)

Z popisu algoritmu plyne, že v žádné fázi nepotřebujeme žádné pomocné fronty.



- Směrování založené na seřazení = **on-line** řešení: permutace **není známá** dopředu a každý uzel **zná pouze** svůj paket.
- V řadě aplikací je tatáž předem známá permutace **prováděna opakovaně**. Typický příklad:

simulace 1 sítě nebo paralelního počítače na jiné síti nebo jiném počítači.

⇒ Pak si lze předpočítat **off-line** časově anebo paměťově optimální směrování.

Věta 6. *Existuje off-line algoritmus, který pro libovolnou permutaci na $M(n, n)$ předpočítá směrování o $(3n - 3)$ krocích a s $\beta = 0$.*

Důkaz. Algoritmus má také 4 fáze (poslední 2 jsou stejné jako předtím).

Algorithm OFFLINEMESHPERM (**In:** permutation $\pi : \{1, \dots, n^2\} \rightarrow \{1, \dots, n^2\}$)

Fáze 1: Ve \forall sloupcích, *předpočítej* permutace takové, aby v každém řádku $M(n, n)$
 \exists nejvýše 1 paket určený pro daný sloupec.

Fáze 2: Realizuj tyto permutace ve všech sloupcích paralelně. (* $n - 1$ kroků *)

Fáze 3: Paralelně ve \forall řádkách, zpermutuj pakety do *správných sloupců*. (* $n - 1$ kroků *)

Fáze 4: Paralelně ve \forall sloupcích, zpermutuj pakety do *správných řádků*. (* $n - 1$ kroků *)

- Pro danou permutaci π na $M(n, n)$, zkonstruuj **směrovací bipartitní graf** G_π s $V(G_\pi) = S \cup D$, kde
 - $S = \{s_1, \dots, s_n\} =$ **zdrojové sloupce** $M(n, n)$,
 - $D = \{d_1, \dots, d_n\} =$ **cílové sloupce** $M(n, n)$,
 - $\langle s_{j_1}, d_{j_2} \rangle \in E(G_\pi) \iff$ sloupec j_1 obsahuje **paket** s cílovým sloupcem j_2 .
- Použij Hallovu větu pro konstrukci n -**barvení** v G_π .

Hallova věta o párování

Věta 7. *Libovolný k -regulární bipartitní graf může být rozdělen do k dokonalých párování čili má k -barvení hran.*



for $i = 1, \dots, k - 1$ **do_sequentially**

begin

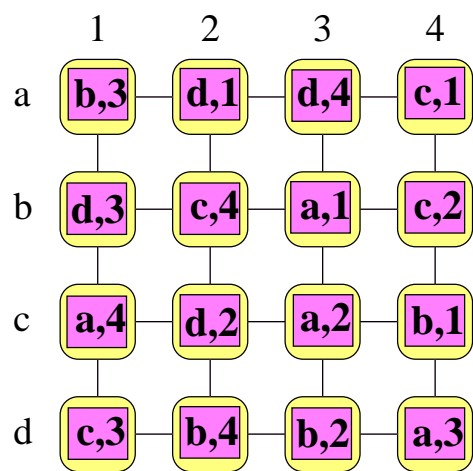
zkonstruuj 1 dokonalé párování a obarvi jeho hrany barvou i ;

odstraň hrany tohoto párování i z G ;

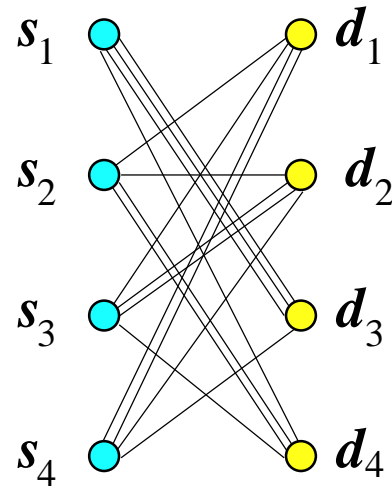
end

Zbývajícimu dokonalému párování v G přiřaď barvu k

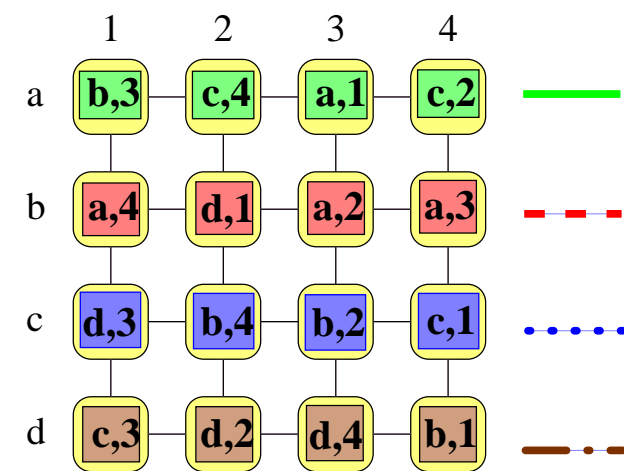
1 dokonalé párování $\approx n$ hran s barvou $i \approx n$ paketů přesunutých do řádku i



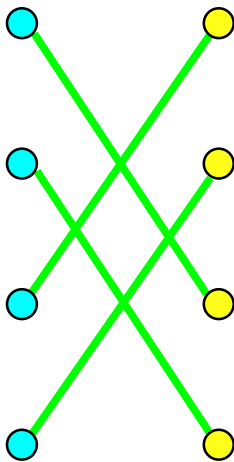
Pocatecni konfigurace



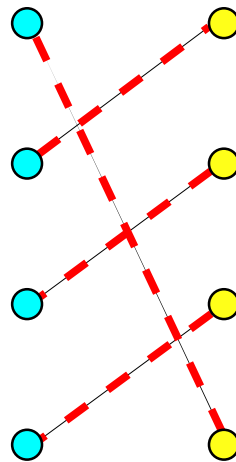
Smerovaci graf



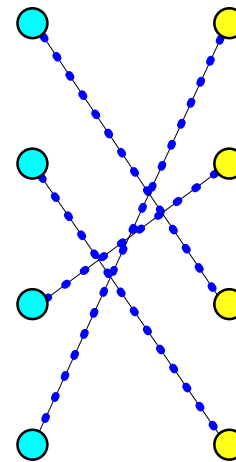
Permutace sloupce



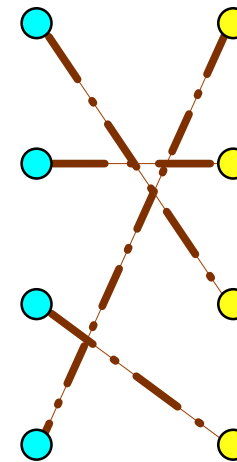
pakety pro radek a



pakety pro radek b



pakety pro radek c

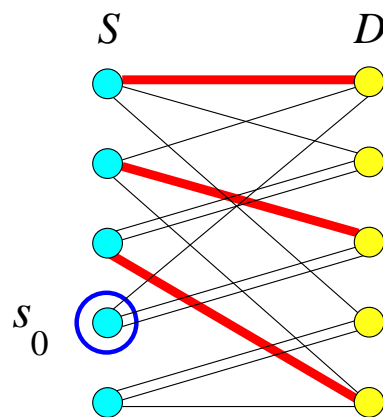


pakety pro radek d

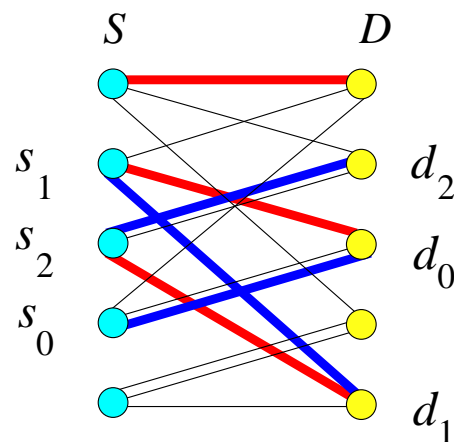
Obarveni smerovaciho grafu

Indukcí přes velikost párování.

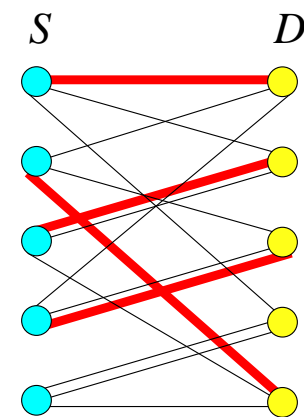
- Máme-li **částečné párování** M v G_π s m hranami, $1 \leq m < n$, pak zkonstruujeme párování M' s $m + 1$ hranami.
- Označme hrany M jako **červené** a všechny ostatní hrany G_π jako **modré**.
- Nechť $s_0 \in S$ je vrchol, který není incidentní s žádnou červenou hranou. Pak
 - \exists jednoduchá cesta P liché délky, **procházka**, začínající v s_0 modrou hranou, používající střídavě červené a modré hrany, a končící modrou hranou ve vrcholu v D , který nemá červenou hranu.
- **Prohod'** barvy hran v P
 - \implies $\#$ červených hran v dosavadním párování se zvýší o 1
 - \implies obdržíme párování s $m + 1$ hranami.



Parovani s 3 hranami



Prochazka



Parovani s 4 hranami

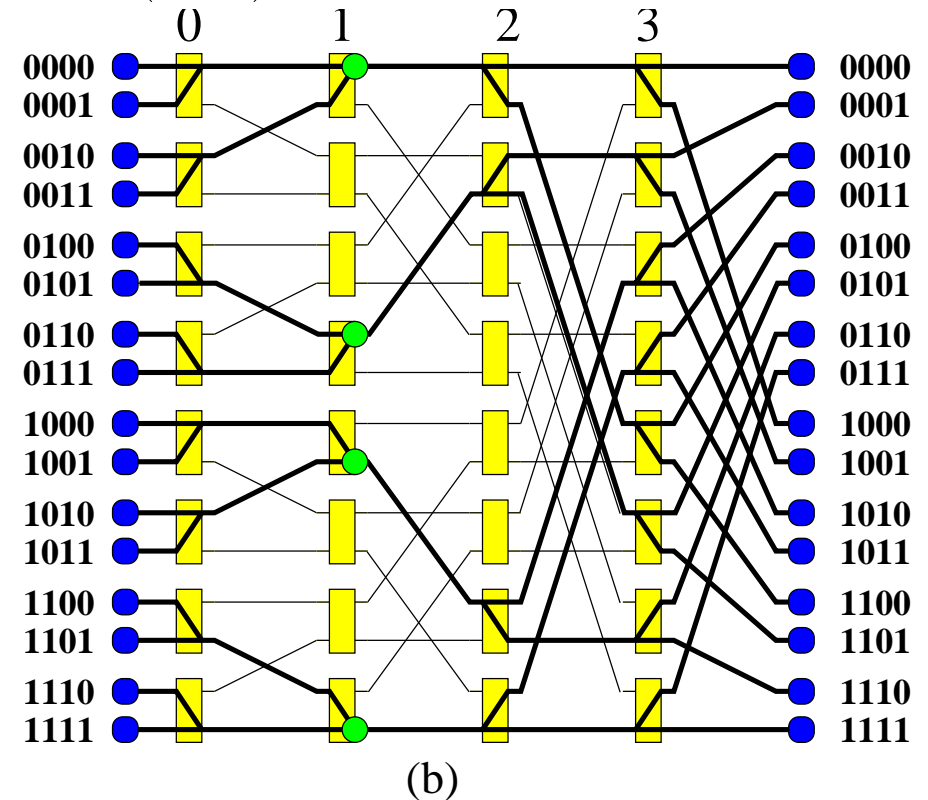
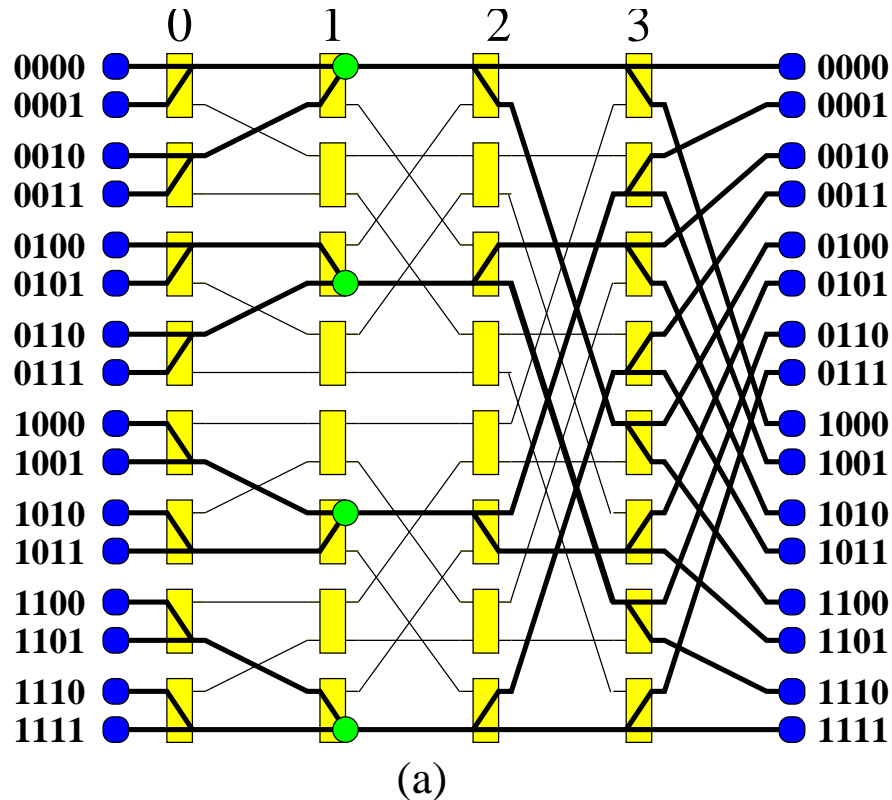
On-line permutační směrování na síti motýlek

Nechť $N = 2^n$. **Permutace** = bitově definované operace na n -bitových řetězcích.

- **Otočení** $\pi_r : u_{n-1} \dots u_0 \mapsto u_0 \dots u_{n-1}$.
- **Prohození** $\pi_t : u_{n-1} \dots u_k u_{k-1} \dots u_0 \mapsto u_{k-1} \dots u_0 u_{n-1} \dots u_k$, je-li $n = 2k$
 $\pi_t : u_{n-1} \dots u_{k+1} u_k u_{k-1} \dots u_0 \mapsto u_{k-1} \dots u_0 u_k u_{n-1} \dots u_{k+1}$, je-li $n = 2k + 1$
- **Přeložení (posun) do** w , $\pi_p^w : i \rightarrow i \text{ XOR } w$ pro daný n -bitový řetězec w .
- **Doplňěk** $\pi_d : u_{n-1} \dots u_0 \mapsto \overline{u_{n-1}} \dots \overline{u_0}$ (= speciální případ přeložení).
- **Cyklický posun** $\circ \delta$, $\pi_s^\delta : i \mapsto i \oplus_p \delta$, $i \in \{1, \dots, p\}$.

Otočení a prohození

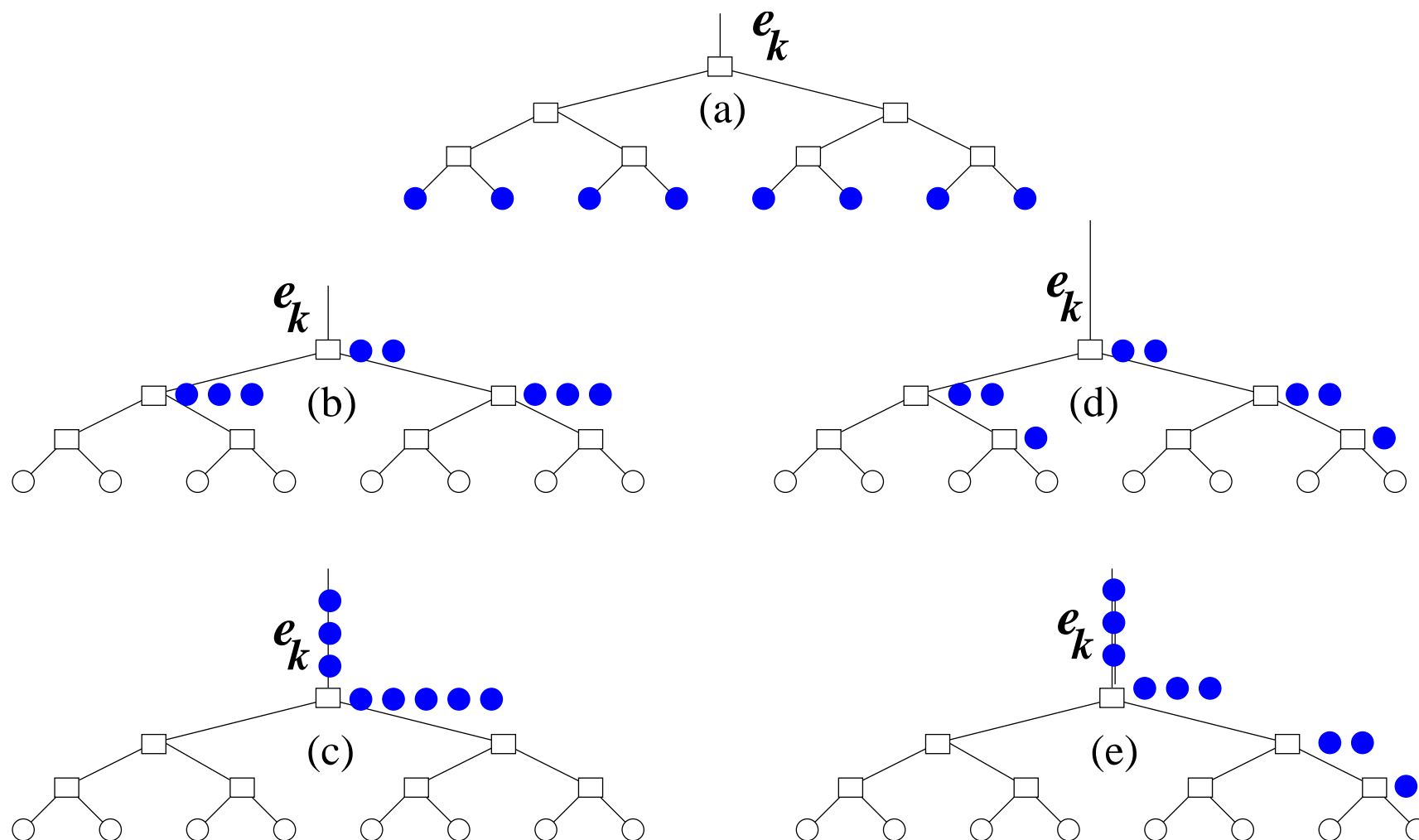
Věta 8. Permutace *otočení* π_r a *prohození* π_t ve všeportovém SF nepřímém motýlku $indBF_n$ potřebuje při minimálním on-line směřování $\Theta(\sqrt{N})$ kroků, kde $N = 2^n$.



Permutace otočení (a) a prohození (b) při minimálním směřování na $indBF_4$.

- Nechť $n = 2k$. (Pro liché n podobné.)
- Nechť U_{k-1} je výstupní port $(k-1, u_{n-1} \dots u_{k+1} u_k u_k u_{k+1} \dots u_{n-1})$, U_k je vstupní port $(k, u_{n-1} \dots u_{k+1} u_k u_k u_{k+1} \dots u_{n-1})$ a $e_k = \langle U_{k-1}, U_k \rangle$ je **střední kanál**.
- 2^k uzlů $(0, u_{n-1} \dots u_k *^k) =$ listy CBT_k s kořenem U_{k-1} .
- Všechny pakety z CBT_k směřují k střednímu kanálu e_k :
 $(0, u) \rightarrow (1, u_{n-1} \dots u_1 u_{n-1}) \rightarrow (2, u_{n-1} \dots u_2 u_{n-2} u_{n-1}) \rightarrow \dots \rightarrow$
 $(k-1, u_{n-1} \dots u_k u_k \dots u_{n-1}) \rightarrow (k, u_{n-1} \dots u_k u_k \dots u_{n-1}) \rightarrow$
 $(k+1, u_{n-1} \dots u_{k-1} u_k u_k u_{k+1} \dots u_{n-1}) \rightarrow \dots \rightarrow (n-1, u_0 \dots u_{n-1})$.
- První paket překročí hranu e_k v kroku $k+1$ a poslední paket v kroku $k+2^k$.
- Tento poslední paket pak potřebuje k kroků, aby se dostal do sloupce nejvíce vpravo
 \implies spodní mez na počet kroků je $2k + 2^k = \log N + \sqrt{N}$.
- $\exists \sqrt{N}$ CBTs $\implies \exists \sqrt{N}$ středních kanálů pro přenos N paketů zleva doprava.
- Paměťové požadavky:
 1. $\beta = O(2^{k-1}) = \sqrt{N/8}$ (záplavový algoritmus),
 2. $\beta = 0$ (priorita zleva).





(b)+(c) 3. a 6. krok v CBT_n s kořenem U_{k-1} při $\beta = \Theta(\sqrt{N})$.

(d)+(e) 3. a 6. krok v CBT_n s kořenem U_{k-1} při $\beta = 0$.

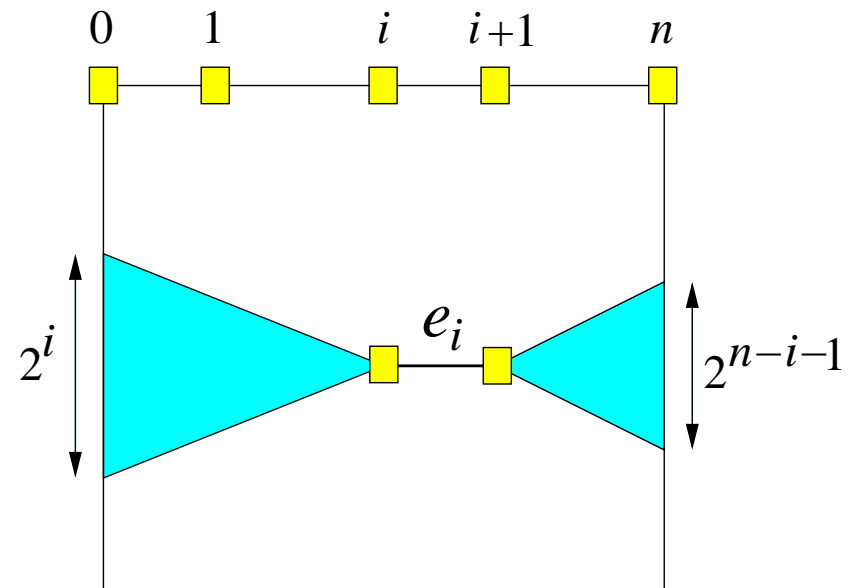
Lemma 9. Věta 8 platí pro každou permutaci

$$\pi(u_{n-1} \dots u_{\frac{n}{2}} u_{\frac{n}{2}-1} \dots u_0) = \pi_1(u_{\frac{n}{2}-1} \dots u_0) \pi_2(u_{n-1} \dots u_{\frac{n}{2}}).$$

Permutace otočení a prohození = **nejhorší případy permutací** pro minimální směrování na hyperkubických sítích.

Věta 10. *Libovolné permutační směrování v oBF_n může být provedeno v $O(\sqrt{N})$ krocích použitím **minimálníhoho** algoritmu, jestliže $\beta = \Theta(\sqrt{N})$.*

- e_i = jakákoli hrana na úrovni i
- g_i = počet nejkratších cest jdoucích skrz e_i
- $g_i \leq \min(2^i, 2^{n-i-1})$
- paket může být zpožděn ve sloupci i nejvýše $g_i - 1$ ostatními pakety
- zpoždění v nejhorším případě je $S = \sum_{i=1}^n (g_i - 1)$
- $S \leq \begin{cases} 3\sqrt{N/2} - n - 2 & \text{je-li } n \text{ liché,} \\ 2\sqrt{N} - n - 2 & \text{je-li } n \text{ sudé.} \end{cases}$



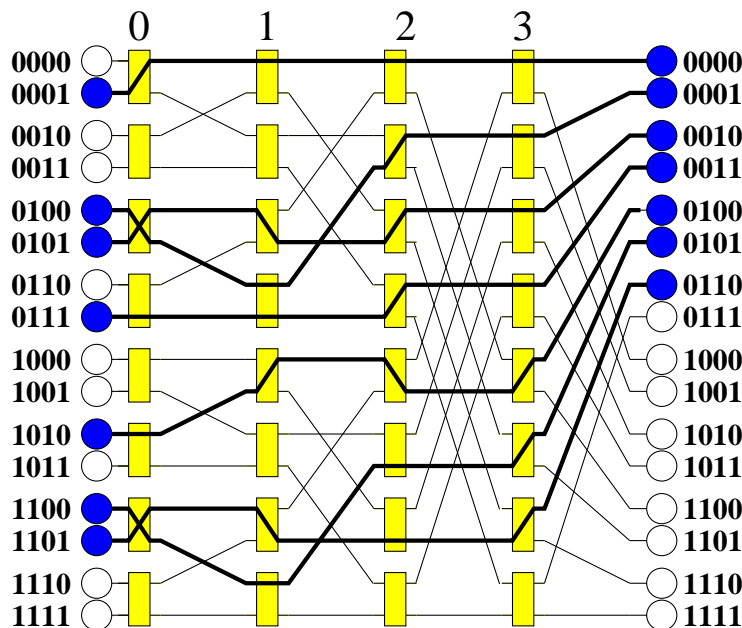
Zhušťovací (*packing*) neúplná permutace

- Směrování paketů z **libovolného podmnožiny vstupů** \mathcal{M} do **prvních** $m = |\mathcal{M}|$ výstupních uzlů tak, že

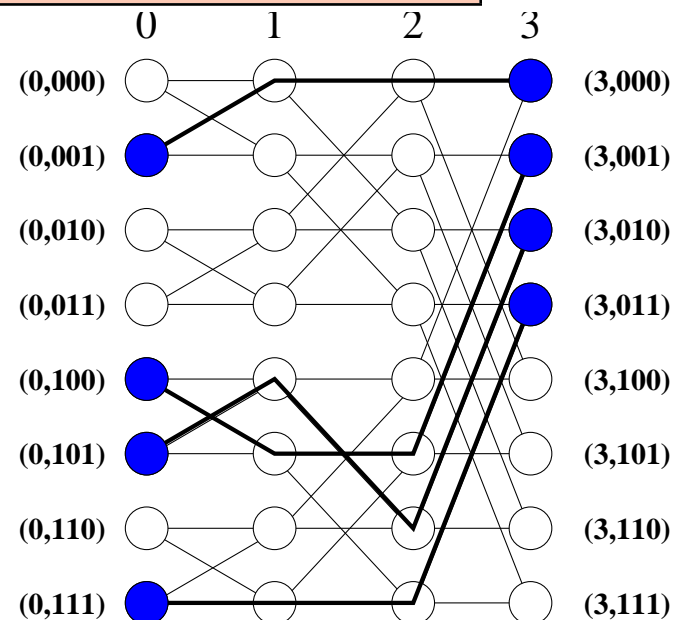
relativní pořadí paketů se nezmění.

- Výpočet **relativního pořadí** paketů v \mathcal{M} : **paralelní prefixový součet**.
- Aplikace: VLSI návrh, kolektivní komunikační problémy 1-mnoha, **cyklický k -posun**.

Příklad zhušťovací permutace na motýlcích



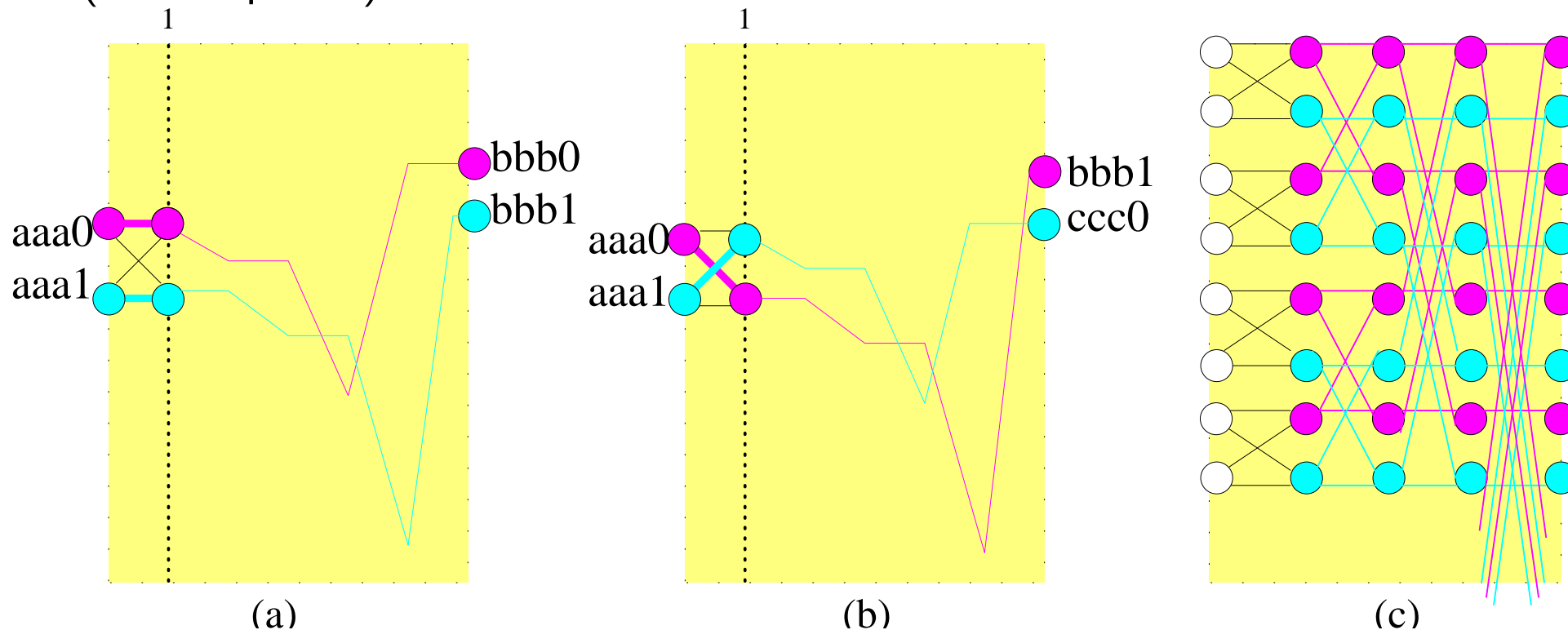
(a) $indBF_4$, $m = 7$.



(b) oBF_3 , $m = 4$.

Věta 11. Minimální směřování na oBF_n poskytuje *uzlově disjunktní* cesty pro *libovolnou* zhušťovací permutaci.

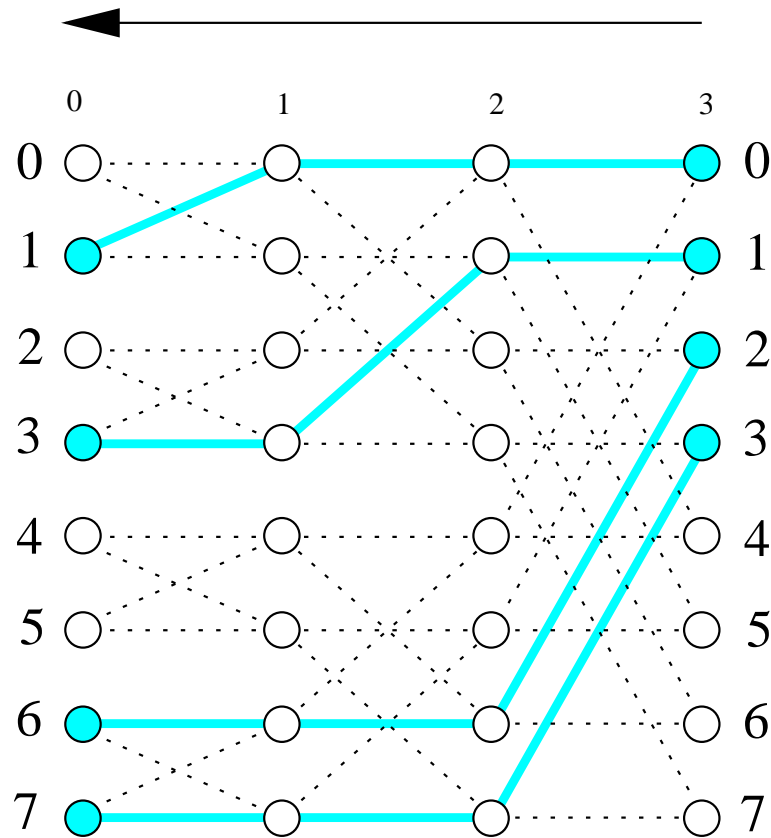
Důkaz. (indukcí přes n)



- Žádné 2 pakety se nemohou srazit na uzlu v **sloupci 1**, viz jediné dva možné případy na obr. (a) a (b). (dva za sebou následující pakety zůstanou těsně za sebou i na výstupech, pouze se můžou posunout a tudíž změnit paritu (hodnotu posledního bitu)).
- Indukce: na žádném dalším stupni nemůže dojít k soupeření o linku, protože pakety se rozdělí na ty se **sudým číslem** (červené) a s **lichým číslem** (zelené).



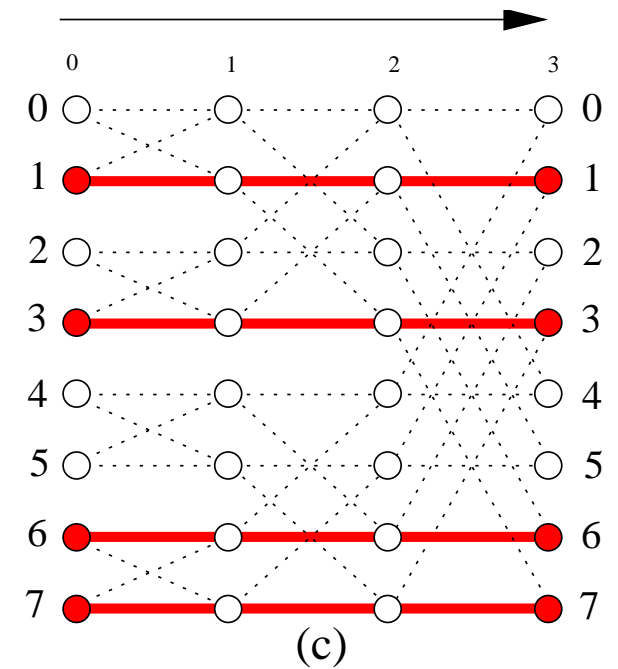
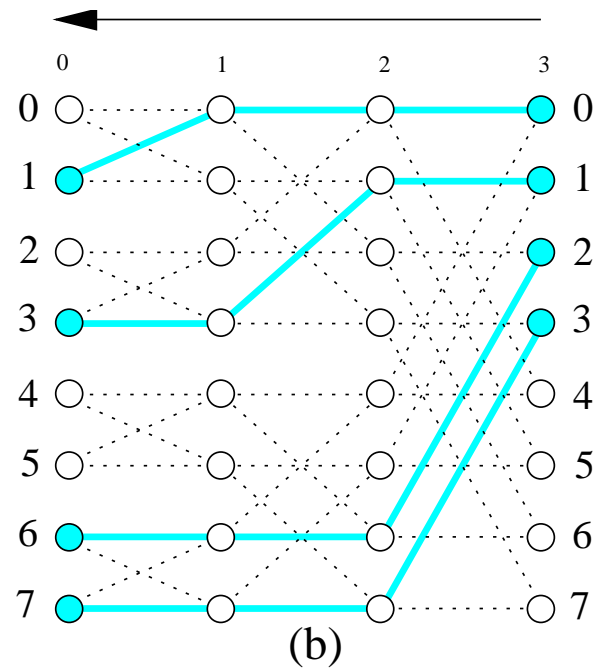
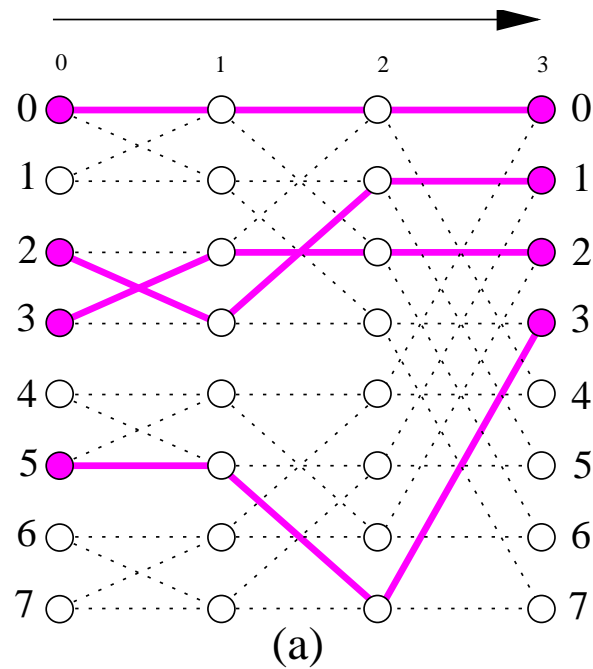
- Zředovací permutace = otočení zhušťovací permutace.



Každá permutace na oBF_n = seřazení paketů podle cílových adres + zředění
v čase $T_{\text{sort}}(N) + O(n)$, kde $N = 2^n$ a $T_{\text{sort}}(N)$ je čas seřazení N položek na oBF_n .

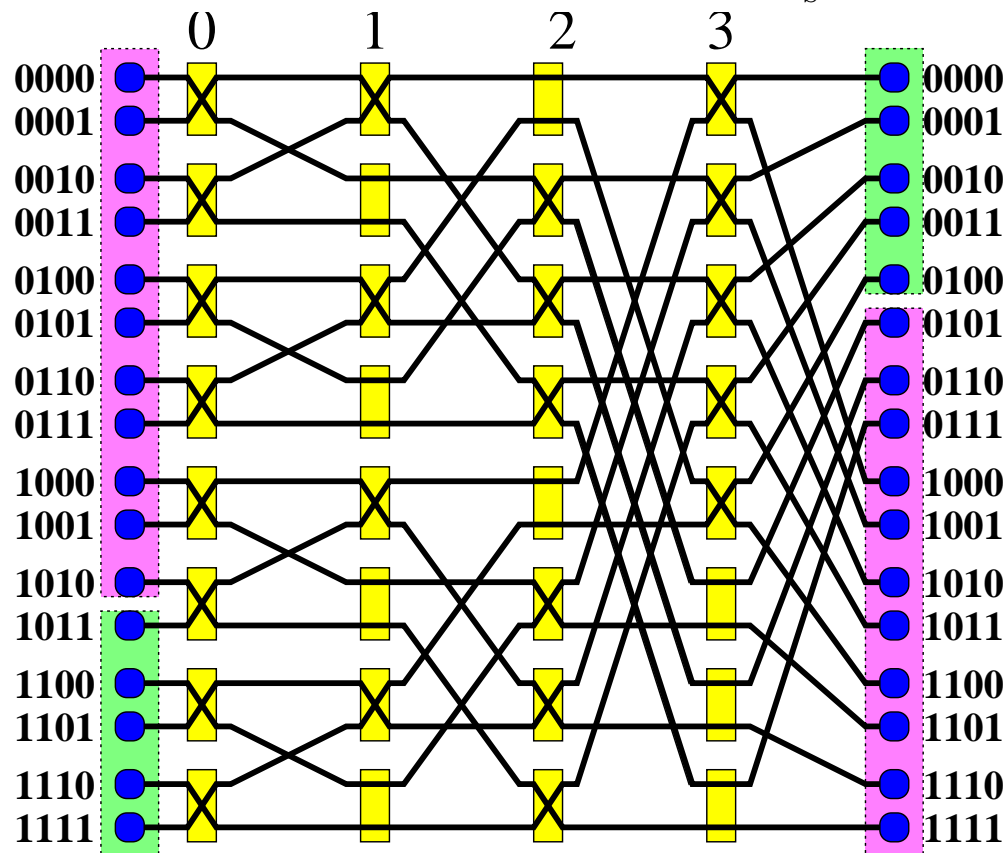
V přednášce #8 jsme viděli, že $T_{\text{sort}}(N) = O(n^2)$.

Libovolná monotónní permutace = složení zhušťovací permutace a zředovací permutace

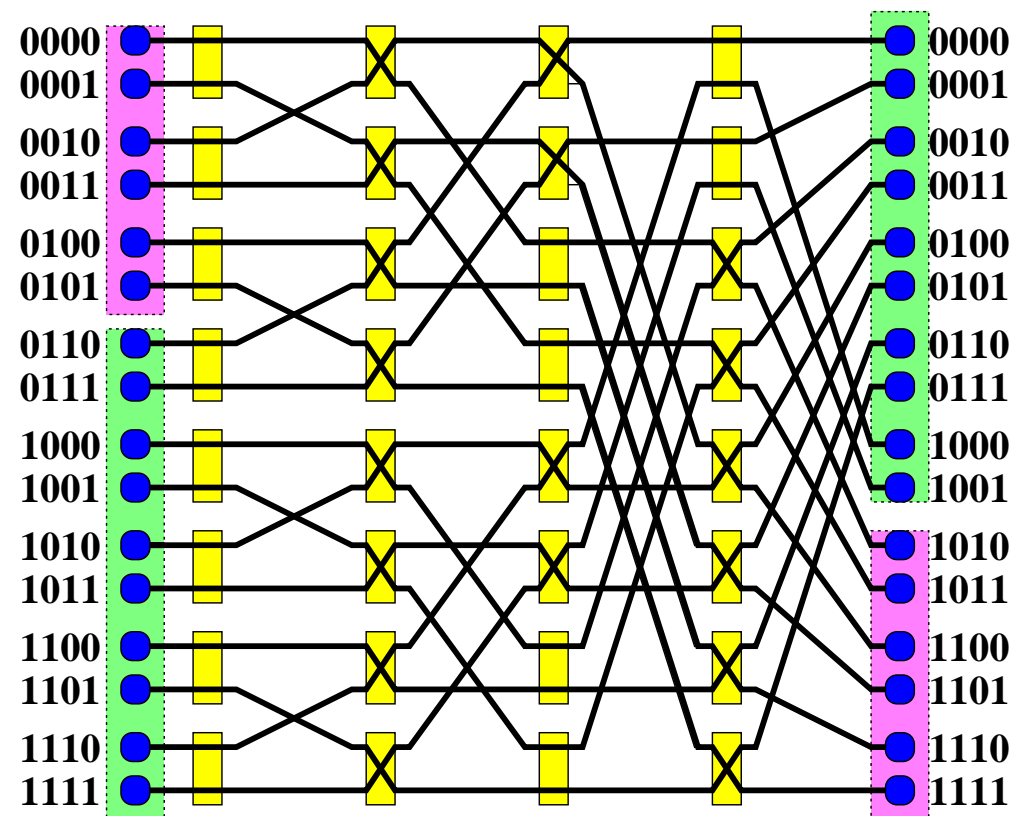


Cyklický posun (CP)

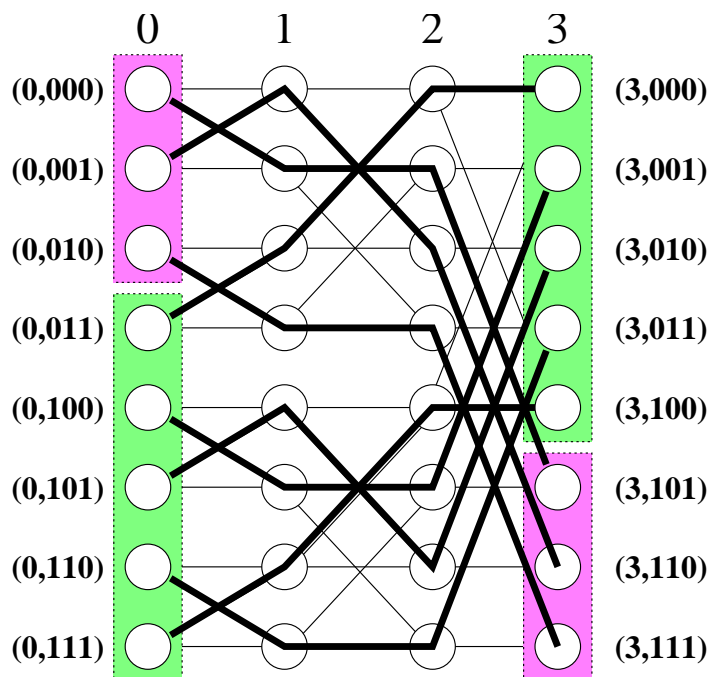
Cyklický posun o δ je permutace $\pi_s^\delta : i \mapsto i \oplus_p \delta$.



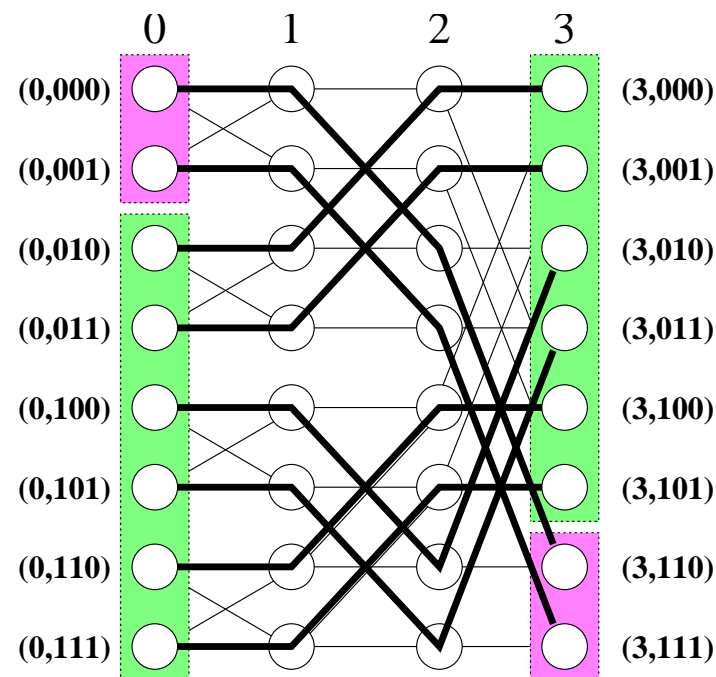
(a) CP o $\delta = 5$ ($= 0101_2$) na $indBF_4$



(b) CP o $\delta = 10$ ($= 1010_2$) na $indBF_4$.



(a) CP o $\delta = 5$ ($= 101_2$) na oBF_3



(b) CP o $\delta = 6$ ($= 110_2$) na oBF_3 .

Věta 12. Minimální směřování na oBF_n poskytuje *uzlově disjunktní* cesty pro permutaci cyklický posun π_s^δ .

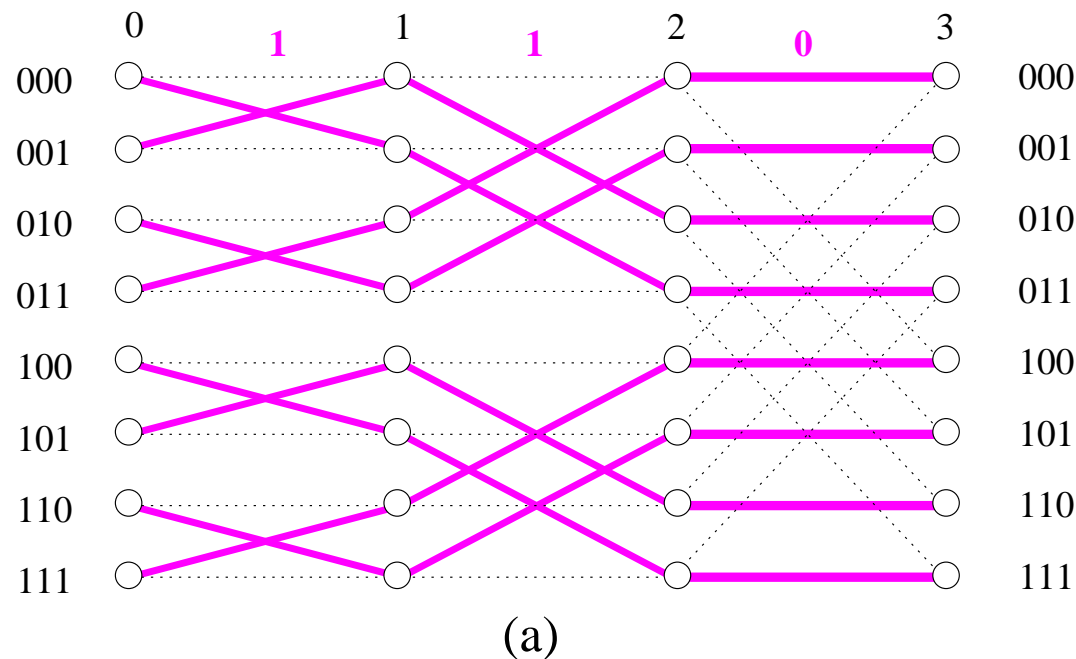
Idea důkazu (indukcí přes n , podobný jako u zhuštění)

- Cesty jsou disjunktní v 1. sloupci, cyklický posun je složení 2 částečných permutací zahuštění, v 1. stupni motýlka se použijí buď pouze přímé nebo pouze křížové linky (stejný argument jako u permutace zhuštění).
- Od 2. stupně se permutace rozpadá na 2 samostatné permutace cyklický posun na disjunktních podmotýlcích oBF_{n-1} .



Věta 13. *Minimální směřování* na oBF_n poskytuje pro permutaci *přeložení* π_p^w , $0 \leq w \leq N - 1$, *uzlově disjunkttní cesty*.

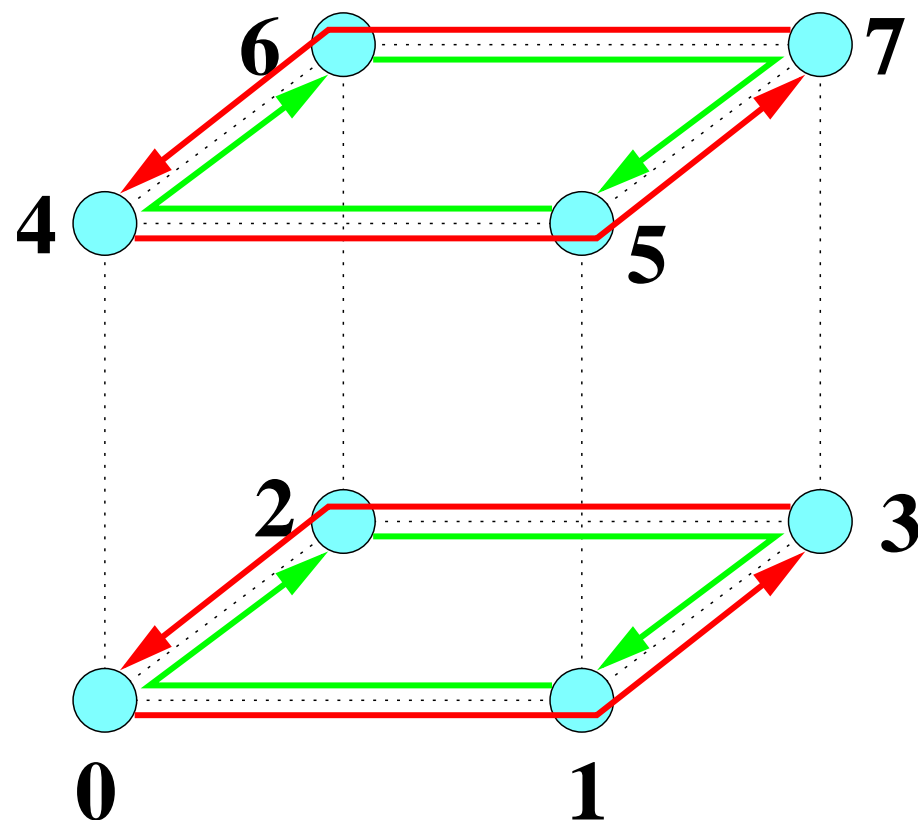
Důkaz. Všechny cesty začínají v různých vstupních uzlech. Na každém stupni oBF_n , všechny cesty používají buď hrany přímé nebo křížové. Proto tyto cesty nikdy nesoupeří o žádnou hranu.



Bezkolizní směřování pro permutaci přeložení π_p^w pro $w = 011$ na oBF_3 .

Permutace *doplňk* $\pi_c \implies$ všechny pakety používají ve všech stupních pouze *křížové hrany*.

- Všechny algoritmy pro minimální permutační směrování na SF všeportové nepřímé síti oBF_n jsou **normální hyperkubické**
 \implies mohou být prováděny na **kterékoli** hyperkubické síti s $O(1)$ zpomalením.
- Libovolné **hranově disjunktí** permutační směrování na **SF** oBF_n
 \iff permutační směrování **s nulovým zahlcením** na **WH plně-duplexní všeportové** Q_n .

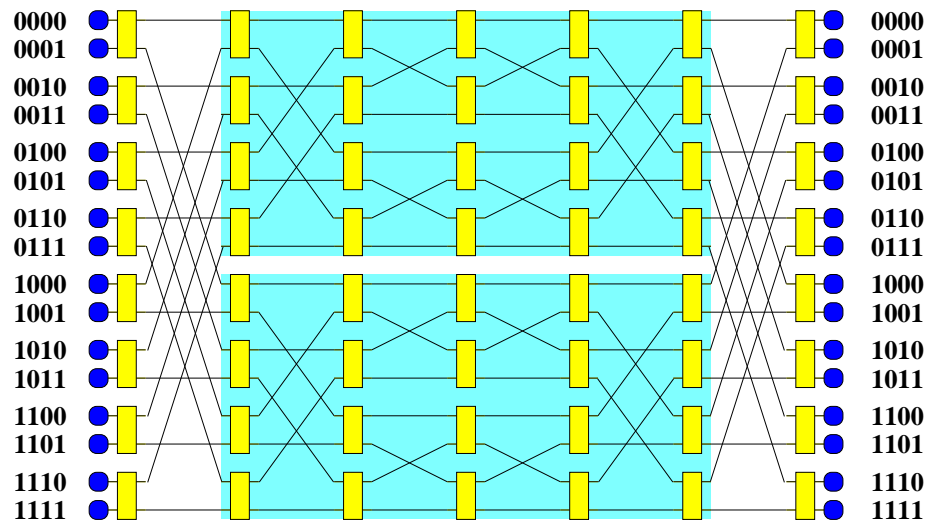


Přeložení π_p^w pro $w = 011$ na WH plně-duplexní všeportové Q_3 používající e -cube směrování.

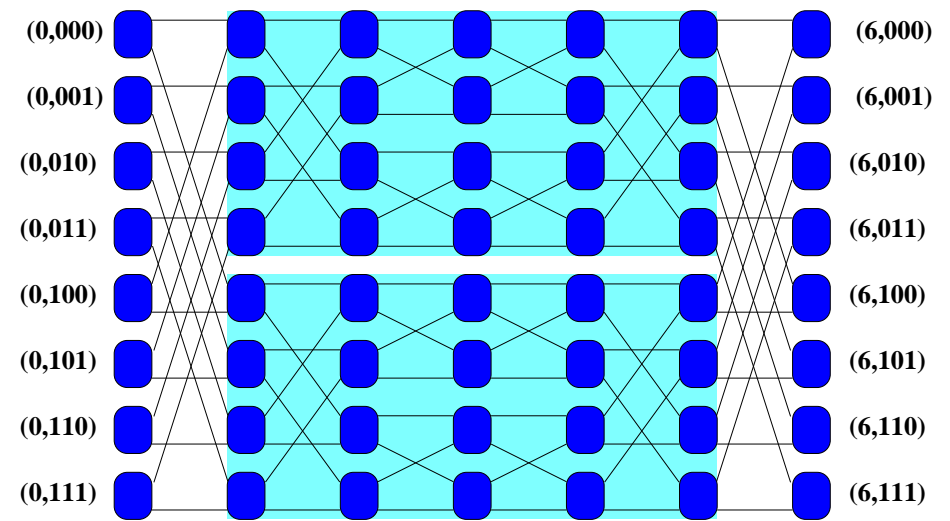
- Benešova síť $BN_n = 2$ motýlkové zády k sobě.
- BN_n má $2n + 1$ sloupců uzlů a $2n$ úrovní hran.
- Hierarchicky rekurzivní: BN_n obsahuje horní BN_{n-1}^0 a spodní BN_{n-1}^1 .
- Přestavitelná úplná permutační síť:

pro libovolnou permutaci vstupů na výstupy, \exists bezkolizní permutační směrování.

- \exists 2 varianty: přímá oBN a nepřímá $indBN$

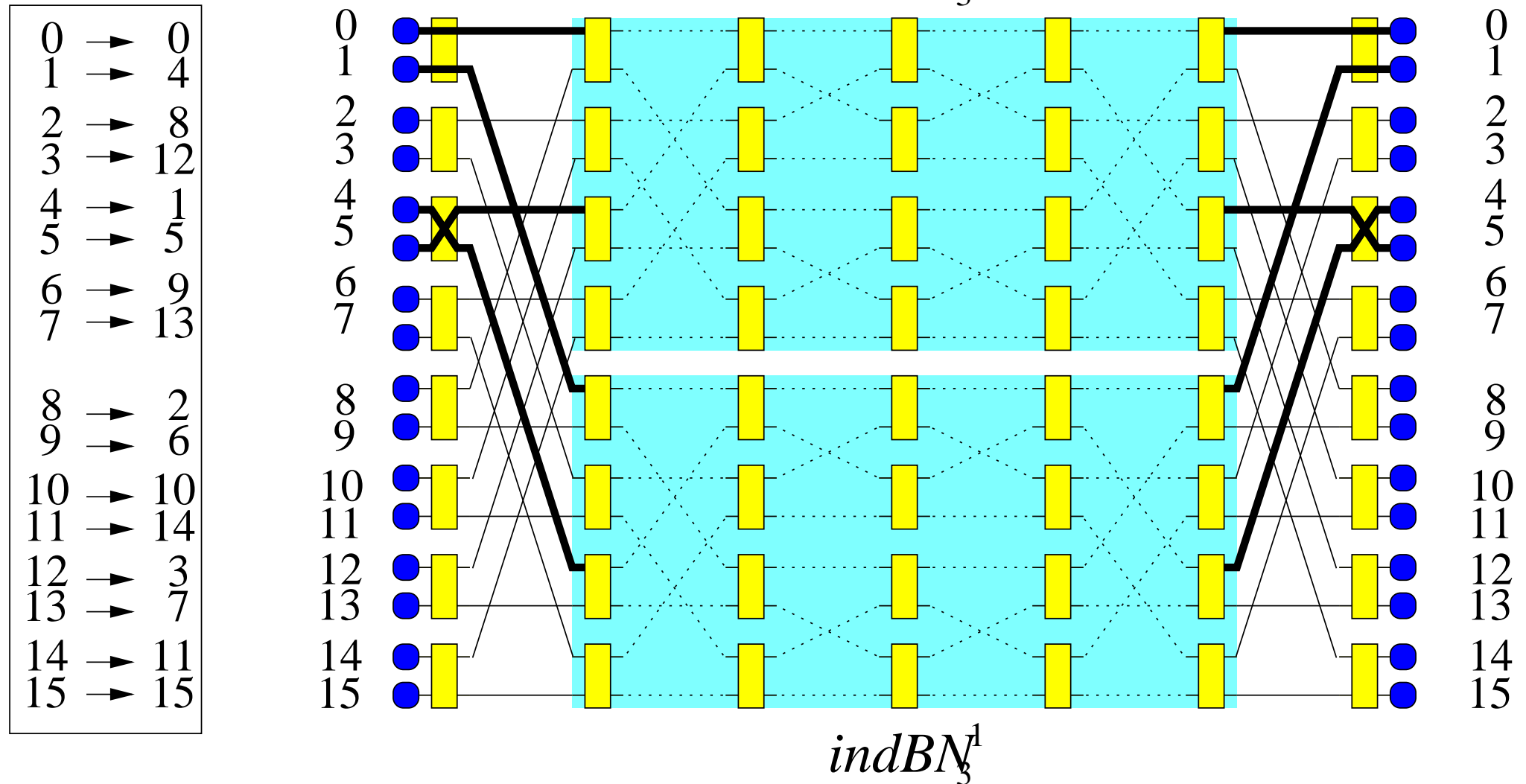


(a) $indBN_4$.



(b) oBN_3 .

Věta 14. Necht' $N = 2^{n+1}$, $\mathcal{N} = \{0, \dots, N - 1\}$, a necht' $\pi : \mathcal{N} \rightarrow \mathcal{N}$ je libovolná permutace. Pak v $indBN_n \exists$ množina N **hranově disjunkttních cest** spojující vstupní kanál i s výstupním kanálem $\pi(i)$ pro všechna $i \in \mathcal{N}$.



Konstrukce prvních úseků hranově disjunkttních cest v $indBN_4$ pro permutaci prohození π_t .

Pozorování:

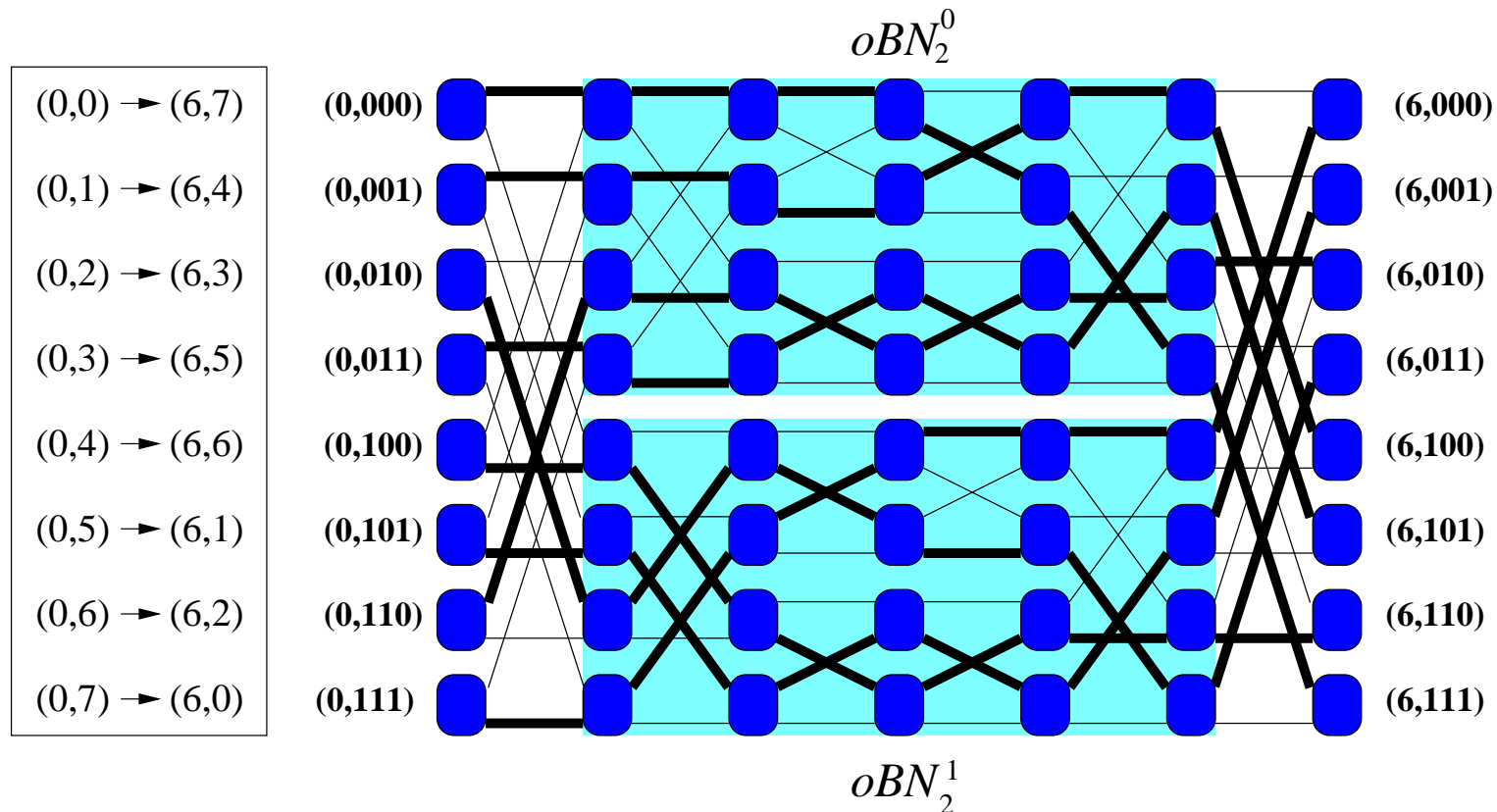
- **Spřažené** vstupní/výstupní kanály i a $j \iff i$ a j se liší v **nejnižším** bitu.
- Podobně se definují **spřažené cesty** $i \rightarrow \pi(i)$ a $j \rightarrow \pi(j)$.
- Každý vst./výst. přepínač má přesně 1 kanál do $indBN_{n-1}^0$ a 1 kanál do $indBN_{n-1}^1$.
- Jestliže daná cesta použije $indBN_{n-1}^0 \implies$ s ní spřažené cesty musí použít $indBN_{n-1}^1$.

Konstrukce (rekurzivní):

1. Tvrzení platí pro $n = 0$. Předpokládejme, že $n \geq 1$ a že tvrzení platí pro $n - 1$.
2. Vezmi jakékoli $i \in \mathcal{N}$ a **zvolme** např. $indBN_{n-1}^0$ pro **vedení cesty** $i \rightarrow \pi(i)$ doprava.
3. Odpovídající výstupní spřaženou cestu vedme **zpátečním směrem** přes $indBN_{n-1}^1$.
4. Jestliže odpovídající spřažená vstupní cesta již byla rozhodnuta, dokončili jsme 1 cyklus permutace a začneme s jakoukoli dosud nezrealizovanou cestou stejně jako v kroku (2).
5. Jinak, pokračujeme doprava přes $indBN_{n-1}^0$ a doleva přes $indBN_{n-1}^1$, dokud neuzavřeme cyklus (tím, že se dostaneme do výchozího vstupního přepínače).
6. !!!! Teprve po nastavení **všech přepínačů v nejlevějším a nejpravějším sloupci**,
známe **indukované** permutace v $indBN_{n-1}^0$ a $indBN_{n-1}^1$ a rekurzivně je vyřešíme. !!!! ♣

Počet různých řešení: závisí na n a počtu cyklů v permutaci π a indukovaných permutacích.

Věta 15. Uzlově-disjunktční směrování. *Nechť $N = 2^n$, $\mathcal{N} = \{0, \dots, N - 1\}$, a nechť $\pi : \mathcal{N} \rightarrow \mathcal{N}$ je libovolná permutace. Pak v oBN_n \exists množina **uzlově-disjunktčních cest**, spojujících dvojice uzlů $(0, i)$ a $(2n, \pi(i))$ pro všechna $i \in \mathcal{N}$.*



Důkaz. Induktivní a konstruktivní jako předchozí, až na jinou definici spřaženosti.

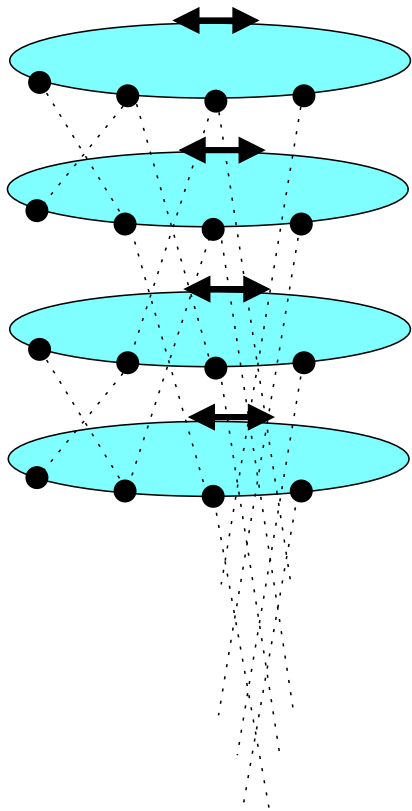
- Vstupní (výstupní) **spřažené** uzly = uzly lišící se v **nejvýznamnějším** bitu.
- **Pozorování:** Jestliže daná cesta vede přes oBN_{n-1}^0 , pak její jak vstupní tak výstupní spřažená cesta vedou přes oBN_{n-1}^1 , a naopak.



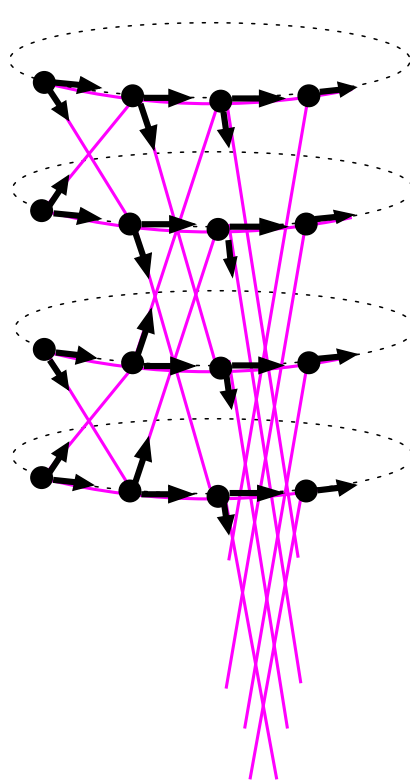
1 průchod skrz $oBN_n = 1$ průchod tam a 1 průchod zpět v oBF_n nebo wBF_n .

Věta 16. Nechť $N = n2^n$ a $\mathcal{N} = \{0, \dots, N - 1\}$. Uvažujme **všeportovou plně-duplexní přímou** síť wBF_n s **nejvýše 1** paketem na 1 uzel. Pak pro **libovolnou** permutaci $\pi : \mathcal{N} \rightarrow \mathcal{N}$ \exists **off-line deterministické permutační** směrování na wBF_n v nejvýše $3n$ krocích a s $\beta = O(1)$.

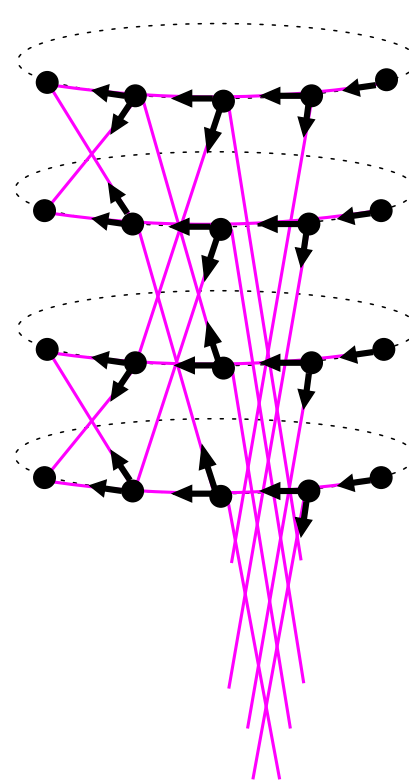
Důkaz. (konstrukční) analogie off-line permutačního směrování na 2-D mřížkách: $oBF_n \approx M(2^n) \times T(n)$ s 2^n horizontálními kružnicemi o n uzlech a n vertikálními sloupci o 2^n uzlech.



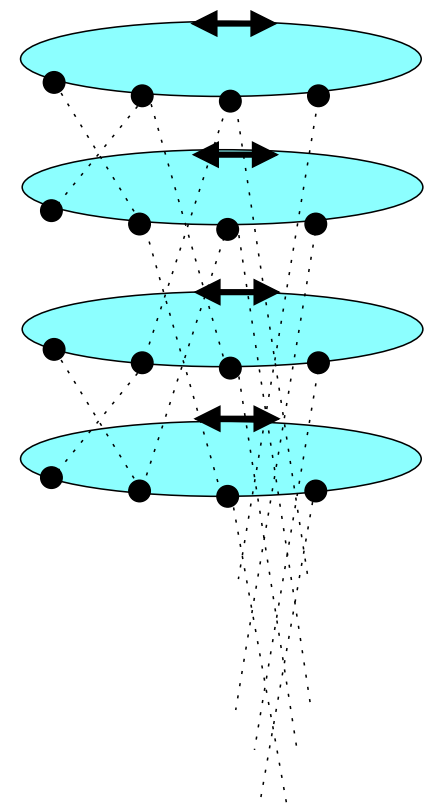
(a) Fáze 2.



(b) Fáze 3, dopředná vlna.



(c) Fáze 3, zpětná vlna.



(d) Fáze 4.

Fáze 1: Předpočítej permutace uvnitř horizontálních kružnic: Hallova věta o párování aplikovaná na n -regulární směrovací bipartitní graf s $|S| = |D| = 2^n$ vrcholy \approx čísla zdrojových a cílových kružnic $\implies n$ dokonalých párování i , kde $i \in \{1, \dots, n\} = n$ množin paketů, která budou srovnány do jednotlivých sloupců i v wBF_n tak, aby adresy všech cílových kružnic paketů v rámci 1 sloupce byly odlišné.

Fáze 2: Paralelně ve \forall horizontálních kružnicích, **proved'** tyto permutace ($n/2$ kroků).
V každém sloupci \exists **nejvýše 1** paket určený pro libovolnou horizontální kružnici.

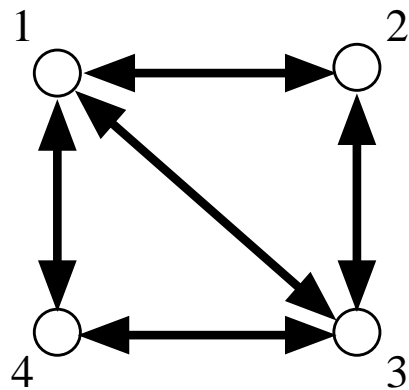
Fáze 3: Paralelně ve \forall virtuálních sloupcích, **permutuj** pakety do správných horiz. kružnic. Protože ve sloupcích wBF_n neexistují vertikální hrany \implies off-line uzlově disjunktní permutační směrování na Benešově síti.

1. Předpočítej off-line uzlově disjunktní cesty pro permutaci každého sloupce zvlášť.
2. $wBF_n =$ uzlově symetrický \implies spusť permutační směrování ve všech sloupcích současně jako n za sebou jdoucích **synchronních vln** tam a pak zpět (n kroků).

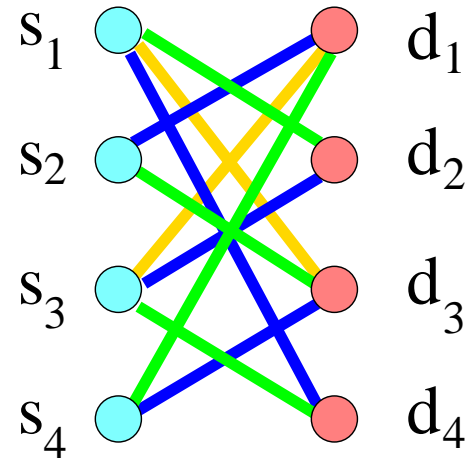
Počet komunikačních kroků je $2n$. Po Fázi 3: \forall pakety jsou v cílových kružnicích.

Fáze 4: Paralelně ve \forall kružnicích, zpermutuj pakety do *správných sloupců* ($n/2$ kroků). ♣

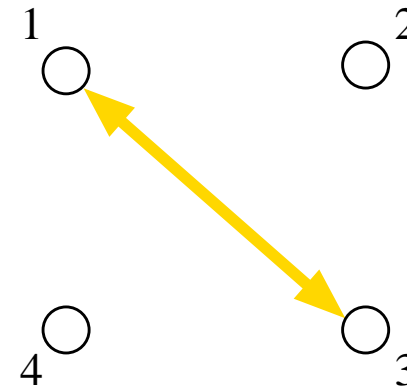
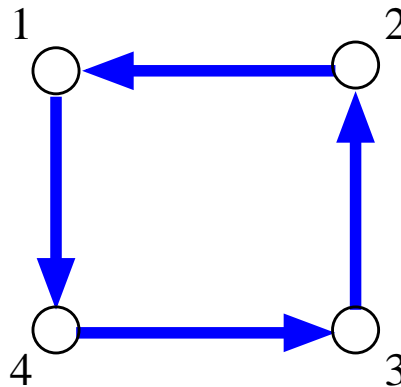
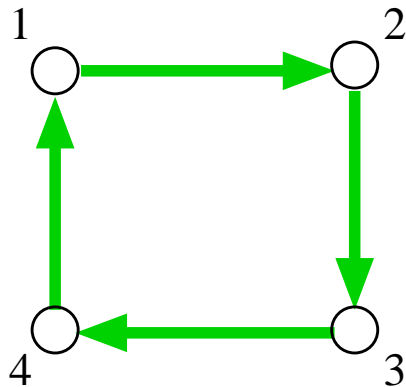
Věta 17. Necht' $N = n2^n$. Necht' G je *libovolná* N -uzlová všeportová plně-duplexní SF propojovací síť s maximálním stupněm uzlu d . Pak přímá všeportová plně-duplexní SF síť wBF_n může simulovat G se zpomalením $O(d \log N)$, je-li dovolen předvýpočet.



1 kom. krok na G



Jeho smerovací graf



Jeho rozklad do castecnych permutaci

- Protože topologie G je libovolná, můžeme použít **libovolné** 1-1 zobrazení $V(G)$ na $V(wBF_n)$.
- Protože stupeň uzlů G je omezen konstantou d , každý uzel v G může poslat nejvýše d paketů a obdržet nejvýše d paketů v 1 komunikačním kroku.

Jeden globální krok komunikace mezi sousedními uzly v G

budeme simulovat pomocí

d permutačních směřování na wBF_n .

- Tato simulace je tudíž serializací: 1 globální krok je serializován do nejvýše d permutací.
- Permutace nejsou nutně úplné, protože G není nutně d -regulární a všechny uzly nepoužívají nutně všechny své kanály v každém komunikačním kroku.



Důsledek 18. *N -uzlový všeportový plně-duplexní wBF_n může simulovat N -uzlovou hyperkrychli se zpomalením $O(\log^2 N)$, je-li dovolen předvýpočet.*

Důkaz. Plyne z předchozí lemmatu, kde G je hyperkrychle velikosti N .

Co je ještě důležitější, z Věty 17 plyne, že hyperkrychle sama je schopna optimálně simulovat jakoukoli topologii s omezeným stupněm uzlu.

Důsledek 19. *Je-li dovolen předvýpočet, pak **libovolnou** $(N \log N)$ -uzlovou síť G s omezeným stupněm uzlu lze simulovat na N -uzlové hyperkrychli se zpomalením $O(d \log N)$, kde d je maximální stupeň uzlu G .*

Důkaz. wBF_n s horizontálními kružnicemi zredukovanými na uzly $= Q_n$. Tudíž,

- každý uzel Q_n simuluje výpočet n uzlů wBF_n
a 1 uzel wBF_n simuluje výpočet 1 uzlu G
 \implies výpočetní zpomalení je $O(n) = O(\log N)$.
- 1 hyperkubická hrana může simulovat $O(1)$ hran wBF_n
a wBF_n simuluje komunikaci v G se zpomalením $O(d \log N)$
 \implies zpomalení komunikace je $O(d \log N)$.