14. Propojovací sítě paralelních počítačů a komunikační algoritmy

Teorie grafů

Následuje plejáda slajdů z 36PAR

Množina uzlů a hran grafu G : $V(G)$, $E(G)$, $N = |V(G)|$

uvažujeme pouze **jednoduché souvislé grafy bez smyček**.

Sousední uzly u a v = **hrana** $\langle u, v\rangle$.

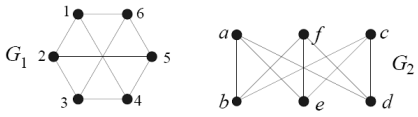
H = **podgraf** grafu G, $H \subset G$: $V(H) \subset V(G)$ a $E(H) \subset E(G)$.

H = **indukovaný podgraf** G : **maximální** podgraf s $V(H)$.

H = **faktor** grafu G : $V(H) = V(G)$.

Klika U_k = **úplný graf** o k uzlech.

Izomorfizmus :$G_1 \equiv G_2$



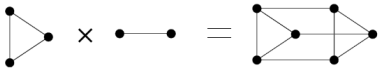
Automorfizmus G : každé izomorfní zobrazení G na sebe sama.

Sjednocení : $G_1 \cup G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$.

Kartézský součin $G = G_1 \times G_2$:

$V(G) = \{(x, y); x \in V(G_1), y \in V(G_2)\}$

$E(G) = \{\langle (x_1, y), (x_2, y)\rangle; \langle x_1, x_2\rangle \in E(G_1)\} \cup \{\langle (x, y_1), (x, y_2)\rangle; \langle y_1, y_2\rangle \in E(G_2)\}$



Výrok 1. *Kartézský součin je komutativní a asociativní operace:* $G_1 \times G_2 \equiv G_2 \times G_1$ *a* $(G_1 \times G_2) \times G_3 \equiv G_1 \times (G_2 \times G_3)$.

Značení: $G \times G = G^2$, $G \times G \times G = G^3$, atd.

Stupeň uzlu u : $\deg_G(u) = \#$ sousedů uzlu u.

Množina stupňů grafu G : $\deg(G) = \{\deg_G(u); u \in V(G)\}$.

Maximální stupeň grafu G : $\triangle(G) = \max(\deg(G))$.

Minimální stupeň grafu G : $\delta(G) = \min(\deg(G))$.

k-**regulární graf** G : $\triangle(G) = \delta(G) = k$.

Řídký graf : $|E(G)| = O(|V(G)|)$ (stupně uzlů jsou omezeny konstantou).

Hustý graf : $|E(G)| = \omega(|V(G)|)$ (stupně uzlů jsou rostoucí funkcí $|V(G)|$).

Uzlově symetrický graf : $\forall\, u_1, u_2 \in V(G)\ \exists$ automorfizmus f takový, že $f(u_1) = u_2$. (a)

Hranově symetrický graf : $\forall\, e_1 = \langle u_1, v_1\rangle, e_2 = \langle u_2, v_2\rangle \in E(G)\ \exists$ automorfizmus f takový, že $f(e_1) = e_2$. (Musí platit i pro $e_2 = \langle v_1, u_1\rangle$.) (b)

Částečně hranově symetrický graf : $\exists$ rozklad $E(G) = E_1 \cup \ldots \cup E_r$ pro $r \geq 2$ takový, že $\forall\, 1 \leq j \leq r\ \forall\, e_1, e_2 \in E_j\ \exists$ automorfizmus f takový, že $f(e_1) = e_2$. (a)+(c)

(a) uzlove sym. (b) hranove sym. (c) castecne hran. sym.

Lemma 2. *(1) Jsou-li G_1 a G_2 uzl. sym., pak $G = G_1 \times G_2$ je také uzl. sym.*
(2) Je-li G hranově symetrický, pak G^k je též hranově symetrický pro každé $k \geq 2$.
(3) Každý hranově symetrický graf je uzlově symetrický.
(4) Každý uzlově symetrický graf je regulární.
(5) Cykly K_k a kliky U_k jsou hranově symetrické.

Procházka, cesta, cyklus v grafu G.

Uzlově disjunktní cesty : $V(P_1(u, v)) \cap V(P_2(x, y)) = \{u, v\} \cap \{x, y\}$

Hranově disjunktní cesty : $E(P_1(u, v)) \cap E(P_2(x, y)) = \emptyset$

Délka cesty $P(u, v)$: $\text{len}(P(u, v)) = \#$ hran v $P(u, v)$.

Vzdálenost uzlů u a v : $\text{dist}_G(u, v) =$ délka nejkratší $P(u, v)$.

Průměrná vzdálenost v N-uzlovém G :

$$\overline{\text{dist}}(G) = \frac{1}{N(N-1)} \sum_{\substack{u,v \\ u \neq v}} \text{dist}_G(u, v).$$

Excentricita uzlu u : $\text{exc}(u) = \max_{v \in V(G)} \text{dist}_G(u, v)$.

Průměr grafu G : $\text{diam}(G) = \max_{u,v} \text{dist}_G(u, v) = \max_u \text{exc}(u)$.

Poloměr grafu G : $r(G) = \min_u \text{exc}(u)$.

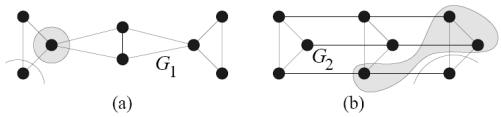
Uzlový řez : množina uzlů, jejichž odebrání způsobí rozpojení grafu.

Hranový řez : množina hran, jejichž odebrání způsobí rozpojení grafu.

(Uzlová) souvislost grafu G : $\kappa(G) =$ velikost **minimálního** uzlového řezu.

Hranová souvislost grafu G : $\lambda(G) =$ velikost **minimálního** hranového řezu.

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$



(a) (b)

(a) $\kappa(G_1) = 1$, $\lambda(G_1) = \delta(G_1) = 2$. (b) $\kappa(G_2) = \lambda(G_2) = \delta(G_2) = 3$.

k-**souvislý graf (hranově** k-**souvislý graf)** : $\kappa(G) = k$ ($\lambda(G) = k$)

Optimální souvislost : $\kappa(G) = \lambda(G) = \delta(G)$

Věta 3. *(Mengerova) Mezi libovolnými 2 uzly G*

$\exists$ nejméně $\kappa(G)$ uzlově disjunktních a nejméně $\lambda(G)$ hranově disjunktních cest.

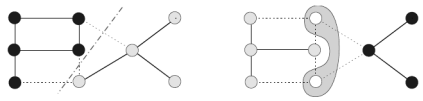
Chybová vzdálenost mezi u a v :
maximum z délek všech **možných co nejkratších uzlově disjunktních** cest mezi u a v.

Chybový průměr : maximum ze všech chybových vzdáleností.

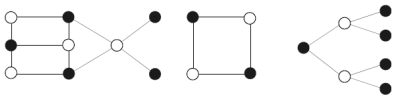
Bisekce

(Hranová) bisekční šířka grafu G, $\mathrm{bw_e}(G)$:
velikost nejmenšího hranového řezu grafu G na **2 poloviny**.

Uzlová bisekční šířka grafu G, $\mathrm{bw_v}(G)$:
velikost nejmenšího uylového řezu grafu G na **2 poloviny** (velikosti nejvýše $\lceil N/2 \rceil$. uzlů)



Bipartitní a vyvážený bipartitní graf :



Hamiltonovská cesta (kružnice) :
cesta (uzavřená) přes všechny uzly (permutace uzlů).

Lemma 4. *Bipartitní graf může mít hamiltonovskou kružnici, jestliže je vyvážený.*

Topologie : množina grafů (instancí topologie), jejichž velikost a struktura je definovaná parametry (**hodnotami dimenzí**).

Hierarchicky rekurzivní topologie : instance nižších dimenzí jsou **podgrafy** instancí vyšších dimenzí.

Inkrementálně škálovatelná topologie : definovaná pro $\forall\ N$.

Částečně škálovatelná topologie : definovaná pro některé, ale ∞ mnoho N.

Efektivně škálovatelná topologie : pro konst. $k > 0$ lze $(N + k)$-uzlovou instanci konstruovat z N-uzlové instance tak, že # odebraných hran je $O(k)$.

Orientované grafy = digrafy :

- **množina orientovaných hran** $A(G)$
- orientovaná hrana $\langle u \rightarrow v \rangle$ je **incidentní z** u **do** v
- $\deg^{\mathrm{in}}_G(u)$ a $\deg^{\mathrm{out}}_G(u)$: **vstupní** stupeň a **výstupní** stupeň uzlu u
- orientovaná cesta
- orientovaný průměr
- silná souvislost

Propojovací sítě pro paralelní počítače (taxonometrie, požadavky na jejich vlastnosti)

Malý a konstantní stupeň uzlu

- Technologický požadavek.
- Stupeň uzlu je omezen konstantou $\implies$ **řídký graf** $\implies$ $|E(G)| = O(N)$.
- Levné a univerzální směrovače $\times$ malá souvislost a velké vzdálenosti.

Malý průměr a malá průměrná vzdálenost

- Algoritmický požadavek.
- Snižuje **komunikační zpoždění** pro
  - jak směrování **citlivé na vzdálenost** (např. store-and-forward),
  - tak směrování **necitlivé na vzdálenost** (např. wormhole) = viz Přednáška 6.

Věta 5. *Spodní mez průměru N-uzlové řídké sítě je $\Omega(\log N)$.*

Důkaz: Je-li dán N-uzlový graf s $\triangle(G) \leq k$, k je konstanta,
$\implies$ # uzlů G dosažitelných nejvýše i kroky je $O(k^i)$
$\implies$ $N = O(k^{\mathrm{diam}(G)})$ $\implies$ $\mathrm{diam}(G) = \Omega(\log N)$.

Konstantní délka hran

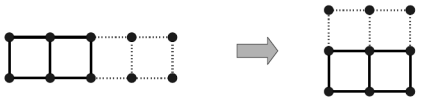
- Rozmístitelnost uzlů v 3-D tak, že délka kabelů je konstantní.

Symetrie

- Návrh paralelních a komunikačních algoritmů je snažší, neboť
  - nezáleží na tom, kde výpočet začne,
  - nezáleží na tom, kterým směrem začne komunikační algoritmus.
- Snažší VLSI návrh a vnořování.

Škálovatelnost

- **Efektivně inkrementálně** topologie: ideální pro dynamicky rekonfigurovatelné multiprocesorové systémy: rekonfigurace systému z N na $N + k$ uzlový vyžaduje $O(k)$ změn v původním propojení.
- Např. 2-D mřížky nejsou efektivně škálovatelné.



Hierarchická rekurzivita

Vysoká souvislost a malé chybové vzdálenosti

- **Redundantní** krátké cesty v případě výpadků uzlů nebo linek.
- **Obcházení** přetížených nebo ucpaných uzlů nebo linek.
- Rozdělení **rozsáhlých paketů** do menších částí posílaných po **paralelních disjunktních** cestách.

Velká bisekční šířka

- Paralelní **binární rozděl-a-panuj** (D&C) algoritmy $\implies$ požadavek **vysoké přenosové kapacity** mezi oběma polovinami:
  - Rozděl řešený problém na dvě poloviny.
  - Řeš je rekurzivně v obou polovinách paralelně s případnou výměnou dat mezi polovinami.
  - Sluč výsledky z obou polovin do konečného výsledku.
- Horní mez je $N/2$, typické hodnoty jsou $N/\log N$ či N^{ε}, $0 < \varepsilon < 1$.
- VLSI návrh: velká bisekční šířka $\implies$ mnoho **externích spojů mezi stavebními moduly**.

Existence hamiltonovských cest a 2-barvení

- Označení procesorů čísly $1, \ldots, p$ zachovávající sousednost.
- Potřebné např. pro třídící algoritmy nebo pro permutace posunu.
- Některé algoritmy používají 2-barvení.

Vnořitelnost jiných a do jiných topologií

- Existence efektivního zobrazení daného grafu procesů do sítě procesorů.
- Schopnost simulovat efektivně jiné topologie.
- Vnořitelnost do VLSI nebo rozmístitelnost v 3-D.

Podpora pro směrování a kolektivní komunikační operace

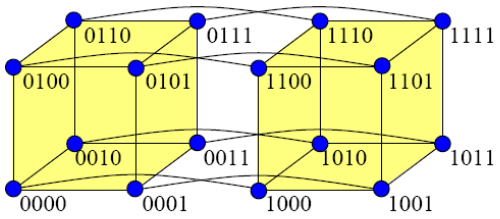
- Dvoubodové minimální směrování.
- Permutační směrování.
- Komunikační operace jeden všem.
- Komunikační operace všichni všem.

Přímé propojovací sítě: Ortogonální a řídké hyperkubické

Ortogonální: hyperkrychle, mřížky, toroidy

Binární hyperkrychle dimenze n, Q_n

Q_4:



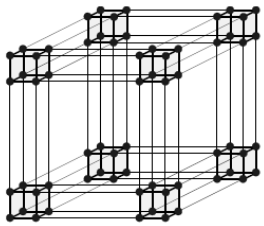
$V(Q_n) = \mathcal{B}^n$ $\quad |V(Q_n)| = 2^n$

$E(Q_n) = \{\langle x, \mathrm{neg}_i(x)\rangle; x \in V(Q_n), 0 \le i < n\}$ $\quad |E(Q_n)| = n2^{n-1}$

$\mathrm{diam}(Q_n) = n$ $\quad \deg(Q_n) = \{n\}$

$\mathrm{bw_e}(Q_n) = 2^{n-1}$

$Q_6 = Q_3 \times Q_3$:



- Regulární graf s **logaritmickým** stupněm uzlů ($\Longrightarrow$ **řídké hyperkubické** sítě).
- **Hammingova vzdálenost** ϱ.
- # uzlů ve vzdálenosti i je $\binom{n}{i}$ $\Longrightarrow$ $\overline{\mathrm{dist}}(Q_n) \doteq \lceil n/2 \rceil$.
- Hierarchicky rekurzivní:
  - $Q_n \equiv Q_p \times Q_{n-p} \equiv Q_p \times Q_q \times Q_{n-p-q} \equiv Q_1^n$.
  - **Kanonická dekompozice** : $Q_n \equiv Q_{n-1}[j=0] \times Q_{n-1}[j=1]$.
  - **Podkrychle** : $s_{n-1}\ldots s_1 s_0$, kde $s_i \in \{0,1,*\}$, $*$ = neutrální symbol.
  - Podkrychle = **termy** v boolské algebře.
- $Q_n \equiv Q_1^n$ $\Longrightarrow$ uzlová a hranová symetrie: $2^n \times n!$ automorfizmů.
  - US: **přeložení** (translace) $u \to v$: zobrazení $\forall x \in V(Q_n)$ $(x \mapsto x \oplus (u \oplus_2 v))$.
  - HS: **rotace**: permutace (přejmenování) dimenzí.
- Částečně, ale efektivně škálovatelná $\Longrightarrow$ **neúplné hyperkrychle**.
- Optimální souvislost: $\lambda(Q_n) = \kappa(Q_n) = n$.
- Největší možná bisekční šířka $\Longrightarrow$ Q_n je ideální pro binární D&C algoritmy.
- Vyvážený bipartitní a hamiltonovský graf.
- Jestliže $\varrho(u,v) = k$, pak v Q_n $\exists$ n uzlově disjunktních cest $P(u,v)$, mezi kterými
  - k cest je délky k a
  - $n-k$ cest je délky $k+2$.
- Chybový průměr Q_n je $n+1$.
- $\exists$ $k!$ různých nejkratších cest mezi dvěma uzly ve vzdálenosti k.
- Minimální e-**cube** směrování: bity v adresách se testují zprava doleva.
- **Optimální** algoritmy pro kolektivní komunikační operace v téměř všech komunikačních modelech.
- **Simuluje efektivně téměř jakoukoli jinou známou topologii**.
- Hlavní **nedostatky** = logaritmický stupeň a nedostatečná škálovatelnost.

n-rozměrná mřížka rozměrů $z_1, z_2, .., z_n$, $M(z_1, z_2, .., z_n)$

$V(M(\ldots)) = \{(a_1, a_2, .., a_n); 0 \le a_i \le z_i - 1 \ \forall i \in \{1, .., n\}\}$

$E(M(\ldots)) = \{\langle (.., a_i, ..), (.., a_i + 1, ..)\rangle; 0 \le a_i \le z_i - 2\}$

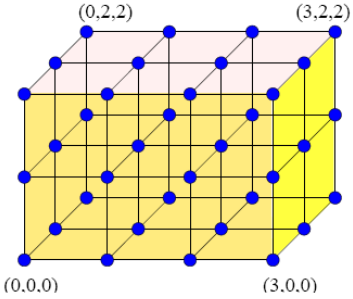
$|V(M(\ldots))| = \Pi_{i=1}^n z_i$ $\quad |E(M(\ldots))| = \Sigma_{i=1}^n (z_i - 1)\Pi_{\substack{j=1 \\ j\neq i}}^n z_j$

$\mathrm{diam}(M(\ldots)) = \Sigma_{i=1}^n (z_i - 1) = \Omega(\sqrt[n]{|V(M(\ldots))|})$

$\deg(M(\ldots)) = \{n, .., n+j\},\ j = |\{z_i; z_i > 2\}|$

$$\mathrm{bw_e}(M(\ldots)) = \begin{cases} (\Pi_{i=1}^n z_i)/\max_i z_i & \text{jestliže } \max_i z_i \text{ je sudé,} \\ \Omega((\Pi_{i=1}^n z_i)/\max_i z_i) & \text{v opačném případu.} \end{cases}$$

mřížka $M(3,3,4)$:



- Předpokládáme, že dimenze mřížky n je **konstantní**.
- $M(k,k,\ldots,k)$ = k-**ární** n-**krychle**.
- Nejpraktičtější mřížky jsou **2-D** (čtvercové, obdélníkové) a **3-D** (krychlové, kvádrové).
- 1-D mřížky = **lineární pole** (inkrementálně škálovatelné) = protipól úplného grafu.
- Není regulární $\Longrightarrow$ není uzlově symetrická.
- Počet uzlů ve vzdálenosti i u 2-D mřížek: od $i+1$ do $4i$.
- Velký průměr: $\sqrt{N} \ge \log N$ od $N > 16$, $\sqrt[3]{N} \ge \log N$ od $N > 1000$.
- Částečně ale neefektivně škálovatelná.
- Hierarchicky rekurzivní:
  - obsahuje podmřížky stejné dimenze, např. $M([1-3], *, [2-5], *) \subset M(6,5,8,3)$,
  - obsahuje podmřížky menších dimenzí, např. $M(*,1,*,3) \subset M(3,4,2,7)$,
  - konstruktor = kartézský součin, např. $M(z_1, z_2, .., z_n) \equiv M(z_1) \times \cdots \times M(z_n)$.
- Optimální souvislost.
- Téměř optimální chybový průměr.
- Chybové vzdálenosti = nejvýše o 4 větší než nechybové.

Poznámky k mřížkám

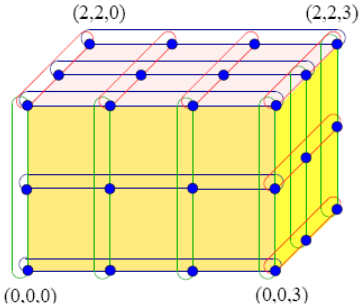
- Problém alokace, fragmentace a zcelování **podmřížek** v multiuživatelském multiprocesoru.
- Komerční paralelní počítače založené na mřížkách: Intel Paragon (2-D mřížka), MIT J-Machine (3-D mřížka), transputery (2-D mřížka), ...
- Pro obecné n je **přesná** hodnota bisekční šířky otevřeným problémem snadno ji lze spočítat pro 2- nebo 3-D mřížky, např. $\mathrm{bw_e}(M(11,8,6)) = 57$.
- Základní minimální směrovací algoritmus = **dimenzně uspořádané** směrování: 2- a 3-D mřížky: **XY** a **XYZ směrování**.
- $\exists$ mnoho komunikačních a paralelních algoritmů **optimálních** vzhledem k **spodním mezím** topologie (velký průměr, malá bisekční šířka).
- Bipartitní (nikoli nutně vyvážená).
- Hamiltonovská, jestliže nejméně jedna strana má **sudou** délku.
- Hamiltonovská cesta existuje vždy.
- Modifikace: **rekonfigurovatelné mřížky, mřížky sběrnic**.

n-rozměrný toroid dimenzí $z_1, z_2, .., z_n$, $K(z_1, z_2, .., z_n)$

toroid = **toroidální** nebo **zabalená mřížka**

$$V(K(\ldots)) = V(M(\ldots))$$
$$E(K(\ldots)) = \{\langle(.., a_i, ..), (.., a_i \oplus_{z_i} 1, ..)\rangle; 0 \le a_i < z_i\}$$
$$|E(K(\ldots))| = n \times \Pi_{i=1}^{n} z_i$$
$$\mathrm{diam}(K(\ldots)) = \Sigma_{i=1}^{n} \lfloor z_i/2 \rfloor$$
$$\deg(K(\ldots)) = \{2n\}$$
$$\mathrm{bw_e}(K(\ldots)) = 2\,\mathrm{bw_e}(M(\ldots))$$

toroid $K(3,3,4)$:

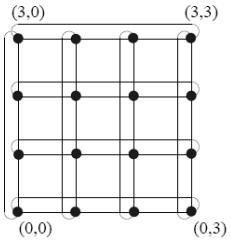


- 1-rozměrný toroid = **kružnice** nebo **prstenec**.
- $K(k,k,..,k) = k$-**ární** n-**toroid**.
- Regulární a uzlově symetrický (automorfizmy = přeložení).
- k-ární n-toroidy jsou i hranově symetrické (automorfizmy = rotace).
- Průměr/průměrná vzdálenost je **poloviční** v porovnání se stejně velkou mřížkou.
- Souvislost/bisekční šířka je **dvojnásobná** v porovnání se stejně velkou mřížkou.
- Částečná škálovatelnost (ještě méně efektivní v porovnání s mřížkami).
- Hierarchický (lze dekomponovat až na **kartézský součin kružnic**) ale: nelze rozložit na stejnorozměrné podtoroidy.
- **Dimenzně uspořádané** minimální směrování (existence kružnic jej činí komplikovanější).
- Hamiltonovský graf.
- Bipartitní $\iff$ všechny délky stran jsou sudé (bipartitní $\implies$ vyvážený).
- Topologicky optimální algoritmy existují pro mnoho základních problémů.
- Komerční MPP: Cray/SGI T3D a T3E, Convex Exemplar (3-D), Intel/CMU iWarp (2-D), KSR (1-D), IBM BlueGene (3-D).

Porovnání hyperkrychlí, mřížek a toroidů

- $M(2,2,..,2)$ je izomorfní s Q_n.
- n-rozměrné mřížky a toroidy jsou zobecněními Q_n.
- Pro určité k a n, $k < n$, k-rozměrná mřížka/toroid může být podgrafem Q_n.

$K(4,4) \equiv Q_4$:



$M(8,8,4) \subset K(8,8,4) \subset Q_8$ a všechny grafy mají $N = 256$.

	$M(8,8,4)$	$K(8,8,4)$	Q_8
diam()	17	10	8
$\lvert E() \rvert$	640	768	1024
$\mathrm{bw_e}()$	32	64	128

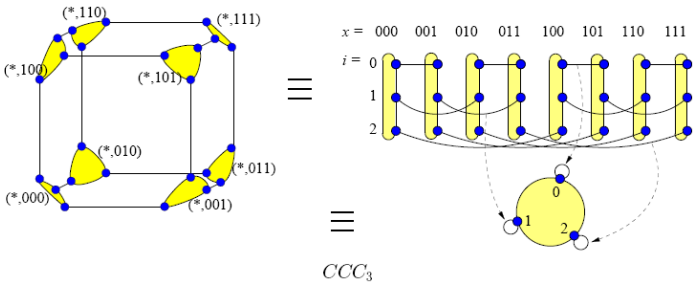
Řídké hyperkubické: Kružnice propojené krychlí a motýlky

Společné vlastnosti:

- $O(1)$ stupeň a $O(\log N)$ průměr,
- škálovatelné hůře než hyperkrychle: $N = n2^n$ nebo podobně,
- bisekční šířka $\Omega(N/\log N)$.

Kružnice propojené krychlí dimenze n, CCC_n

$V(CCC_n) = \{(i,x); 0 \le i < n \wedge x \in \mathcal{B}^n\}$
$E(CCC_n) = \{\langle (i,x),(i \oplus_n 1,x)\rangle, \langle (i,x),(i,\mathrm{neg}_i(x))\rangle; (i,x) \in V(CCC_n)\}$
$|V(CCC_n)| = n2^n$
$|E(CCC_n)| = n2^{n-1} + n2^n$
$\mathrm{diam}(CCC_n) = (2n-2) + \lfloor n/2 \rfloor$ pro $n > 3$, $\mathrm{diam}(CCC_3) = 6$
$\deg(CCC_n) = \{3\}$
$\mathrm{bw_e}(CCC_n) = 2^{n-1}$

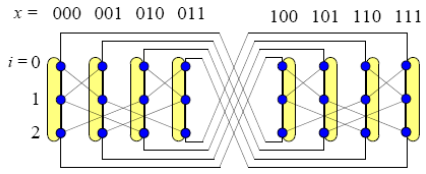


CCC_3

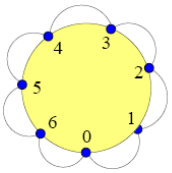
- CCC_n je faktor grafu $Q_n \times K(n)$.
- Q_n i $K(n)$ jsou uzlově symetrické $\implies$ CCC_n je uzlově symetrický.
- Částečně hranově symetrický ($\exists$ **hyperkubické** a **kružnicové** hrany).
- Není hierarchicky rekurzivní.
- Optimální souvislost 3 (mezi dvěma uzly $\exists$ 3 disjunktní cesty).
- Minimální směrování: poněkud komplikované.
- Jednoduché **neminimální** směrování:
  1. Zjisti, v kterých bitech se liší adresa výchozí a cílové kružnice.
  2. Nechť i = pozice prvního takového bitu zleva.
  3. Přesuň se do i-tého uzlu ve výchozí kružnici.
  4. Přejdi pomocí e-cube algoritmu do cílové kružnice.
  5. V ní se přesuň do cílového uzlu.
- Vyvážený bipartitní graf $\iff$ n je sudé.
- Hamiltonovský graf.

Zabalený motýlek dimenze n, wBF_n

$V(wBF_n) = \{(i,x); 0 \le i < n \wedge x \in \mathcal{B}^n\}$
$E(wBF_n) = \{\langle (i,x),(i \oplus_n 1,x)\rangle, \langle (i,x),(i \oplus_n 1,\mathrm{neg}_i(x))\rangle \mid (i,x) \in V(wBF_n)\}$
$|V(wBF_n)| = n2^n$
$|E(wBF_n)| = n2^{n+1}$
$\mathrm{diam}(wBF_n) = n + \lfloor \frac{n}{2} \rfloor$
$\deg(wBF_n) = \{4\}$
$\mathrm{bw_e}(wBF_n) = 2^n$



wBF_3

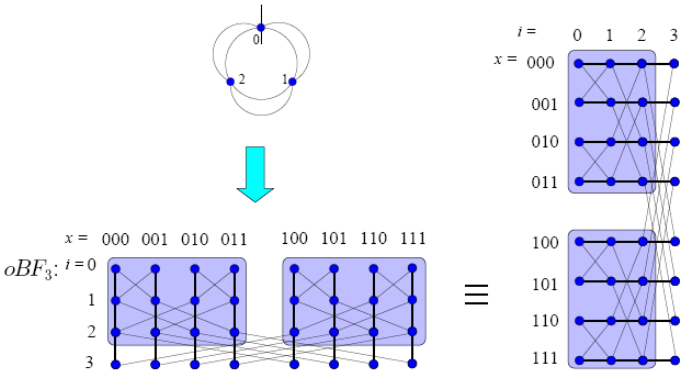


wBF_7

Základní vlastnosti má stejné jako CCC, až na to, že má více hran, větší bisekční šířku a menší průměr.

Obyčejný motýlek dimenze n, oBF_n

$V(oBF_n) = \{(i,x); 0 \le i \le n \wedge x \in \mathcal{B}^n\}$
$E(oBF_n) = \{\langle (i,x),(i+1,x)\rangle, \langle (i,x),(i+1,\mathrm{neg}_i(x))\rangle \mid i < n\}$
$|V(oBF_n)| = (n+1)2^n$
$|E(oBF_n)| = n2^{n+1}$
$\mathrm{diam}(oBF_n) = 2n$
$\deg(oBF_n) = \{2,4\}$
$\mathrm{bw_e}(oBF_n) = 2^n$



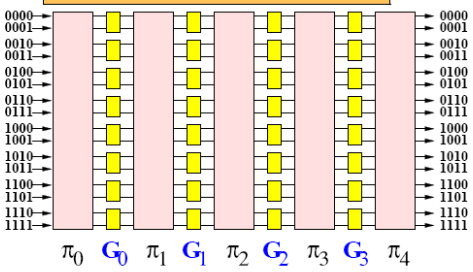
- Uzly oBF_n jsou organizovány do **sloupců** (**stupňů**) $0 \le i \le n$ a **řad** $0 \le x \le 2^n - 1$.
- Dva druhy hran: **přímé** a **křížové** (**hyperkubické**).
- Není uzlově symetrický a není regulární.
- Není hamiltonovský.
- Je **hierarchicky rekurzivní**: oBF_n obsahuje dva oBF_{n-1} jako podgrafy.
- Optimální souvislost 2.
- $\exists$ pouze jediná nejkratší cesta mezi $(0,x)$ a (n,y) (= e-cube směrování).
- Triviálně bipartitní.

Poznámky k řídkým hyperkubickým sítím

- Byly použity v prototypových počítačích.
- Jsou přirozenou topologií pro řadu základních paralelních algoritmů.
- Vynikající vlastnosti pro VLSI implementaci.

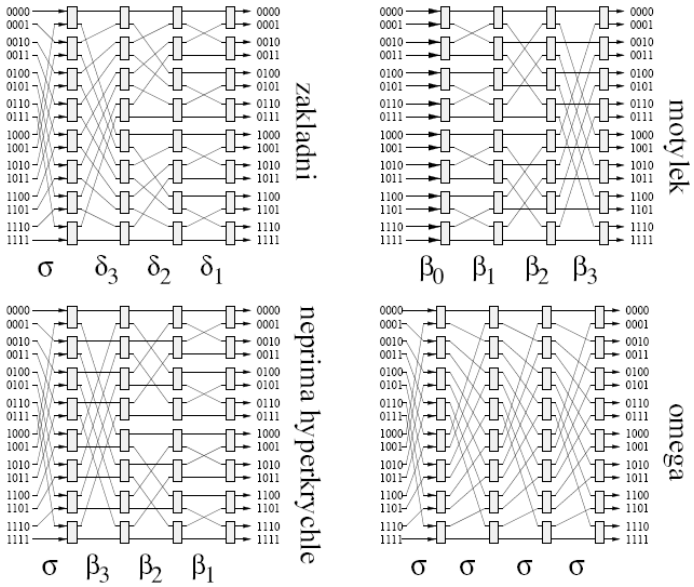
Nepřímé vícestupňové sítě (MIN)

Schéma generické jednosměrné MIN



- Přepínače 2×2 mohou být v 1 ze 4 stavů:

- **Banyan** $N \times N$ **MIN:** $\exists$ jedinečná cesta mezi **lib. dvojicí** vstupu a výstupu.
- k-ární **delta MIN:** Banyan MINs $N \times N$ skládající se ze **stupňů** N/k přepínačů $k \times k$.
- **Obecná** k-ární MIN $= K$ stupňů N/k přepínačů $k \times k$, $N = k^n$.
- Typicky $k = 2$. Pak:
  - Spodní mez na $K = \Omega(\log N) = \Omega(n)$.
  - $K = O(\log N) \implies$ levnější náhrada křížových přepínačů
    - $K = \log N \implies$ **blokující** MIN
    - $K = 2 \log N - 1 \implies$ **přestavitelné** MIN
- **Jednosměrné** nebo **obousměrné**.



(1) σ = **dokonalé promíchání**: $\sigma(\boxed{x_{n-1}} x_{n-2} \dots x_0) = x_{n-2} \dots x_0 x_{n-1}$

(2) β = **motýlek**: $\beta_i(x_{n-1} \dots x_{i+1} \boxed{x_i} x_{i-1} \dots \boxed{x_0}) = x_{n-1} \dots x_{i+1} x_0 x_{i-1} \dots x_i$

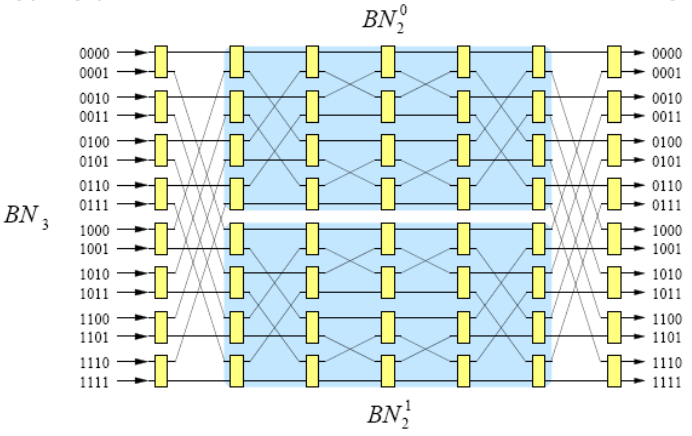
(3) δ = **základní**: $\delta_i(x_{n-1} \dots x_{i+1} x_i x_{i-1} \dots x_1 \boxed{x_0}) = x_{n-1} \dots x_{i+1} x_0 x_i x_{i-1} \dots x_1$

- Počet realizovatelných permutací je menší než $N!$, neboť pro $K = \log N$ je

$$\left(2^{\frac{N}{2}}\right)^{\log N} = N^{\frac{N}{2}} < N! = \sqrt{2\pi N} \left(\frac{N}{e}\right)^N \left(1 + \Theta\left(\frac{1}{N}\right)\right).$$

- Všechny tyto varianty blokujících MIN jsou **topologicky equivalentní**: množina realizovaných permutací je pro všechny varianty stejná, liší se pouze v pořadí, v jakém jsou jednotlivé adresní bity vystavovány na LSB pozici, kde mohou být invertovány.
- Deterministické minimální směrování: mezi daným vstupem a výstupem $\exists!$ cesta (Banyan).

Typický představitel: **Benešova síť** = back-to-back butterfly.



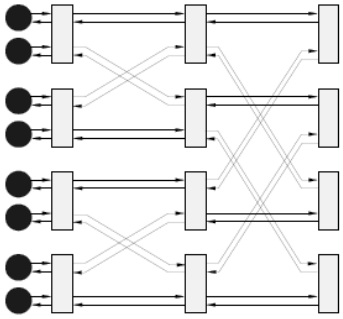
Přestavitelné jednosměrné MIN

- $K = 2 \log N - 1$

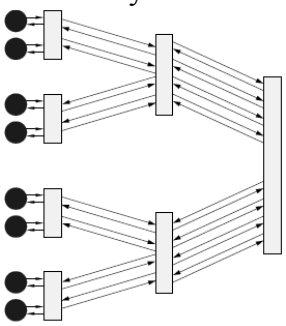
$$\left(2^{\frac{N}{2}}\right)^{2 \log N - 1} > N!$$

- Pro zadanou permutaci vstupů na výstupy lze předpočítat bezkolizní nastavení přepínačů.
- Typický představitel: **Benešova síť** = back-to-back butterfly.

Obousměrný MIN



Tlustý strom



Problém vnoření:

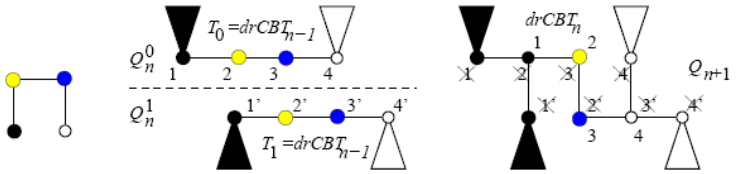
- ***Load*** – počet uzlů, které jsou simulovány jedním procesorem
- ***Dilatace*** – počet hran, na které je namapována jediná původní hrana
- ***Expanze*** – kolik původních hran prochází jedinou hranou propojovací sítě

■ **Nejlepší a krásně rekurzivní** vnoření $CBT_n \xrightarrow{\text{emb}} Q_{n+1}$ s $\boxed{\text{dil} = 2 \text{ a load} = \text{ecng} = 1}$.
Trik: přeměň CBT_n na **vyvážený** bipartitní graf $drCBT_n$ s 2^{n+1} uzly

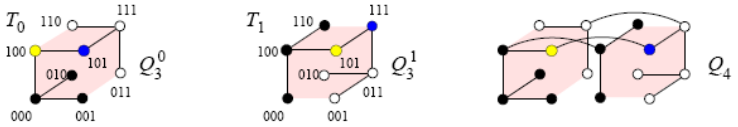
zdvojením kořenu.

Potom

$drCBT_n$ **je faktorem** Q_{n+1}.



(a) Indukční základ. (b) Indukční hypotéza v Q_n^0 a automorfismus v opačné Q_n^1. (c) Indukční krok.



(a) $T_0 \xrightarrow{\text{emb}} Q_3^0$. (b) $T_1 \equiv T_0 \xrightarrow{\text{emb}} Q_3^1$. (c) Indukční krok.

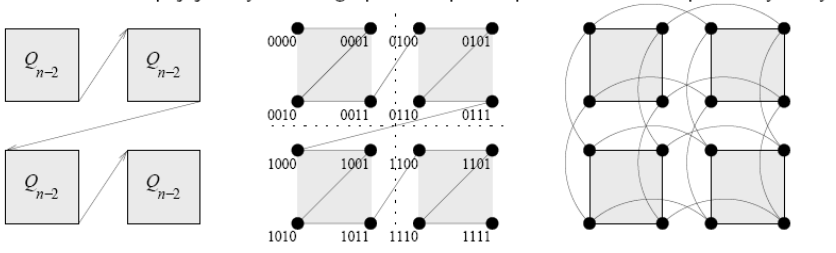
1. Kořen stromu je umístěn do libovolného uzlu hyperkrychle.
2. Každý uzel x vnoreného stromu
   - zná dimenzi hyperkrychle $n \geq 1$
   - zná svou hyperkubickou adresu $\varphi(x)$
   - zná číslo své hladiny $l(x)$ ve stromu,
   - vypočte si číslo cesty $t(x) = l(x) \pmod n$, reprezentované jako n-bitové slovo s jedinou 1 na pozici $t(x)$
   - má ***náhodný generátor*** svého ***flip bitu*** $fb(x)$.
3. Jestliže listový proces x stromu vnorený do '(x) porodí jednoho nebo 2 syny, vygeneruje fb(x) a provede následující kroky:

if $(\text{fb}(x) = 0)$ **then** umísti levého syna (pokud existuje) do svého uzlu $\varphi(x)$;
umísti pravého syna (pokud existuje) do uzlu $\varphi(x) \,\text{XOR}\, t(x)$;
else umísti levého syna (pokud existuje) do uzlu $\varphi(x) \,\text{XOR}\, t(x)$;
umísti pravého syna (pokud existuje) do svého uzlu $\varphi(x)$.

Věta 9. *Pro strom s M uzly a pro hyperkrychli s N uzly, tento algoritmus dává* $\text{dil} = 1$ *a* $\text{load} = O(M/N + \log N)$ *s vysokou pravděpodobností.*

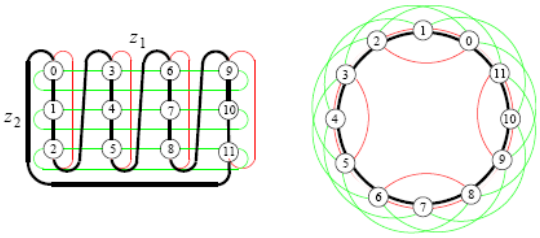
Vnoření hyperkrychlí do mřížek

Peanova křivka: spojuje uzly v lexikographickém pořadí při dělení střídavě podle osy x a y.

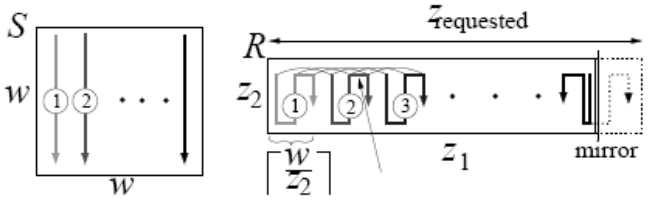


(a) Indukční krok. (b) φ vnoření $Q_4 \xrightarrow{\text{emb}} M(4,4)$. (c) ξ vnoření $Q_4 \xrightarrow{\text{emb}} M(4,4)$.

Vnoření toroidů do toroidů

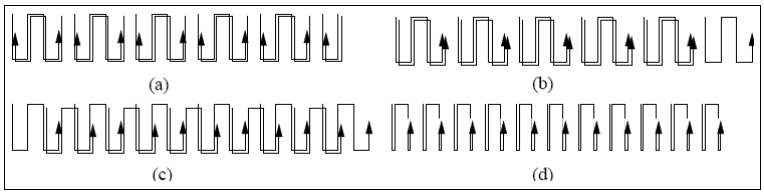


Vnoření čtvercových mřížek do obdélníkových mřížek

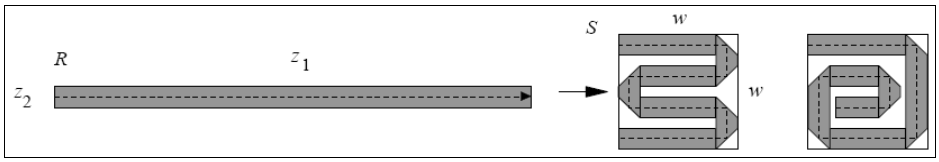


nebo s lepší expanzí:

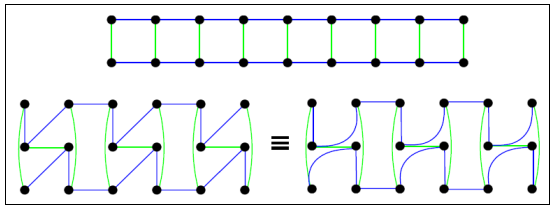
1) Vlož zrcadlo doprostred R
2) Použij stejné hady, ale zdvojuj je
3) Použij stejné hady, ale částecne je prekrývej.
4) Použij užší hady (= hady s load=2)



Vnoření obdélníkových mřížek do čtvercových mřížek

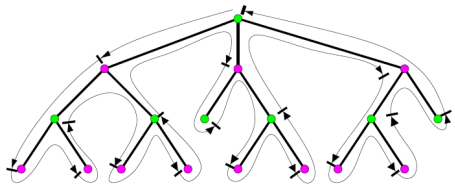


Vnořování mezi 2D obdélníkovými mřížkami



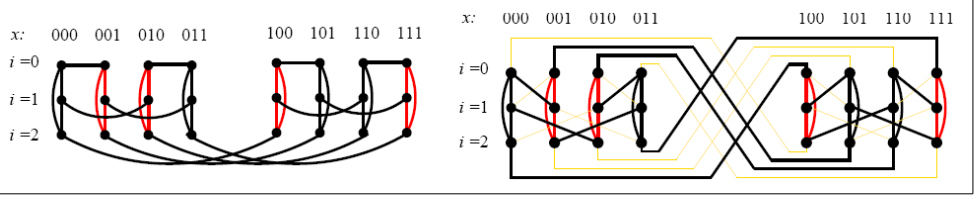
Vnoření lineárního pole (kružnice) do libovolného grafu

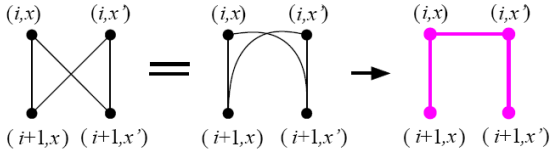
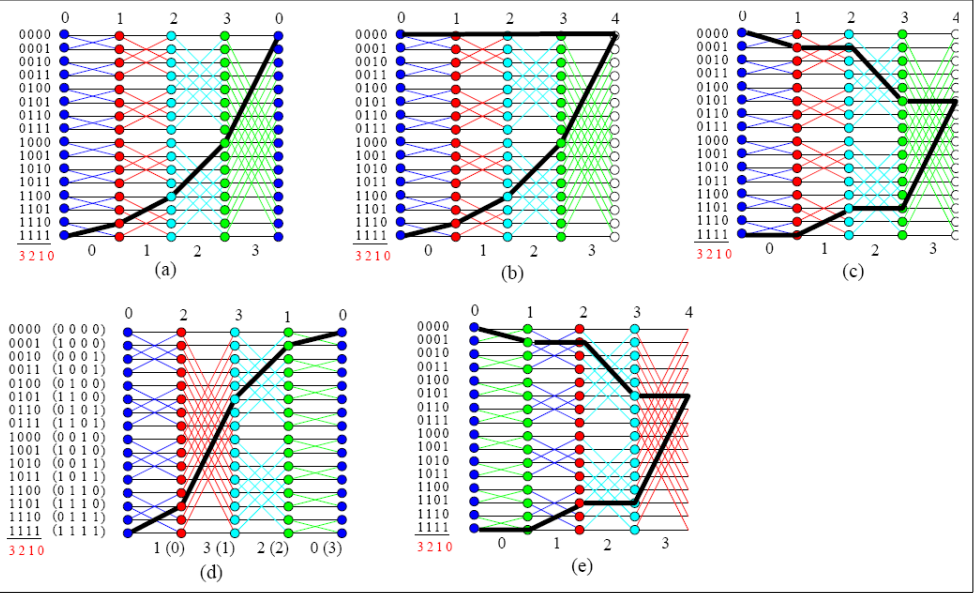
1. vytvoříme kostru



CCC a oba typy motýlků jsou výpočetně ekvivalentní (důkaz – 4 části)

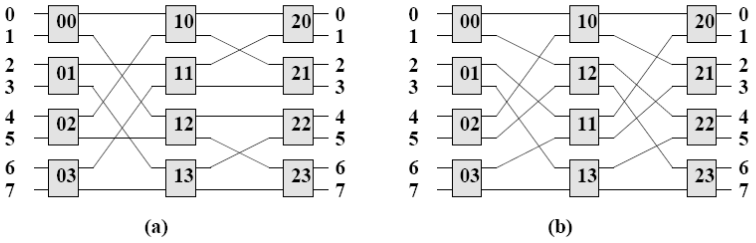
CCC_3 **je faktorem** $\varpi\, BF_3$



ϖBF_n do CCC_n	
oBF_n do ϖBF_n	Sloučením koncových uzlů řad oBF_n dostaneme kružnice ve ϖBF_n
ϖBF_n do oBF_n	

Lemma 23. *Delta sítě stejného typu a počtu a velkosti stupňů jsou izomorfní.*
Důkaz. Všechny tyto sítě se liší pouze v pořadí, v jakém jsou bity adresy vystavovány inverzi, každý bit právě jednou.

Příklad 24. Izomorfismus MIN sítí motýlek (a) a Omega (b).



Komunikační algoritmy

Latence L – průměrná doba, za kterou bode doručena zpráva

směrovací algoritmy

- on line / off line - on line - rozhoduje pouze podle lokální situace
- deterministický/nedeterministický – cesta při det. směrování záleží jen na počáteční koncové adrese (necitlivé směrování)
- greedy / non greedy – greedy - nejkratší cesta
- adaptivní / neadaptivní – adaptivní - záleží na stavu sítě
- distribuované / centralizované

zpráva logická jednotka komunikace, různá délka

- packet – část zprávy, pevná délka
- flit – nejmenší jednotka, pevná délka

t_S SW a HW zpoždení v zdrojovém a cílovém uzlu nutné pro zformátování paketu, validaci dat a kopírování dat mezi frontou a pamětí směrovače

$t_m = 1/q$ zpoždení mezi sousedními smerovaci na 1 fit (v $[s/B]$

m délka zprávy ve slabikách

t_d per-hop latency - doba než začátek zprávy dorazí k dalšímu uzlu
d délka cesty

Přepínání kanálů – sonda projede celou cestu a nastaví směrovače. Cesta sondy k cíli trvá $d(t_r + p(t_w + t_m))$. Potvrzení délky 1 poté putuje zpět. Cesta trvá $d(1(t_w + t_m))$, protože se již nenastavují směrovače. Jakmile dorazí potvrzení, okamžitě se začíná odesílat samotná zpráva. Celkový čas potřebný pro přenesení zprávy délky m bude $\boxed{t_{CS} = d(t_r + (p+1)(t_w + t_m)) + m t_m}$.

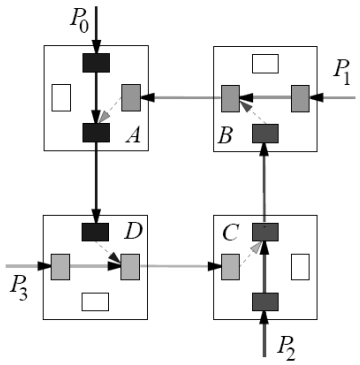
Store and forward – zpráva/packet procházející přes uzel je nejprve celá uložena, pak celá odeslána. Paket je rozdělen do flitů. Každý paket je směrován individuálně ze zdroje do cíle. Směrovací rozhodnutí jsou činěna až poté, co byl celý paket uložen ve vstupní frontě směrovače. Každý směrovač musí určit směrovací rozhodnutí a pak teprve celý paket přeskočí do dalšího směrovače, což trvá $t_r + m(t_w + t_m)$, a tento postup se opakuje d krát. $\boxed{t_{SF} = d(t_r + m(t_w + t_m))}$

Wormhole routing – Zpráva putuje po jednotlivých flitech od zdroje do cíle. Směrovače mají vyrovnávací paměti pouze na jeden, či několik málo **flitů** $\boxed{t_{WH} = d(t_r + t_w + t_m) + m\,max(t_w, t_m)}$

Průřezové přepínání (VCT) – První flit v paketu ví kam má jít celý paket, ostatní jdou za ním jako ovce. Jakmile se dostane takovýto první flit do vstupní fronty, okamžitě nastavuje směrovač a jede dál. Nečeká se až dorazí celý paket. Z pochopitelných důvodů nelze multiplexovat více paketů naráz, ale musí být nejprve odeslán celý paket.

Problém zablokování a jeho řešení (graf kanálových závislostí, neblokující směrování, virtuální kanály)

Zablokování - Obecně: skupina paketů nemuže učinit žádný pokrok, protože každý z nich už některé prostředky zabírá, ale pro další postup potřebuje prostředky, držené jiným agentem a tento řetěz požadavků tvoří cyklus. Na zablokovanou část se nabalují další čekající pakety (efekt sněhové koule).



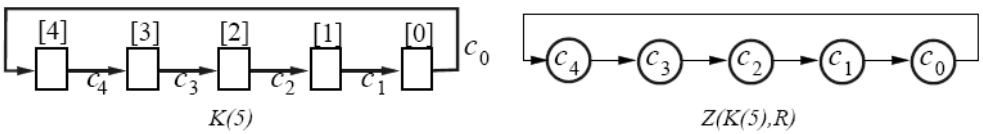
Ilustrace 1: Zablokování 4 paketů ve WH síti

Řešení zablokování v ortogonálních topologiích:

- Detekce a zotavení: nejméně opatrné, možný velký zisk, ale i ztráty.
- Prevence: konzervativní přidělování všech prostředků najednou
  $\Longrightarrow$ jejich malé využití – použito v technice přepínání kanálů (CS).
- Vyhnutí se zablokování: postupné přidělování prostředků tak, aby globálně nemohlo k zablokování dojít (viz dále).

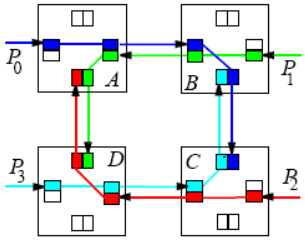
Definice 2. *Graf kanálových závislostí* $Z = Z(G, R)$*:*

- *uzly* $V(Z)$ = *kanály* c_i *sítě* G,
- $\langle c_1, c_2 \rangle \in E(Z) \iff R$ *může v* G *směrovat paket z kanálu* c_1 *na kanál* c_2, *t.j., pro nějaké dva uzly* u *a* d *platí* $R(u, c_1, d) = c_2$.



Věta 3. *Deterministická směrovací funkce* R *na grafu* G *nemůže vést k zablokování* $\iff$ $Z(G, R)$ *je acyklický.*

- Omezení směrovací funkce R na R' tak, aby $Z(G, R')$ byl acyklický a G zůstal při použití R' (silně) souvislý.
- Funguje v mřížkách a hyperkrychlích:
  - Uspořádání (seřazení) dimenzí (směrů).
  - R': po použití kanálu v dané dimenzi se může použít pouze kanál stejné nebo menší dimenze.
  - Příklady R': XY směrování v 2-D mřížkách, XYZ v 3-D mřížkách, e-cube v hyperkrychlích.
- Příklad: Vyhnutí se zablokování 4 paketů



$P_0 \to C$	$P_1 \to D$	$P_2 \to A$	$P_3 \to B$

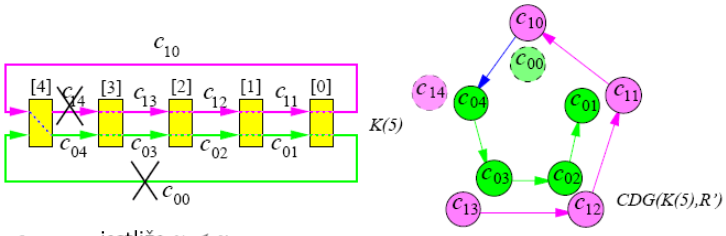
ve WH 2-D mřížce s XY směrováním

Zablokování v toroidu:

Pro jednoduchost si představme jednorozměrný toroid (kružnici). Nepomůže nám ani pokud se rozhodneme pohybovat pouze jedním směrem (např proti směru hod. ručiček), viz. například cyklický posuv o 2, který již způsobí zablokování. Řešení:

Každý fyzický kanál nahradíme $k \geq 2$ virtuálními kanály, které se budou na fyzický kanál spravedlivě multiplexovat na bázi *flitů* (nikoliv celých paketů). Fyzický kanál musí být schopen rozlišovat mezi flity jednotlivých paketů. Existuje věta, která říká, ze pro každý toroid libovolné dimenze stačí 2 virtuální kanály pro každý fyzický kanál (oba ve stejném směru). Pokud cesta vede od nižšího uzlu k vyššímu (uzly jsme si v kruhu označily podle obrázku), nižší kanál není vůbec využit. V opačném případě se projde přes c_{10} do uzlu 4, odkud se přejde do nižších kanálů.

Obecně:



- $c_{1u}, \ldots, c_{10}, c_{0(z-1)}, \ldots, c_{0(v+1)}$ jestliže $u < v$,
- $c_{0u}, \ldots, c_{0(v+1)}$ jestliže $u > v$.

, kde u,v jsou uzly

Nepravidelné sítě: Algoritmy typu Nahoru*/Dolů*

Definice: Souvislý graf G je **korektní**, pokud lze uvalením orientace na jeho hrany zkonstruovat **orientovaný** graf (digraf) G' takový, že:

1. existuje jediný kořen = uzel, ze kterého nevycházejí orientované hrany,
2. neexistují orientované kružnice (G' je acyklický digraf).

ALGORITHM CORRECTORIENT(G, r)

Předpoklady: Každý uzel má jednoznačné ID a kořen r je určen.

Fáze 1: Zkonstruuj **kostru do šířky** $T(r)$ s kořenem r použitím standardního distribuovaného algoritmu.

Fáze 2: Orientuj každou hranu $\langle u, v \rangle \in E(G)$

směrem ke kořenu r	pokud $\text{depth}_{T(r)}(u) \neq \text{depth}_{T(r)}(v)$,
směrem k uzlu s menším ID	v ostatních případech.

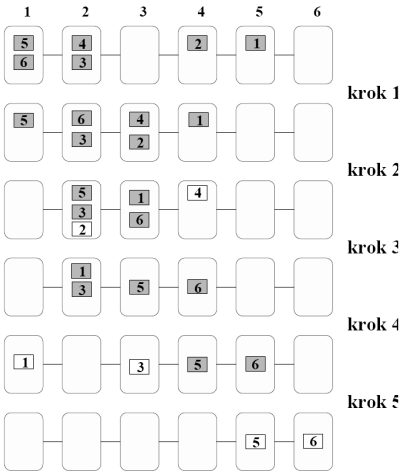
Důkaz, že algoritmus CORRECTORIENT generuje korektní graf:
Každý vnitrní uzel T(r) má nejméne 1 hranu orientovanou smerem ke korenu a pouze koren nemá žádnou. Protože každý uzel má jednoznačné ID, v G' neexistuje orientovaná kružnice, protože hrany jsou orientované pouze směrem ke kořenu nebo uzlům s nižším ID.

Algoritmy pro permutace (přímé, náhodné, založené na třídění, s předvýpočtem)

Čas od času je třeba v paralelním počítači data mezi jednotlivými procesory nějak vyměnit. Ne každý procesor se musí takové výměny účastnit. Každý uzel je zdrojem nebo příjemcem nejvýše jednoho paketu. (***neúplná permutace***). ***Úplná permutace*** = každý je zdrojem a příjemcem právě jednoho paketu.

Ortogonální topologie:

1D mřížka (lineární pole)



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.*

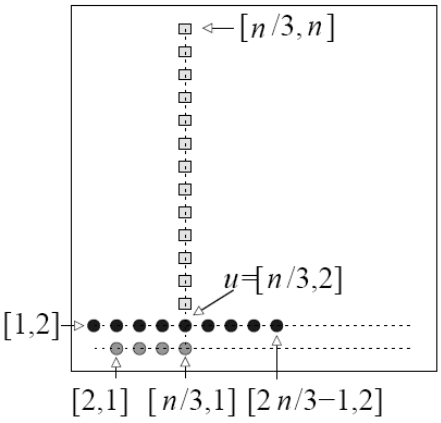
Pak řízení toku pomocí strategie Farthest-First (FF) lze tuto komunikaci provést v nejvýše $n-1$ *krocích.*

(Farthest-First = první půjde ten paket, který to má nejdál)

Důkaz: Je-li v nějakém uzlu připraveno více paketů týmž směrem, strategie FF upřednostní vždy paket s nejvzdálenějším cílem, nejen na počátku, ale i kdykoliv během výpočtu... z toho to je myslím už jasný. *Na slajdech je důkaz přes matematickou indukci. Obrázek:*

2D mřížka (mřížka)

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 $2n-2$ krocích, což je optimální, ale směrovače potřebují fronty na $\beta = \max(2, 2n/3-1)$ paketů.*



uvažujeme opět SF.

Závěr: potřebujeme fronty délky 2/3 velikosti mřížky..přibližně

Důkaz: můžeme si představit jako dlouhé chodby pro směrování v dimenzi X a jakýsi páter noster v každém uzlu, který má kapacitu 1 paket. V nejhorším případě k páternosteru dorazí jeden paket zleva, jeden zprava a jeden přijede zdola. Předpokládejme, že všichni chtějí jet nahoru.

- max. # soupeřících paketů = # cílových uzlů $= n-2$. (v nejhorším možném případě, kdy jede $n-2$ lidí z druhého patra nahoru – viz obrázek)
- všech $n-3$ paketů může do u dorazit v $n/3-1$ krocích

$\Longrightarrow$ u musí dočasně uložit $1+n-3-(n/3-1) = 2n/3-1$ paketů.

(všechno co přišlo mínus to, co jsem odeslal)

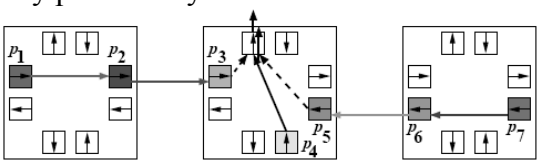
Vícerozměrná mřížka (zobecnění předchozího)

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

$n-2$ je maximální počet paketů, které mohou být kolizní

$\frac{n-3}{2k-1}$ minimální počet kroků, během kterých je tam mohu nashromáždit

Protichůdné požadavky na paměť a na čas – pakety P1, P2, P6 a P7 chtějí směřovat rovně, zatímco P4, P4 a P3 nahoru. Prostřední směrovač může obsloužit pouze jeden z paketů, tedy řekněme paket P4. Pakety P3 a P5 čekají ve frontě a přitom blokují P1, P2, P6 a P7. Pokud by prostřední směrovač měl vlastní paměť, do které by pakety P3 a P5 mohl uložit, byl by problém vyřešen.



Metody minimalizace paměťových požadavků – obecně obtížné, obvykle se užívají 3 postupy:

1) randomizace
2) permutační směrování na seřazení paketů
3) permutace s předvýpočtem

Randomizace – 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 $\pi : \{1, \ldots, p\} \to \{1, \ldots, p\}$)
for all $i := 1, \ldots, p$ **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.}

- není spolehlivé (může se stát, že více uzlů vygeneruje stejnou adresu)
- důkaz netriviální, založen na teorii pravděpodobnosti

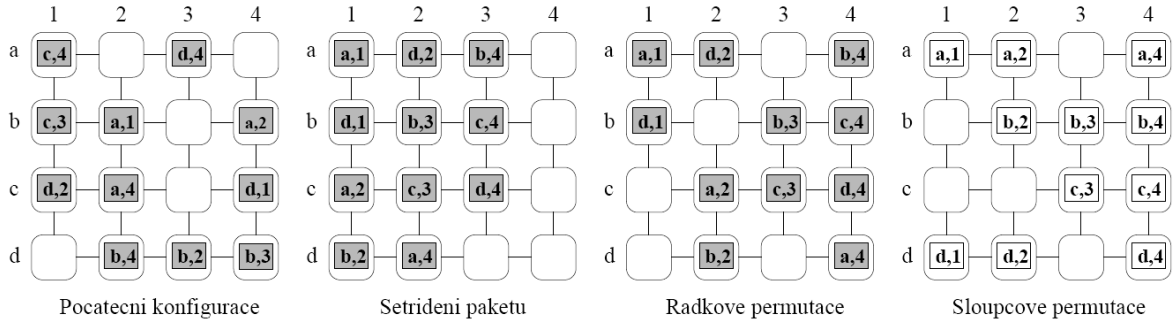
Věta 4. *Jakoukoli permutaci v $M(n,n)$ lze pomocí randomizace realizovat v $2n + o(n)$ krocích s $\beta = O(\log n)$ frontami s pravděpodobností aspoň $1 - O(1/n^2)$.*

Náznak důkazu:

Algorithm MeshRandPerm (**In:** permutation $\pi : \{1, \ldots, n^2\} \to \{1, \ldots, n^2\}$)
{**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. }

Online permutační směrování založené na řazení

Předpokládáme, že máme mřížku, každý paket má svojí cílovou adresu. Procesory pracují s paketem jako s položkou, která má klíč (adresa cílového uzlu). Podle klíče pakety seřadí. Složitost takového řazení je: $T_{\text{sort}}(M(n,n)) = 3n + o(n)$.Třídíme lexikograficky nejprve podle čísel sloupců. Protože řazení se provádí do vertikálního hada (následující obrázek ukazuje horizontálního hada), musíme každý druhý sloupec překlopit (obr b). Je vidět, že v setříděné mřížce nejsou v žádném řádku pakety se stejným sloupcem. Nyní tedy každý paket směřuje do jiného sloupce. Pakety se v řádcích přesunou do správného sloupce (obr c). V posledním kroku nastane opět bezkolizní permutace v každém ze sloupců (proč bezkolizní vyplývá z permutace v 1D mřížce – viz začátek podkapitoly)



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 $M(n,n)$.

, trvá to tedy tolik, jako setřídění mřížky do hada + $3n$ (převrácení sudých řádků v setříděném hadovi, řádkové permutace, soupcové permutace). Paměť není potřeba.

Offline permutační směrování – Dosud jsme předpokládali, že počáteční konfigurace není nikde centrálně známá. Celková permutace v offline známá všem, nebo alespoň někomu. Výhodné, když se nějaká permutace často opakuje.

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: Algorimus má také 4 fáze (druhé dvě shodné s minulým příkladem). Směrovací graf je regulální (každý uzel má stejný počet hran), bipartitní. Arita uzlu je 4 ⇒ teoreticky by mělo jít, aby každý uzel

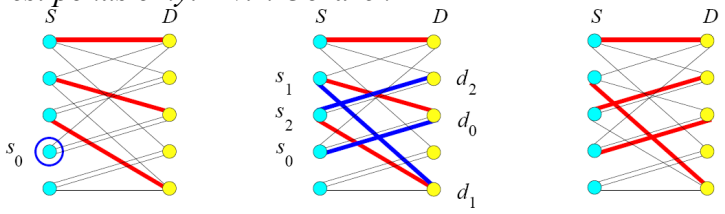
Algorithm OfflineMeshPerm (**In:** permutation $\pi : \{1, \ldots, n^2\} \to \{1, \ldots, n^2\}$)
Fáze 1: Ve $\forall$ sloupcích, *předpočítej* permutace takové, aby v každém řádku $M(n,n)$ $\exists$ nejvýše 1 paket určený pro daný sloupec.
Fáze 2: Realizuj tyto permutace ve všech sloupcích paralelně. $(* \; n-1$ 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 $*)$

měl 4 hrany o 4 různých barvách. Úkolem je, jak nalézt takové barvení:

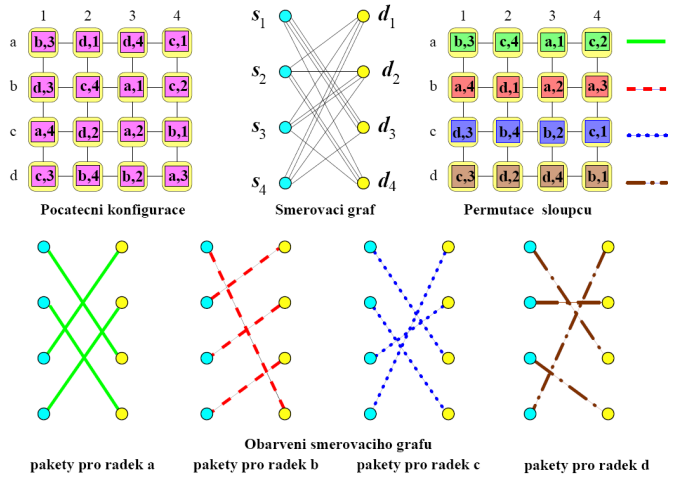
$s_1 \ldots s_4$ reprezentují zdrojové sloupce v mřížce

$d_1 \ldots d_4$ reprezentují cílové sloupce v mřížce

každý paket je reprezentován jednou hranou. Např. paket c,3, který směřuje do a,2 je reprezentován hravou $[s_3, d_2]$. Pakety přeneseme do odpovídajících řádků (zelené pakety do zelených řádků). V žádném řádku nejsou 2 pakety se stejným číslem sloupce. V sloupci nesmí vzniknout 2 pakety se stejnou barvou. Pokud tedy vyřešíme barvení, máme směrovací tabulku. Barvení se řeší pokus omyl – viz. Obrázek



1 dokonalé párování ≈ n hran s barvou i ≈ n paketů přesunutých do řádku i



Hledáme červené barvení: vyplníme první 3 hrany víceméně od oka, ale čtvrtá hraja již nemá žádnou možnost. Rozhodneme se tedy obarvit hranu do d0, což však nás nutí odstranit hranu směřující do d0. S1 si musí hledat nového partnera, vybere si třeba d1, což připraví o partnera uzel s2, který má ale možnost jit do d2. Tímto je iterace ukončena. (pozn. Že hrany, které rušíme jsou červené, hrany, které vytváříme jsou modré. Po ukončení hledání odstraníme červené hrany na cestě [S0, D0, S1, D1, S2, D2] a modré přebarvíme na červené). Protože v této cestě je vždy o jednu modrých víc než čevených, přibyde nám ve výsledku jedna hrana.

Hyperkubické topologie

Online permutační směrování na síti motýlek

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

- Otočení $\pi_r : u_{n-1} \ldots u_0 \mapsto u_0 \ldots u_{n-1}$.
- Prohození $\pi_t : u_{n-1} \ldots u_k u_{k-1} \ldots u_0 \mapsto u_{k-1} \ldots u_0 u_{n-1} \ldots u_k$, je-li $n = 2k$

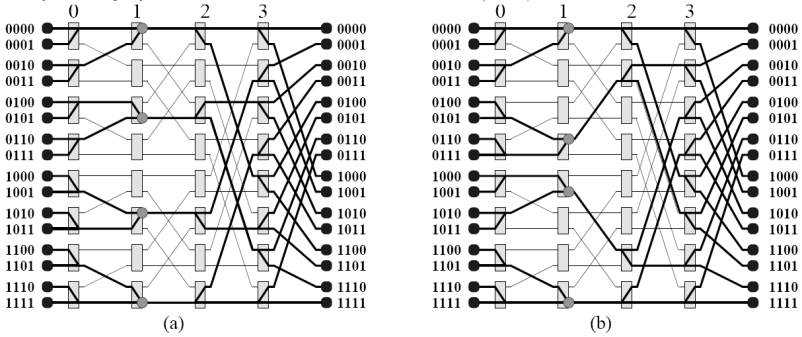
  $\pi_t : u_{n-1} \ldots u_{k+1} u_k u_{k-1} \ldots u_0 \mapsto u_{k-1} \ldots u_0 u_k u_{n-1} \ldots u_{k+1}$, je-li $n = 2k+1$
- Přeložení (posun) do w, $\pi_p^w : i \to i \,\mathrm{XOR}\, w$ pro daný n-bitový řetězec w.
- Doplněk $\pi_d : u_{n-1} \ldots u_0 \mapsto \overline{u}_{n-1} \ldots \overline{u}_0$ (= speciální případ přeložení).
- Cyklický posun o δ, $\pi_s^\delta : i \mapsto i \oplus_p \delta$, $i \in \{1, \ldots, p\}$.

Nejhorší permutace pro motýlka:

Otočení a prohození

- spodní mez je rovna průměru sítě (log n) ale té nelze dosáhnout (jsme řádově u odmocniny)
- vytvoří se mosty, přes které chce přejít mnoho paketů
- paměťové nároky lze eliminovat „pravidly slušného chování“ = pokud je plno, tak sem nelez.

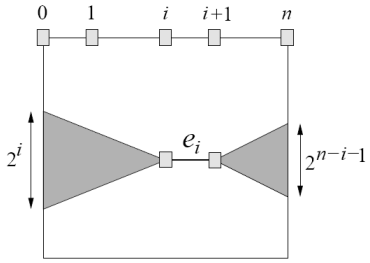
Věta 8. *Permutace otočení π_r a prohození π_t ve všeportovém SF nepřímém motýlku $indBF_n$ potřebuje při minimálním on-line směrování $\Theta(\sqrt{N})$ kroků, kde $N = 2^n$.*



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

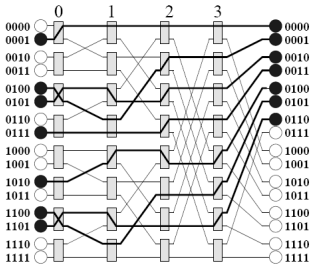
- důkaz, že žádná jiná permutace nemůže být asymtoticky horší, než tyto dvě:

- 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 $S = \Sigma_{i=1}^{n}(g_i - 1)$
- $S \leq \begin{cases} 3\sqrt{N/2} - n - 2 & \text{je-li } n \text{ liché,} \\ 2\sqrt{N} - n - 2 & \text{je-li } n \text{ sudé.} \end{cases}$ ♣

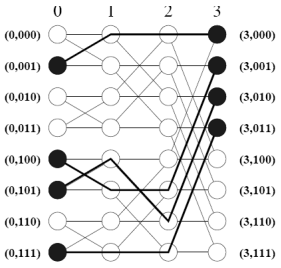


Permutace pro motýlka jako stvořené:

Zhušťovací permutace:

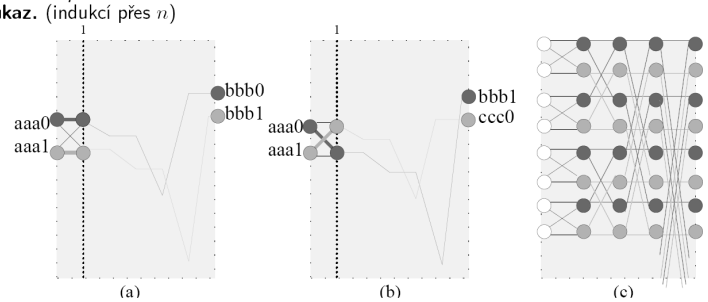


(a) $indBF_4$, $m = 7$.



(b) oBF_3, $m = 4$.

Věta 11. *Minimální směrování na oBF_n poskytuje uzlově disjunktní cesty pro libovolnou zhušťovací permutaci.*
Důkaz. (indukcí přes n)



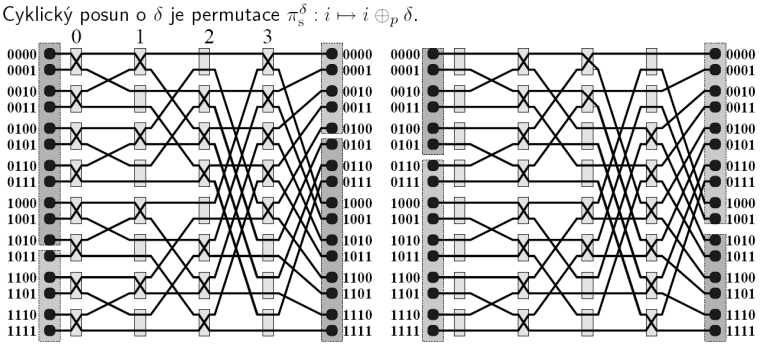
- Žádné 2 pakety se nemohou srazit na uzlu v sloupci 1, viz jediné dva možné případy na obr. (a) a (b). (dva za sebou následující pakety zůstanou těsně za sebou i na výstupech, pouze se můžou posunout a tudíž změnit paritu (hodnotu posledního bitu)).
- Indukce: na žádném dalším stupni nemůže dojít k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

Zřeďovací permutace – pouze otočené zhuštění

Monotónní permutace = složení zhušťovací a zřeďovací permutace

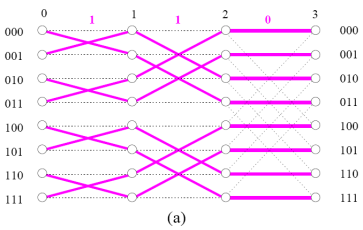
Cyklický posun

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



Permutace přeložení a doplněk

Věta 13. *Minimální směrování na oBF_n poskytuje pro permutaci přeložení π_p^w, $0 \leq w \leq N-1$, uzlově disjunktní 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 ž žádnou hranu.



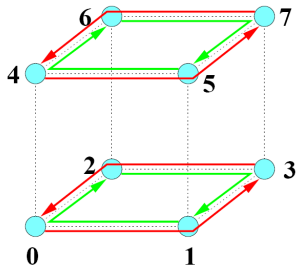
Bezkolizní směrování pro permutaci přeložení π_p^w pro $w = 011$ na oBF_3.

Permutace doplněk π_c $\implies$ všechny pakety používají ve všech stupních pouze křížové hrany.

Permutační směrování na binární hyperkrychli

28

- 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 disjunktní permutační směrování na SF oBF_n
  $\iff$ permutační směrování s nulovým zahlcením na WH plně-duplexní všeportové Q_n.

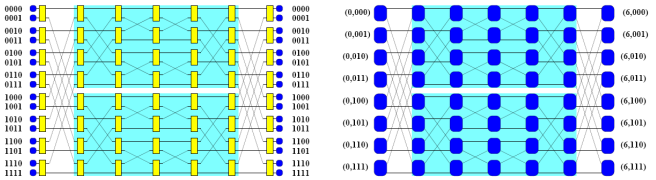


Přeložení π_p^w pro $w = 011$ na WH plně-duplexní všeportové Q_3 používající e-cube směrování.

Off-line permutační směrování na Benešově síti

- 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^0_{n-1} a spodní BN^1_{n-1}.
- Přestavitelná úplná permutační síť:

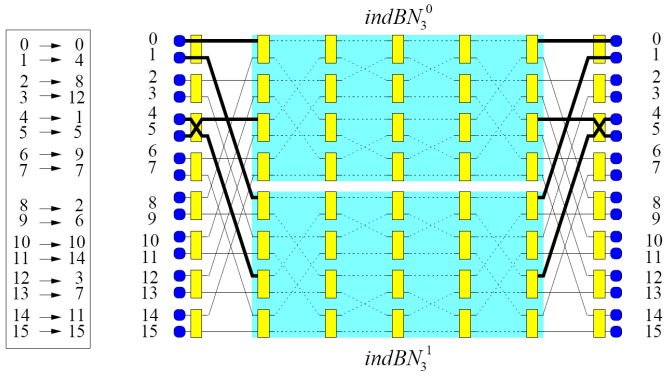
  pro libovolnou permutaci vstupů na výstupy, $\exists$ bezkolizní permutační směrování.
- $\exists$ 2 varianty: přímá oBN a nepřímá $indBN$



(a) $indBN_4$. (b) oBN_3.

Off-line hranově disjunktní permutační směrování

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



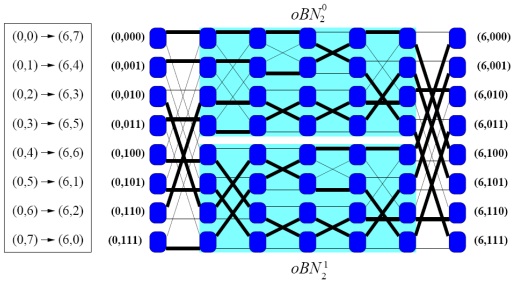
Konstrukce prvních úseků hranově disjunktních cest v $indBN_4$ pro permutaci prohození π_t.

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. $indBN^0_{n-1}$ pro vedení cesty $i \to \pi(i)$ doprava.
3. Odpovídající výstupní spřaženou cestu veďme zpátečním směrem přes $indBN^1_{n-1}$.
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 $indBN^0_{n-1}$ a doleva přes $indBN^1_{n-1}$, 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 $indBN^0_{n-1}$ a $indBN^1_{n-1}$ a rekurzivně je vyřešíme. !!!! ♣

Off-line uzlově disjunktní permutační směrování

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

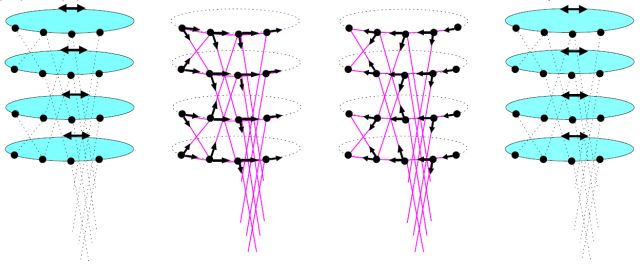


Off-line optimální směrování pro libovolnou permutaci na oBF_n

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 $\mathcal{N} = \{0, \ldots, N-1\}$. Uvažujme všeportovou plně-duplexní přímou síť wBF_n s nejvýše 1 paketem na 1 uzel. Pak pro libovolnou permutaci $\pi : \mathcal{N} \to \mathcal{N}$ $\exists$ 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.

Důkaz

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 $i \in \{1, \ldots, n\} = n$ 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 , proveď 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.

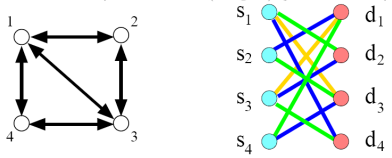
1. Předpočítej off-line uzlově disjunktní cesty pro permutaci každého sloupce zvlášť.
2. wBF_n = uzlově symetrický $\implies$ spusť permutační směrování ve všech sloupcích současně jako n za sebou jdoucích synchronních vln tam a pak zpět (n kroků).

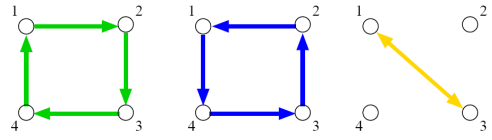
Počet komunikačních kroků je $2n$. Po Fázi 3: $\forall$ pakety jsou v cílových kružnicích.

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

Off-line simulace libovolné topologie na oBF_n

Věta 17. *Nechť $N = n2^n$. Nechť 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.*





Důkaz

- 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 směrování 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.

Důsledky 37

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
  $\Longrightarrow$ 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 $O(d \log N)$
  $\Longrightarrow$ zpomalení komunikace je $O(d \log N)$. ♣

Komunikační operace

Typy směrovacích úloh

1) one to one
- single pair – pouze jeden uzel posílá druhému
- permutační – každý posílá 1 zprávu a každý 1 dostane
- vícenásobné – každý posílá max.1 zprávu a každý max.1 dostane
- multicasting – permutační směrování pouze na skupině uzlů

2) kolektivní
- one to all broadcast (OAB) - jedenposílá stejnou zprávu všem ostatním
- one to all scatter (OAS) - jeden posílá všem, každému jinou zprávu
- all to one gather (AOG) - všichni posílají zprávu jednomu
- all to all broadcast (AAB) - všichni všem, každý uzel dělá OAB
- all to all scatter gather (AAS)- všichni všem, každý dělá OAS

3) synchronizační
4) globální
- redukce – společně vyprodukují jednu hodnotu
- scan – společně vyprodukují pole hodnot

Komunikační modely

- simplex komunikace pouze jedním směrem
- poloduplex přepínání mezi komunikací v jednom a druhém směru
- plný duplex komunikace v obou směrech zároveň
- 1-portový model procesor v jednou okamžiku komunikuje s jediným sousedem
- všeportový model procesor komunikuje se všemi sousedy zároveň
- k-portový procesor komunikuje s k sousedy zároveň
- kombinující procesor složí procházející zprávy do jediné a tu odešle
- nekombinující není kombinující

[*opravdu nemá smysl jinak suplovat tuhle prednášku... takže doporučuju prolítnout slajdy, který přikládám*]