Přednáška #6: **Směrování, techniky přepínání a zablokování**

Klasifikace komunikačních problémů

Komunikace jeden-jednomu : Informace je pouze vyměňována (nikoli duplikována).

- jedna komunikující dvojice:
  - základní algoritmy pro minimální směrování: přednáška 4
  - žádné problémy se zablokováním či zahlcením
- více komunikací typu jeden-jednomu : několik komunikujících dvojic: dnešní přednáška
- permutační směrování: relace zdroj-cíl = permutace uzlů: Přednáška 9

Komunikace jeden-mnoha: 1 zdroj a mnoho cílů: přednáška 10

- vysílaní ve skupině (*multicast*, MC)
- vysílání jeden-všem (*one-to-all broadcast*, OAB)
- rozesílání jeden-všem (*one-to-all scatter*, OAS)

Kolektivní komunikace všichni-všem : všechny uzly = zdroje i cíle: přednáška 11

- vysílání všichni-všem (*all-to-all broadcast*, AAB)
- rozesílání všichni-všem (*all-to-all scatter*, AAS)

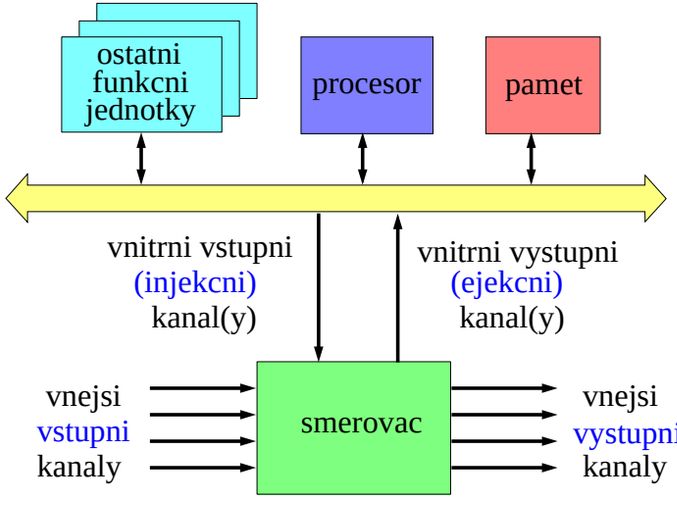
Základní parametry návrhu propojovacích sítí

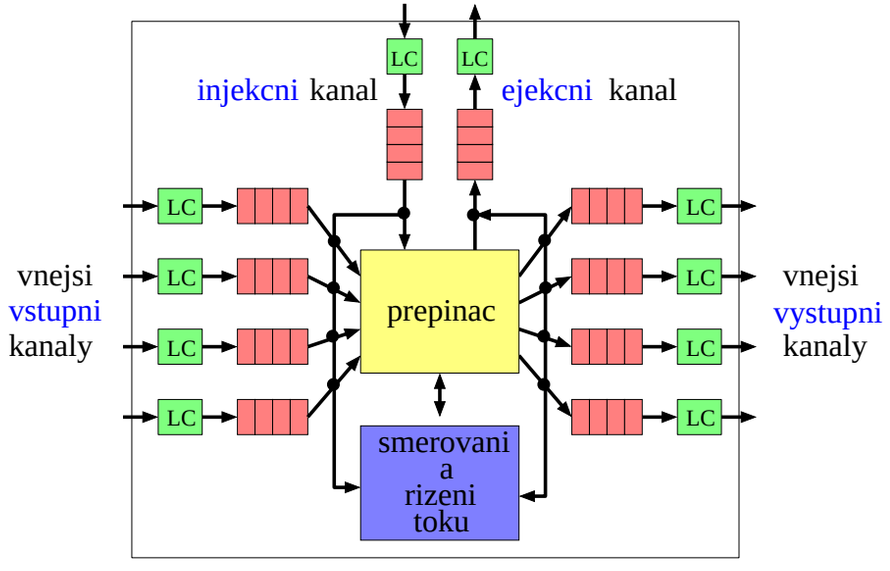
- Topologie: určuje jak jsou uzly spolu propojeny kanály (minulé 2 přednášky).
- Směrovací algoritmus: určení trasy ze zdrojového uzlu do cílového uzlu = posloupnost kanálů (hran, linek) $c_1, c_2, c_3, \ldots$.
- Řízení toku: mechanismy pro přidělování sdílených prostředků (kanálů a pam.front) sítě zprávám/paketům.

Analogie ze života:

- topologie = silniční síť a její popis pomocí mapy
- směrovací algoritmus = řidič auta
- řízení toku = semafory, policisté, odstavná parkoviště/pruhy

Typická architektura směrovače





Směrovač: HW koprocesor (přepínač, kanály, jednotka pro směrování a řešení konfliktů)

Kanál: fronty zpráv, linkové kontroléry (LC) a komunikační médium (např. koaxiální kabel)

Vnější kanály: propojují směrovače mezi sebou a definují topologii propojovací sítě

- 1-portový směrovač
- všeportový směrovač
- výstupně všeportový směrovač

Sousední uzly: uzly s přímo propojenými směrovači

Vnitřní kanály: implementují fyzické HW rozhraní směrovač $\leftrightarrow$ lokální procesor

- 1-portový procesor: 1 injekční a 1 ejekční kanál
- k-portový procesor: k injekčních a k ejekčních kanálů
- vše-portový procesor: # injekčních/ejekčních kanálů = # vnějších výstupních/vstupních kanálů

Přepínač: propojuje vstupní kanály na výstupní

Fronta: FIFO paměť pro jednu nebo několik jednotek komunikace

- směrovače s frontami na vstupu i výstupu
- směrovače s frontami pouze na vstupu
- směrovače s frontami pouze na výstupu

Směrovost kanálů: jednosměrné, poloduplexní, plně duplexní

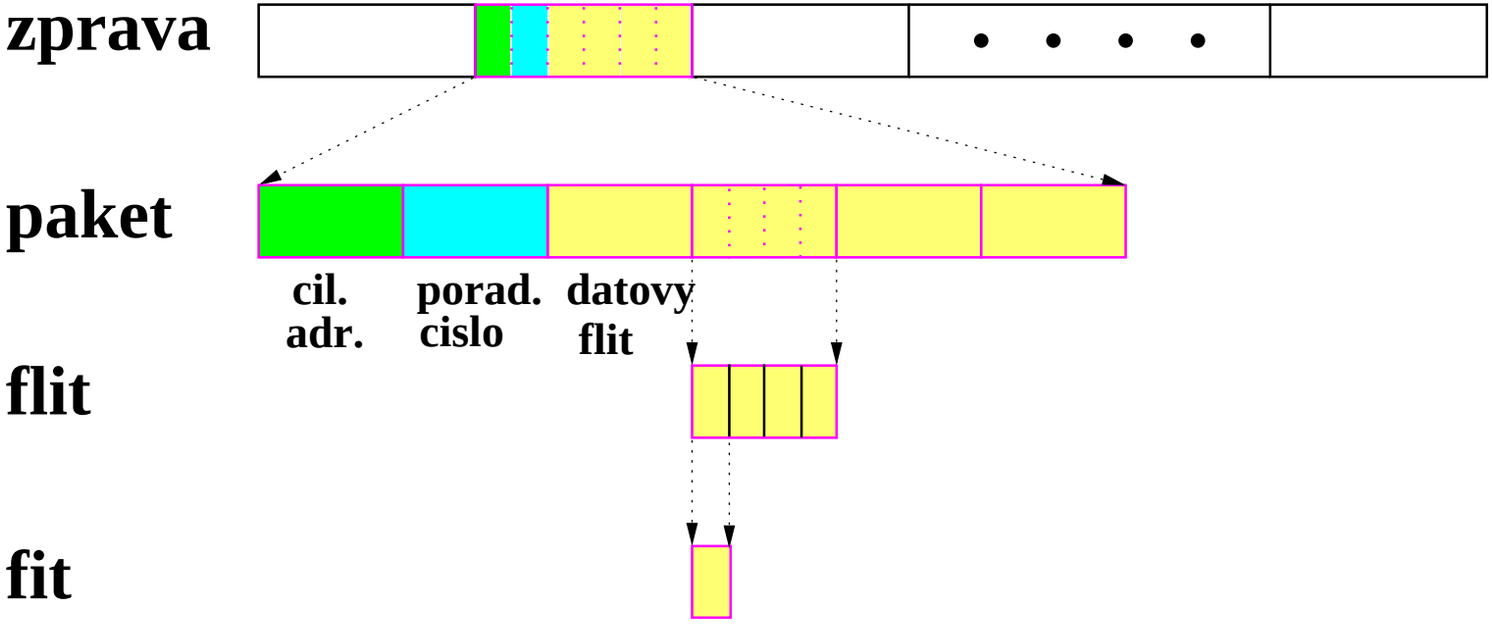
Jednotky komunikace

Zpráva: jednotka komunikace z hlediska programu, proměnná délka

Paket: jednotka komunikace se směrovací informací pevné délky.
Skládá se z hlavičkového flitu a datových flitů.

Flit: jednotka komunikace na linkové vrstvě. Flity jsou 1 nebo několik slov dlouhé, mohou být několika typů a linkový protokol přenosu flitů typicky vyžaduje několik cyklů.

Fit: nejmenší jednotka komunikace na fyzické úrovni, která je přenesena přes 1 fyzickou linku v 1 cyklu.



Směrovací algoritmus

- Směrovací algoritmus se skládá ze:

  směrovací relace R**,** která vrací množinu možných výstupních kanálů (tras) a **výběrové funkce** ρ**,** která z této množiny vybírá 1 položku.

- Otázky zablokování souvisejí primárně s R, otázky adaptivity primárně s ρ.

- Příklady ($V =$ množina uzlů, $C =$ množina kanálů, $P =$ množina cest, $\mathcal{P} =$ potenční množina):

$$R : V \times V \mapsto \mathcal{P}(P)$$
$$R : V \times V \mapsto \mathcal{P}(C)$$
$$R : C \times V \mapsto \mathcal{P}(C)$$

Klasifikace směrovacích algoritmů

1. Rozhodování o směrování
2. Adaptivita
3. Minimálnost
4. Progresivnost
5. Implementace směrovacích algoritmů

1. Kdy a kde je provedeno směrovací rozhodnutí

Distribuované (inkrementální) směrování: směrovače počítají směrování z cílových adres v hlavičkách, tedy $R : C \times V \mapsto \mathcal{P}(C)$ nebo $R : V \times V \mapsto \mathcal{P}(C)$.

- Symetrické nebo regulární topologie $\Longrightarrow$ všichni směrovače používají týž směrovací algoritmus.

Zdrojové (*all-at-once*) směrování: zdrojové uzly předurčí úplné trasy před vložením paketů do sítě, tedy $R : V \times V \mapsto \mathcal{P}(P)$.

- Směrovače pouze čtou/značkují/ustřihávají směrovací informace.
- Je-li $k = \#$ výstupních kanálů na 1 směrovač a $\delta =$ délka trasy
  $\Longrightarrow$ velikost hlavičky $= \delta \log k$ bitů (IBM SP-2).
- Křižovatkové (*street-sign*) směrování: (hlavně pro ortogonální topologie)
  - implicitní směr = přímý,
  - hlavička obsahuje dvojice: explicitní směrovka a adresa jejího uzlu.

Hybridní (vícefázové) směrování:
- Zdrojový uzel předpočítá mezilehlé uzly.
- Přesné trasy mezi nimi jsou distribuovaně rozhodnuty směrovači.

Existují 3 úrovně adaptivity: žádná, pseudo, plná.

Deterministické směrovací algoritmy: Vždy generují tutéž jedinou trasu pro danou dvojici zdrojové a cílové adresy, tzn. R je funkce. Příklady: XY, XYZ, e-cube.

Datově necitlivé směrovací algoritmy: výběrová funkce ρ je necitlivá ke stavu sítě.

- Jakékoli deterministické směrování je datově necitlivé.
- Datově necitlivé směrování není nutně deterministické. Výběrová funkce ρ může z volných cest/kanálů vybrat náhodně nebo cyklicky.

Adaptivní směrovací algoritmy: Výběrová funkce ρ vybírá směr z lokálního hlediska co nejméně přetížený s cílem vyhnout se zahlceným nebo porouchaným částem sítě. Funkce ρ používá informace o stavu kanálů (délky front čekajících zpráv, počty odešlých zpráv za posledních τ komun. kol, sondování v sousedství ap.).

- Typická kombinace je distribuované adaptivní směrování.
- Zdrojové adaptivní směrování lze použít, pouze nemění-li se komunikační stav sítě příliš rychle.

Minimální algoritmy:

- Alternativní názvy: lačné, přímé, po nejkratší trase, nebo přírůstkové.
- Každé směrovací rozhodnutí přivádí paket blíže k cíli.
- Náchylné k statickému zablokování ($deadlock$).
- Deterministické a datově necitlivé směrovací algoritmy jsou obvykle minimální.
- Minimální adaptivní směrování: výběr z více nejkratších cest využívající stavu sítě.

Neminimální algoritmy:

- Alternativní názvy: nelačné, nepřímé, obcházecí, nebo nepřírůstkové.
- Pakety mohou být poslány dále od svých cílů.
- Datově necitlivé nebo plně adaptivní.
- Netrpí zablokováním, ale náchylné k dynamickému zablokování ($livelock$).

Progresivní směrování:

- Každé směrovací rozhodnutí alokuje nový kanál a délka trasy roste.
- Paket tím není nutně přiváděn blíže k cíli.
- Při zablokované trase paket
  - buď čeká
  - nebo je odkloněn (náhodně nebo adaptivně).
- Minimální směrování je vždy progresivní.

Směrování s návratem:

- Při zablokované trase se paket stáhne zpět a uvolní předtím rezervované kanály (část nebo všechny).
- Komplikované protokoly (zpětné signály vysílači).
- Důležitá je volba algoritmu pro výpočet prodlevy znovuvyslání neúspěšného paketu.
- Vždy adaptivní.

Rozhodování o směrování by mělo být rychlé. V případě distribuovaného směrování by se mělo provádět v HW.

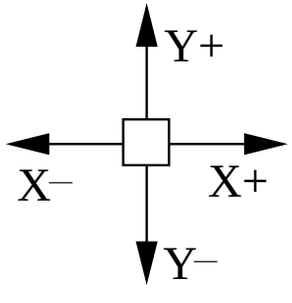
Konečný automat: HW nebo SW algoritmus implementující nějaký konečný automat.

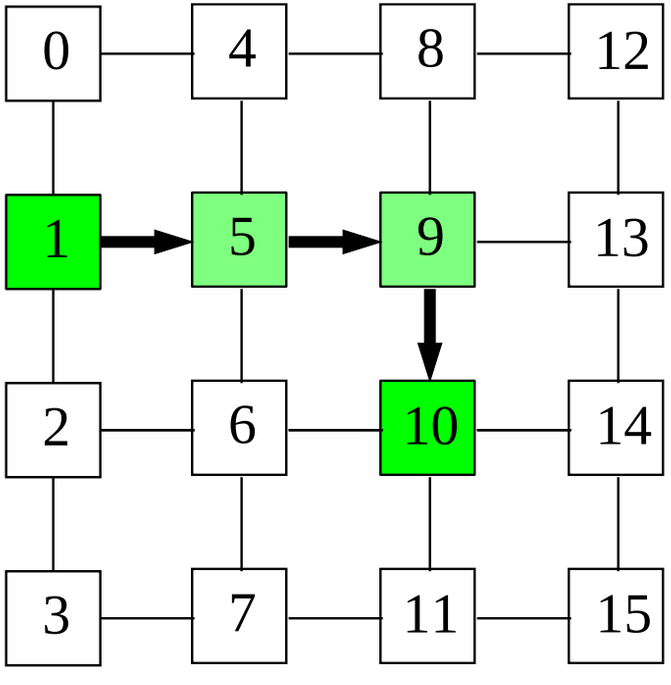
Směrovací tabulky: N položek

- Zdrojové směrování: 1 položka = specifikace celé trasy.
- Distribuované směrování: 1 položka = číslo výstupního kanálu.
- Statické nebo dynamicky udržované (Myrinet).
- Nevýhoda: velké paměťové nároky a závislost velikosti tabulek na velikosti sítě.
  - Jedno možné řešení = intervalové směrování.

Intervalové směrování (IR)

- Tabulka = pouze 1 položka/1 výstupní kanál.
- Položka = interval cílových adres.
- Nalezení optimálních schémat pro IR je obecně obtížné.
- **Problém:** Je-li dána topologie G, $\exists$ 1-intervalové minimální směrování?
- Např.: ANO pro 2-D mřížky a XY směrování.





uzel	kanál	interval
1	X+	$4-15$
	Y+	$0-0$
	Y–	$2-3$
5	X+	$8-15$
	X–	$0-3$
	Y+	$4-4$
	Y–	$6-7$
9	X+	$12-15$
	X–	$0-7$
	Y+	$8-8$
	Y–	$10-11$

Optimální 1-intervalové XY směrování v 4×4 mřížce

Metrika složitosti komunikačních operací - propustnosti

Šířka kanálu $w =$ velikost fitu $=$
$=$ počet bitů, které může fyzický kanál přenést najednou mezi 2 sousedními směrovači.
Předpokládáme, že $w = 1$ byte [B]).

Rychlost kanálu $q =$ špičková rychlost přenosu bitů po 1 fyzickém vodiči (v [B/s]).

Propustnost kanálu $B = q$ (ve [B/s]).

Bisekční propustnost sítě G: $B_{\mathrm{B}}(G) = B \times \mathrm{bw}_e(G)$ (v [B/s]).

Síťová propustnost sítě G: $B_{\mathrm{T}}(G) = B \times |E(G)|$ (v [B/s]).

Metrika složitosti komunikačních operací - latence

Zpoždění kanálu $t_m = 1/q$ = zpoždění mezi sousedními směrovači na 1 fit (v [s/B]).

Startovní zpoždění t_s = SW a HW zpoždění v zdrojovém a cílovém uzlu nutné pro

- zformátování a složení paketu,
- validace dat a jejich kopírování mezi pamětí uzlu a frontou směrovače.

Směrovací zpoždění t_r = čas pro směrovacího rozhodnutí během budování trasy (v [s]).

Přepínací zpoždění t_w = čas přenosu přes přepínač ze vst. na výst. kanály (v [s/B]).

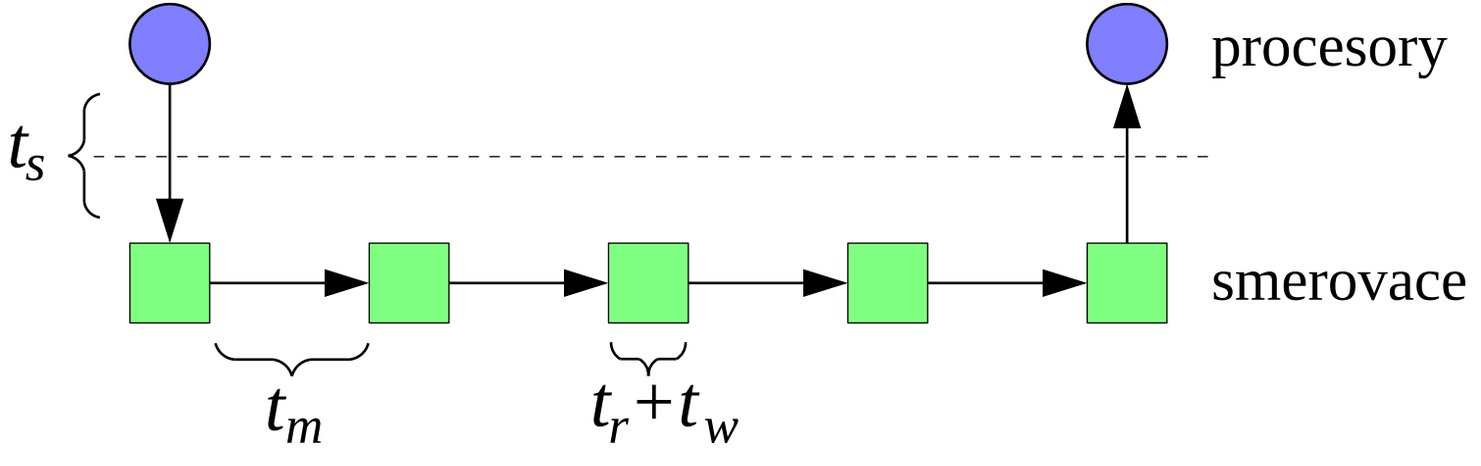
Základní síťové zpoždění = čas bezkonfliktního přenosu paketu sítí bez uvažování startovního zpoždění (= čas od vstupu hlavičky do zdrojového směrovače do výstupu konce paketu z cílového směrovače).

Základní komunikační zpoždění = základní síťové zpoždění + startovní zpoždění.

Celkové síťové zpoždění = základní síťové zpoždění + doba blokování.

Předpoklady modelu časové složitosti bezkonfliktní komunikace

1. $\mu =$ velikost paketu (v [B]).
2. $\delta =$ délka přenosové trasy.
3. Platí $t_s \gg t_m \approx t_w \approx t_r$.
4. Směrovače mají vstupní i výstupní fronty.
5. Doba přenosu μ-bytového paketu mezi 2 sousedními směrovači je μt_m (v [s]).



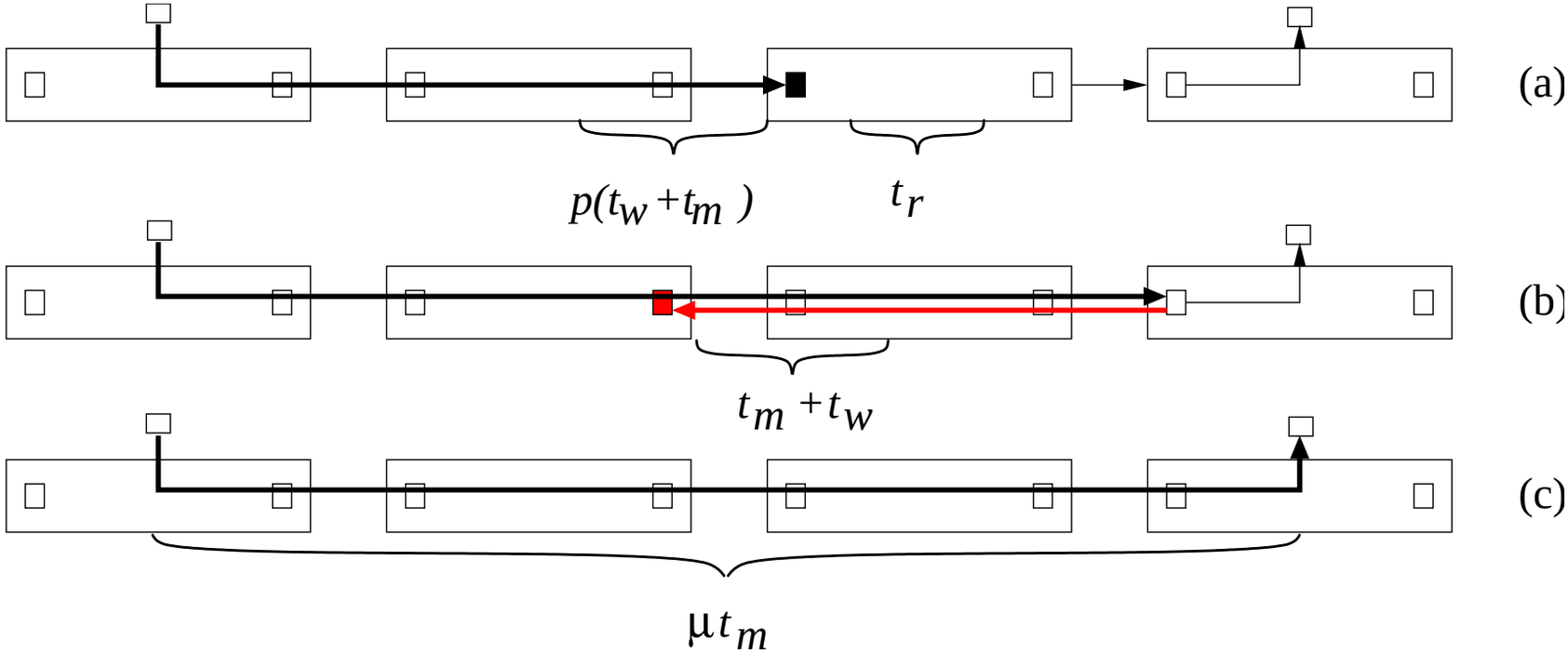
Přepínání kanálů (*Circuit Switching*, CS)

Zkonstruování fyzického obvodu: rezervován před vlastním přenosem dat.

- Budován směrovací sondou s cílovou adresou a dalšími řídícími informacemi.
- Sonda je dlouhá $p > 1$ flitů.
- Sonda postupuje přes mezilehlé směrovače a rezervuje fyzické linky.
- Trasa je nastavena poté, co sonda dorazí do cíle.
- Pak je poslán nazpět potvrzovací flit.

Přenos zprávy:

- Začíná po obdržení potvrzení.
- Celá zpráva je přenášena plnou přenosovou rychlostí vytvořené trasy.
- Trasa = HW obvod, který je rezervován po celý čas přenosu dat.
- Sonda je ukládána v každém směrovači, ale datové bity nikoli!!!
  $\Longrightarrow$ obvod funguje jako jediný vodič.
- Neexistují žádná omezení na délku zprávy (= souvislý řetězec bitů).
- Výhodné, jsou-li zprávy dlouhé a ne příliš časté.
- Uvolnění obvodu: např. posledními datovými bity nebo cílovým uzlem.



Základní komunikační zpoždění

Přenos zprávy délky μ na vzdálenost δ trvá čas

$$t_{\mathrm{CS}}(\mu, \delta) = t_s + \delta(t_r + (p+1)(t_w + t_m)) + \mu t_m.$$

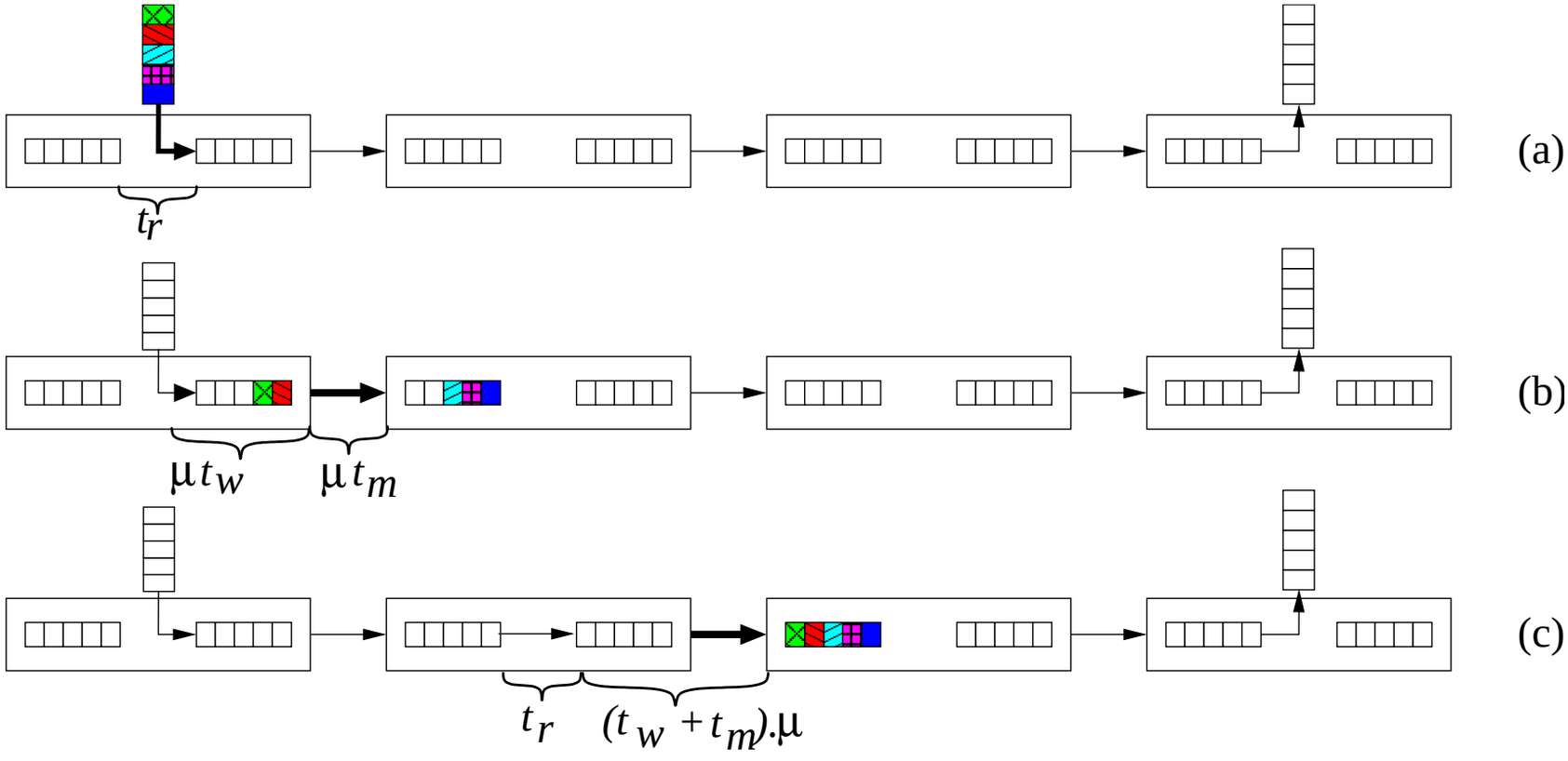
1. Sonda potřebuje čas $\delta(t_r + p(t_w + t_m))$, aby se dostala do cíle.
2. Potvrzení putuje zpět čas $\delta(t_w + t_m)$.
3. Přenos dat trvá čas μt_m.

Přepínání ulož-pošli-dál (*Store-and-Forward*, SF)

- Zprávy jsou rozděleny do paketů pevné délky.
- Pakety jsou rozloženy do flitů, počínaje hlavičkovým flitem.
- Směrovače mají vstupní a výstupní fronty pro celé pakety.
- Každý paket je individuálně směrován ze zdroje do cíle.
- Jeden krok = hop = zkopírování celého paketu z výstupní fronty do dalšího vstupní fronty.
- Směrovací rozhodnutí jsou činěna směrovačem pouze
  po té, co byl celý paket uložen ve vstupní frontě.

Poznámky

- SF se také označuje přepínání paketů.
- Výhodné pro krátké a časté zprávy (z celé trasy je obsazen nejvýše 1 kanál).
- Velké fronty: dražší a pomalejší směrovače, nebo omezená velikost paketů.
- Komunikační zpoždění je úměrné součinu velikosti paketu a délky trasy.
  $\Longrightarrow$ tlak na minimální směrování a nízký průměr sítě.



Základní komunikační zpoždění

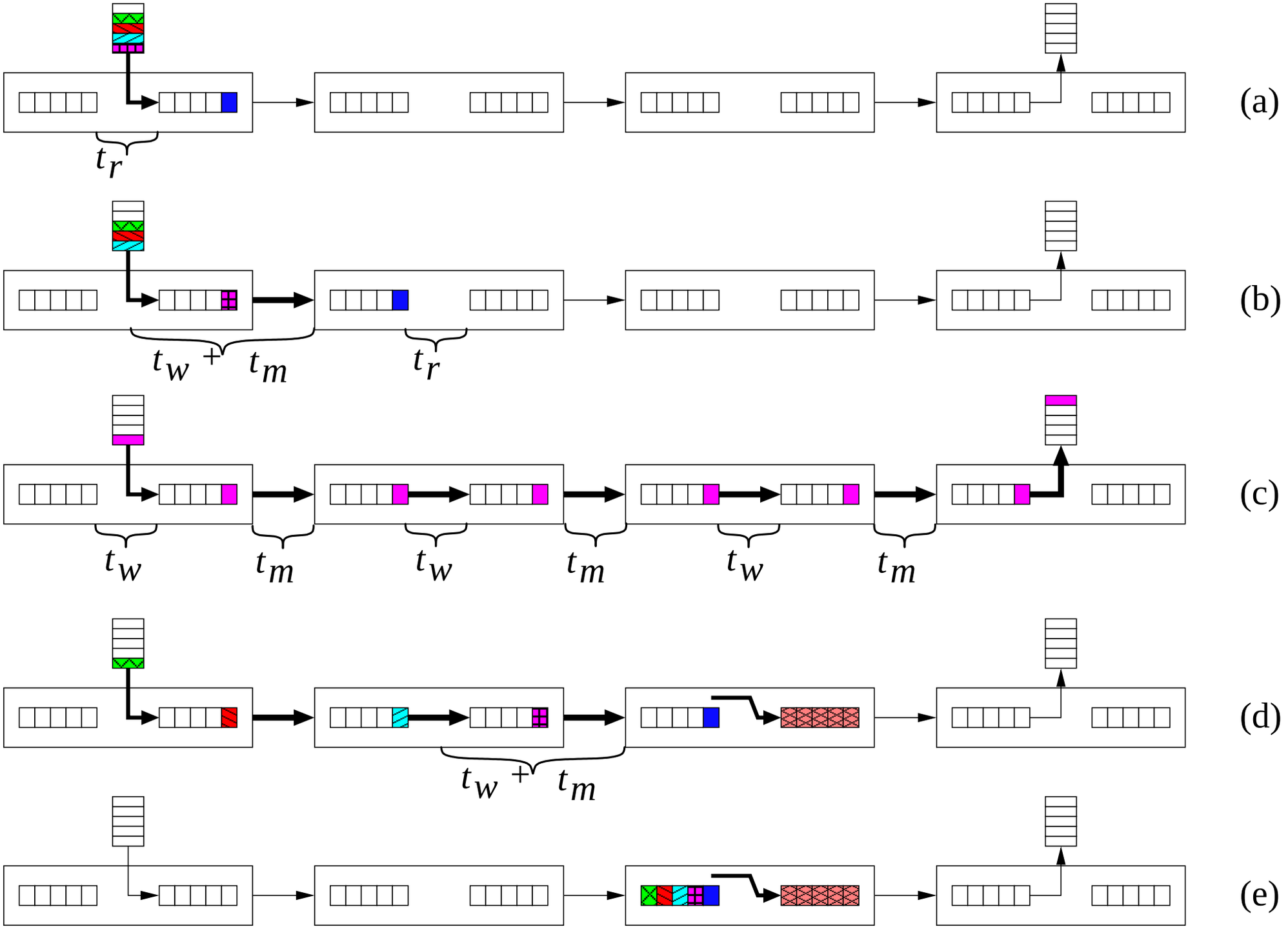
Přenos paketu délky μ na vzdálenost δ trvá

$$t_{\mathrm{SF}}(\mu, \delta) = t_s + \delta(t_r + (t_w + t_m)\mu).$$

Každý směrovač musí nejprve učinit směrovací rozhodnutí a pak teprve celý paket přeskočí do dalšího směrovače, což trvá $t_r + (t_w + t_m)\mu$, a tento postup se opakuje δ krát.

Průřezové přepínání (*Virtual Cut-Through*, VCT)

- Zprávy jsou rozloženy do paketů.
- Směrovače mají fronty pro celé pakety (jako při SF přepínání).
- Přišedší hlavičkový flit nečeká na uložení celého paketu, ale prořízne do dalšího směrovače, jakmile bylo učiněno směrovací rozhodnutí a výstupní kanál je volný.
- Každý další flit je uložen, ale také se okamžitě prořízne do dalšího směrovače, je-li výstupní kanál volný.
- Bezkolizní paket má podobu volného řetězce flitů vedoucího skrz mezilehlé směrovače.
- Všechny fronty podél trasy jsou blokovány pro jiné komunikační požadavky.
- Pokud hlavička nemůže pokračovat, následující flity se postupně dotahují a kanály, které dosud obsazovaly, se postupně uvolňují.


(a)
t_r
(b)
$t_w + t_m$
t_r
(c)
t_w
t_m
t_w
t_m
t_w
t_m
(d)
$t_w + t_m$
(e)

Základní komunikační zpoždění

Paket délky μ je přenesen na vzdálenost δ v čase

$$t_{\mathrm{VCT}}(\mu, \delta) = t_s + \delta(t_r + t_w + t_m) + \mu \max(t_w, t_m).$$

1. $\delta(t_r + t_w + t_m)$: zpoždění hlavičky při provádění směrovacích rozhodnutí, přepínaní a přesunech mezi směrovači.
2. $\max(t_w, t_m)$: Rychlost přenosu řetězce flitů, jakmile hlavička dosáhne cíle, předpokládáme-li, že směrovače mají jak vstupní tak výstupní fronty.
   - V případě pouze vstupních front nebo pouze výstupních front bychom měli $t_w + t_m$ místo $\max(t_w, t_m)$.

Poznámky

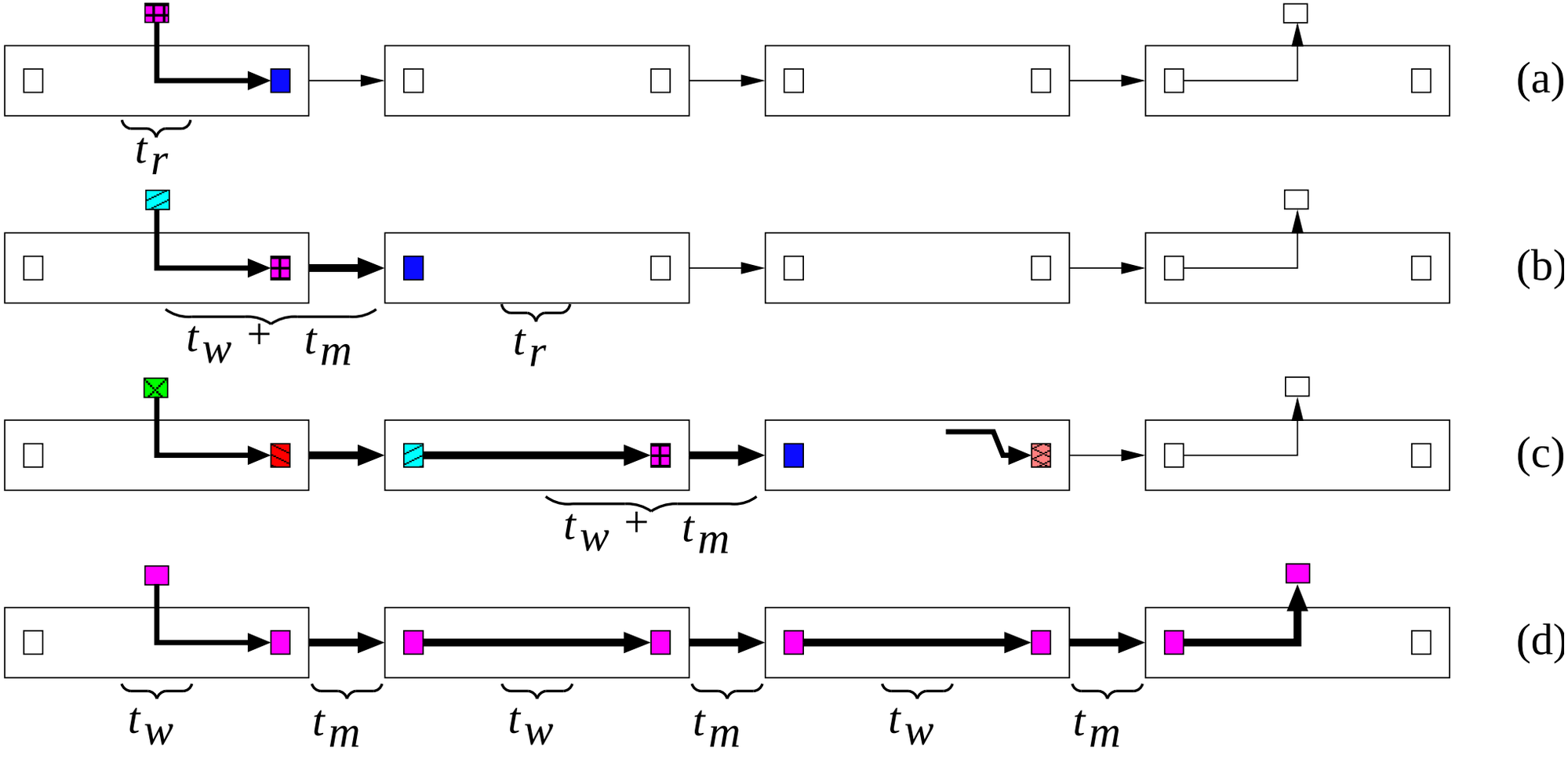
- Pouze hlavičkový flit obsahuje směrovací informace
  $\Longrightarrow$ každý příchozí datový flit následuje svého předchůdce do téhož výstupního kanálu
  $\Longrightarrow$ flity různých paketů nelze prokládat nebo multiplexovat na stejném fyzickém kanálu.
- Toto je nejsložitější a nejnákladnější technika mezi 4 zde uvedenými, ale díky vyspělosti dnešních VLSI technologií je dnes nejvíce používána.

Červí přepínání (*Wormhole*, WH)

- Pakety jsou rozloženy do flitů a jako had se posouvají podél vytčené trasy přesně jako u bezkonfliktního VCT přepínání.

$$t_{\mathrm{WH}}(\mu, \delta) = t_s + \delta(t_r + t_w + t_m) + \mu \max(t_w, t_m)$$

- Směrovače nemají fronty pro celé pakety, ale pouze malé fronty pro 1 nebo několik flitů.
- Hlavní nedostatek: Nemůže-li hlavička pokračovat dále, protože výstupní kanál je obsazen, celý řetěz flitů zamrzne, blokujíce fronty a linky ve směrovačích podél trasy.
- WH přepínání je náchylné k zablokování.
- WH přepínání umožňuje malé, levné a rychlé směrovače
  $\Longrightarrow$ v 90. letech nejběžnější přepínací technika v komerčních počítačích.


(a)
t_r
(b)
$t_w + t_m$
t_r
(c)
$t_w + t_m$
(d)
t_w
t_m
t_w
t_m
t_w
t_m

Zjednodušené výrazy pro komunikační zpoždění

Typické paralelní architektury:

- $t_s \gg t_m$
- $t_m \approx t_w \approx t_r$

Zjednodušené výrazy:

$$t_{\mathrm{SF}}(\mu, \delta) = t_s + \delta(t_r + (t_w + t_m)\mu) \doteq t_s + \delta\mu t_m$$

$\Longrightarrow$ **SF** je citlivé na vzdálenost.

$$t_{\mathrm{CS}}(\mu, \delta) \doteq t_{\mathrm{VCT}}(\mu, \delta) = t_{\mathrm{WH}}(\mu, \delta) \doteq t_s + \delta t_d + \mu t_m, \quad \text{kde} \quad t_d = t_r + t_w + t_m$$

$\Longrightarrow$ CS, VCT a **WH** jsou necitlivé na vzdálenost.

Pro velká μ (řádově stovky flitů) je $\mu t_m \gg \delta t_d$ ve většině paralelních strojů.

Proč je WH směrování důležité?

- Přepínání WH je jednoduché, levné a necitlivé na vzdálenost.
- WH 2-D toroid může výkonově překonat WH hyperkrychli zhruba téže ceny.

Hypotéza: cena sítě $\approx$ počet vodičů mezi směrovači

	$Q_{\log p}$	$K(\sqrt{p}, \sqrt{p})$
# kanálů na 1 směrovač	$\log p$	4
stejná cena	sériový 1-vodičový kanál	$\log p/4$ vodičů na 1 kanál
průměrná vzdálenost	$\log p/2$	$\sqrt{p}/2$
průměrné síťové zpoždění	$\frac{\log p}{2} t_d + \mu t_m$	$\frac{\sqrt{p}}{2} t_d + \frac{4}{\log p} \mu t_m$

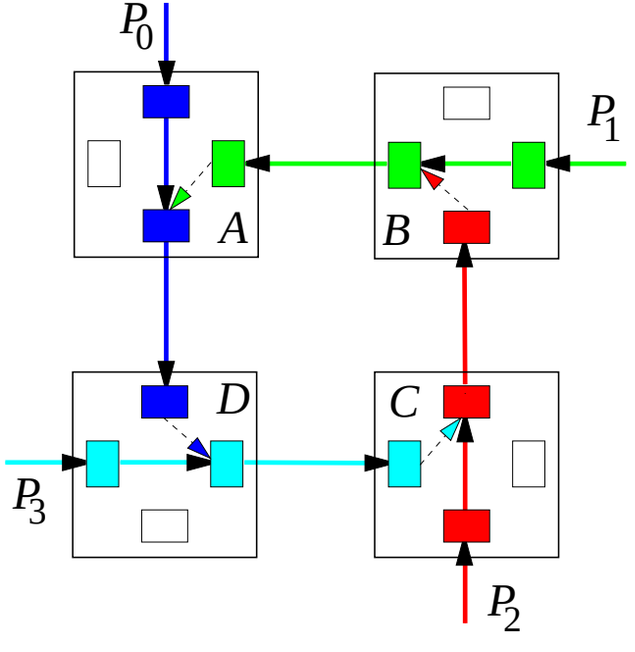
$\Longrightarrow$

Pro velká μ, první podvýrazy lze vynechat a pro $p \geq 16$, WH toroid je pro bezkonfliktní komunikaci v průměrném případě rychlejší než stejně velká WH hyperkrychle téže ceny.

Zablokování

- Obecně: skupina agentů (paketů) nemůže 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.
- Katastrofické důsledky pro komunikační síť: na zablokovanou část se nabalují další čekající agenti (efekt sněhové koule).
- Příklad zablokování 4 paketů $P_0, \ldots, P_3$ ve WH síti:

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



Zablokovaná konfigurace

- SF,VCT s pouze výstupními frontami, deterministické směrování:

  1. V síti existuje množina paketů K, z nichž žádný zatím nedorazil do cíle.
  2. Žádný paket z K nemůže pokračovat, protože fronta kanálu, nabídnutého směrovací funkcí, je obsazena jiným paketem z K.

- WH, deterministické směrování:

  1. V síti existuje množina paketů K a žádný hlavičkový flit zatím nedorazil do cíle.
  2. Žádný hlavičkový flit z K nemůže pokračovat, protože fronta kanálu nabídnutého směrovací funkcí, je obsazena flitem jiného paketu z K.
  3. Za každým čekajícím hlavičkovým flitem následují čekající datové flity.

Řešení problému zablokování

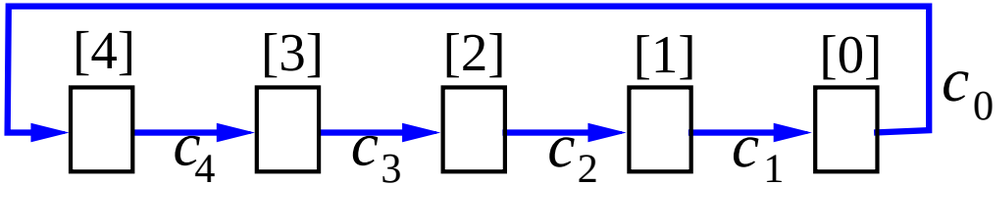
- Detekce a zotavení: nejméně opatrné, možný velký zisk, ale i ztráty.
- Prevence: konzervativní přidělování všech prostředků najednou
  $\Longrightarrow$ jejich malé využití – použito v technice přepínání kanálů (CS).
- Vyhnutí se zablokování: postupné přidělování prostředků tak, aby globálně nemohlo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

Graf kanálových závislostí

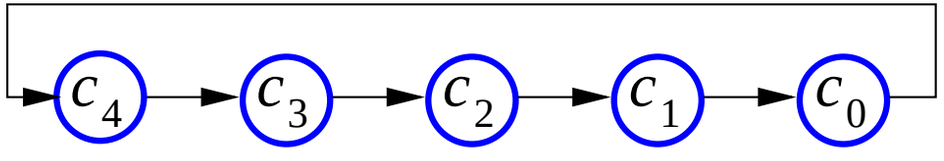
Definice 1. *Deterministická směrovací funkce R na grafu G: pro každý vstupní uzel $u \in V(G)$, pro každý jeho vstupní kanál c_1 a pro každý cílový uzel d, směrovací funkce R určí výstupní kanál $c_2 = R(u, c_1, d)$.* ♣

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 $R(u, c_1, d) = c_2$.* ♣



K(5)



Z(K(5),R)

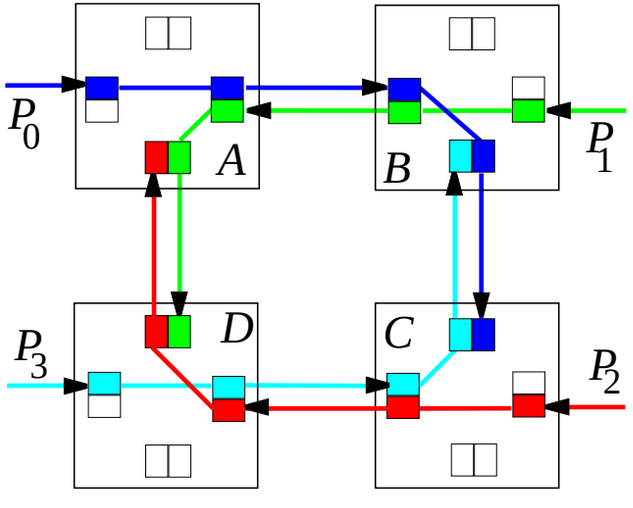
Věta 3. *Deterministická směrovací funkce R na grafu G nemůže vést k zablokování $\iff$ $Z(G, R)$ je acyklický.* ♣

Vyhnutí se zablokování I

- 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



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

Lemma 4. *Graf orientovaný algoritmem* CORRECTORIENT *je korektní.*

Důkaz. Každý vnitřní uzel $T(r)$ má nejméně 1 hranu orientovanou směrem ke kořenu a pouze kořen 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. ♣

Definice 5. *Uvažujme souvislý graf G a jeho korektní digraf G'. Pak legální orientované trasy pro směrovací fci R' $Nahoru^*/Dolu^*$ v G' jsou pouze takové, které se sestávají*

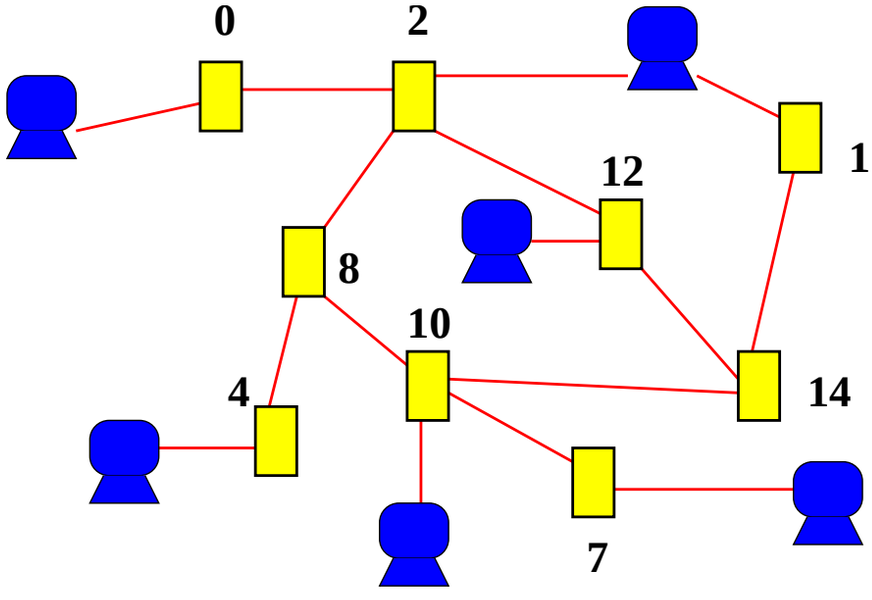
- *z 0 nebo více hran ve směru orientace G' (potenciální část $Nahoru$),*
- *následovaných 0 nebo více v opačném směru orientace G' (potenciální část $Dolu$).* ♣

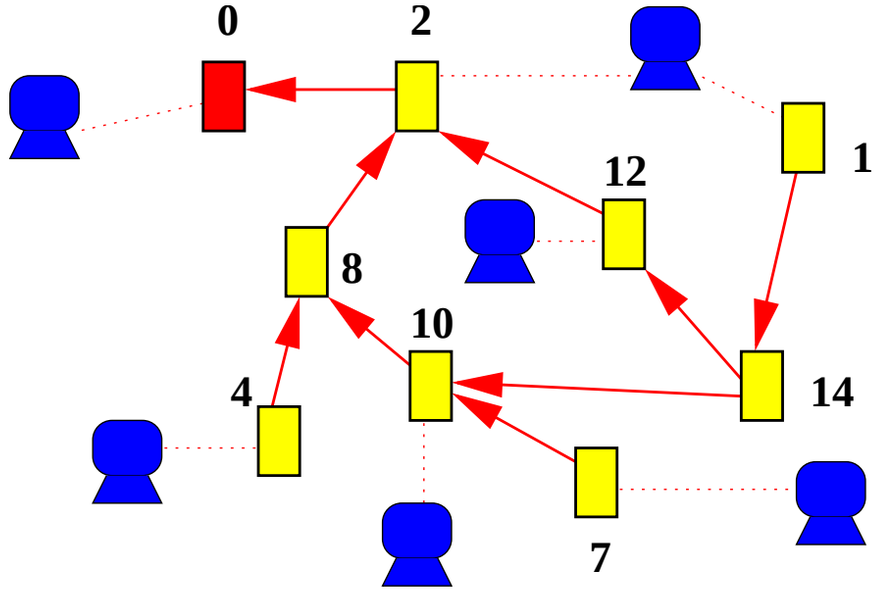
Lemma 6. *G' je silně souvislý vzhledem k legálním trasám konstruovaným fcí R' a $CDG(G', R')$ je acyklický.*

Důkaz. Sporem. Protože $T(r)$ kostra do šířky, každý uzel je dostupný z každého uzlu legální trasou přes kořen r. Zřetězením několika legálních tras nemůže vzniknout cyklická posloupnost požadavků na kanály, neboť by musela existovat trasa, ve které je nějaká hrana z části $Dolu$ následovaná nějakou hranou z části $Nahoru$. ♣

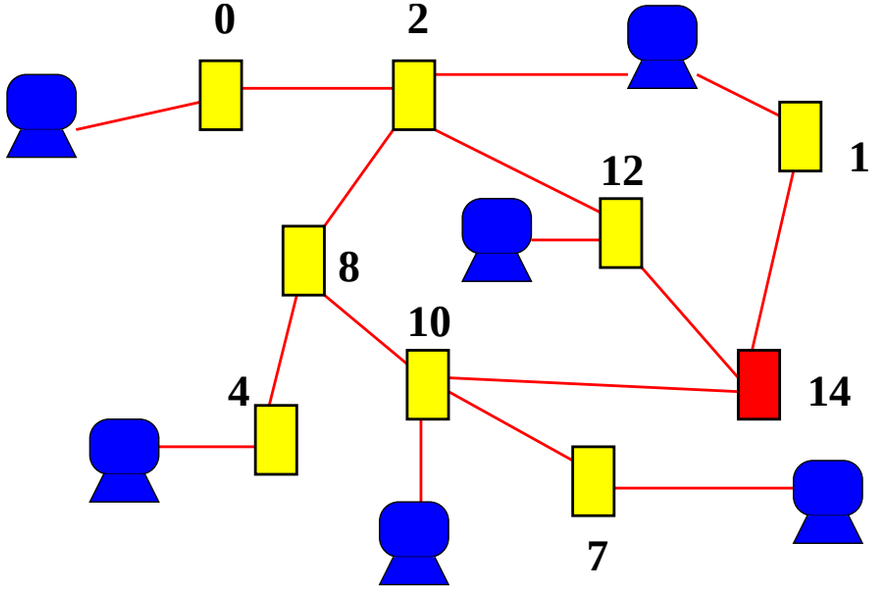
Korektní orientované grafy v svazcích počítačů

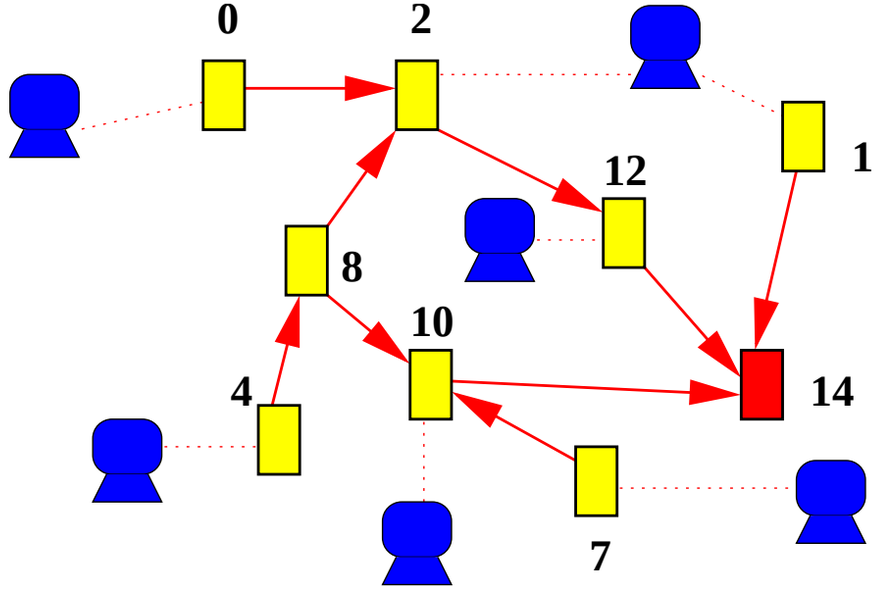
(Autonet. MDST alg.: Kořen = uzel s nejmenším ID.





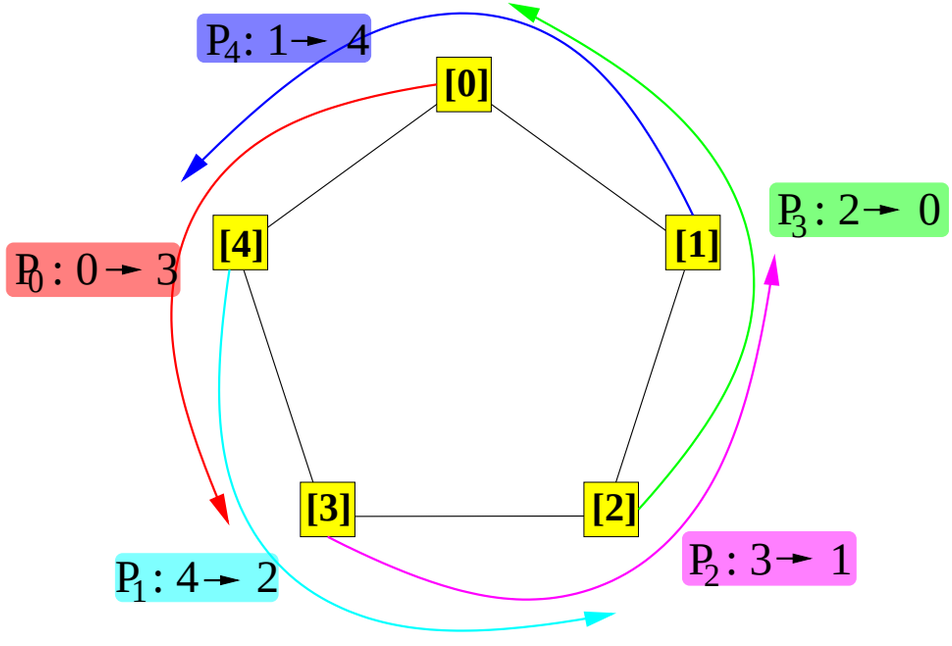
Myrinet: POST alg.: Kořen = uzel, který detekoval změnu v konfiguraci



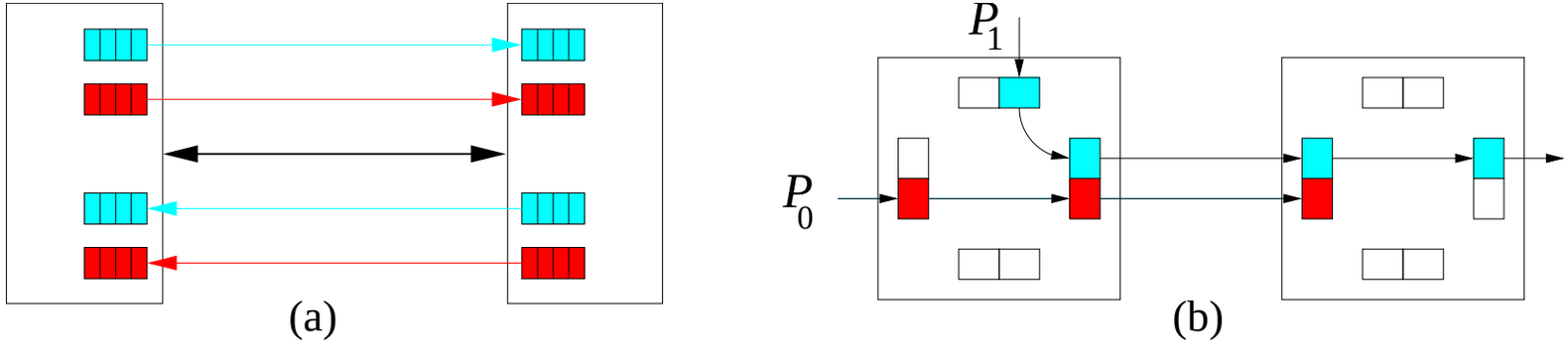


Zablokování v toroidech a virtuální kanály

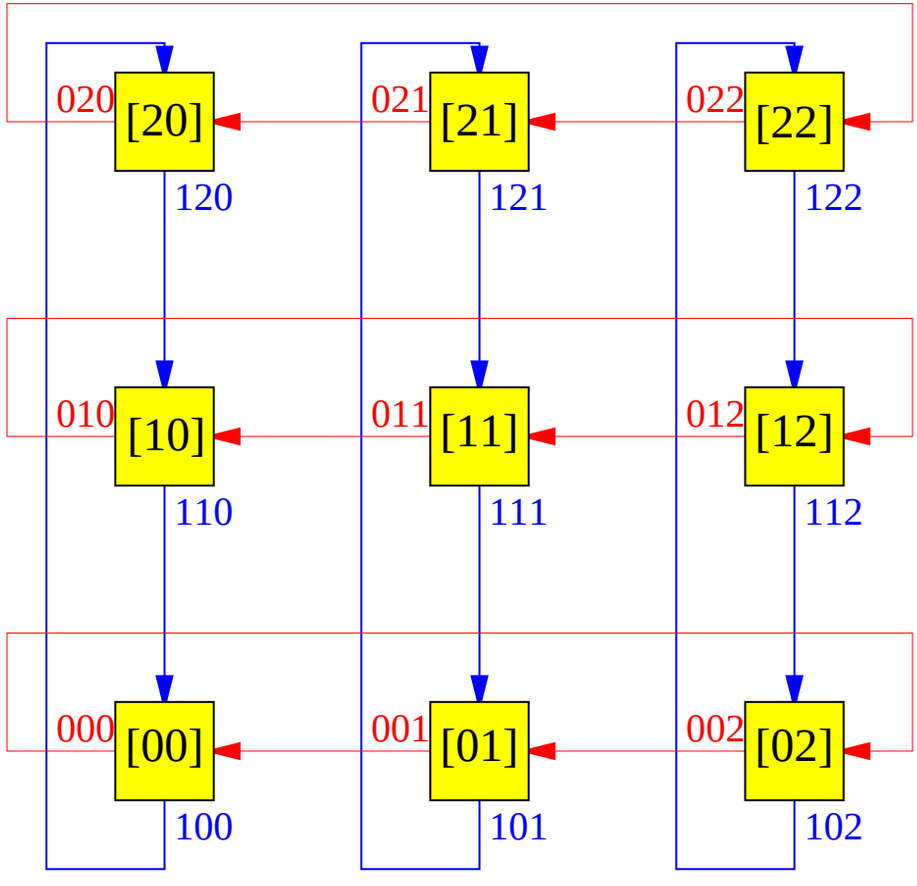
- Restrikce směrovací fce R na dimenzionálně uspořádanou R' v toroidech nefunguje, viz příklad permutace tornádo čili cyklický posun.



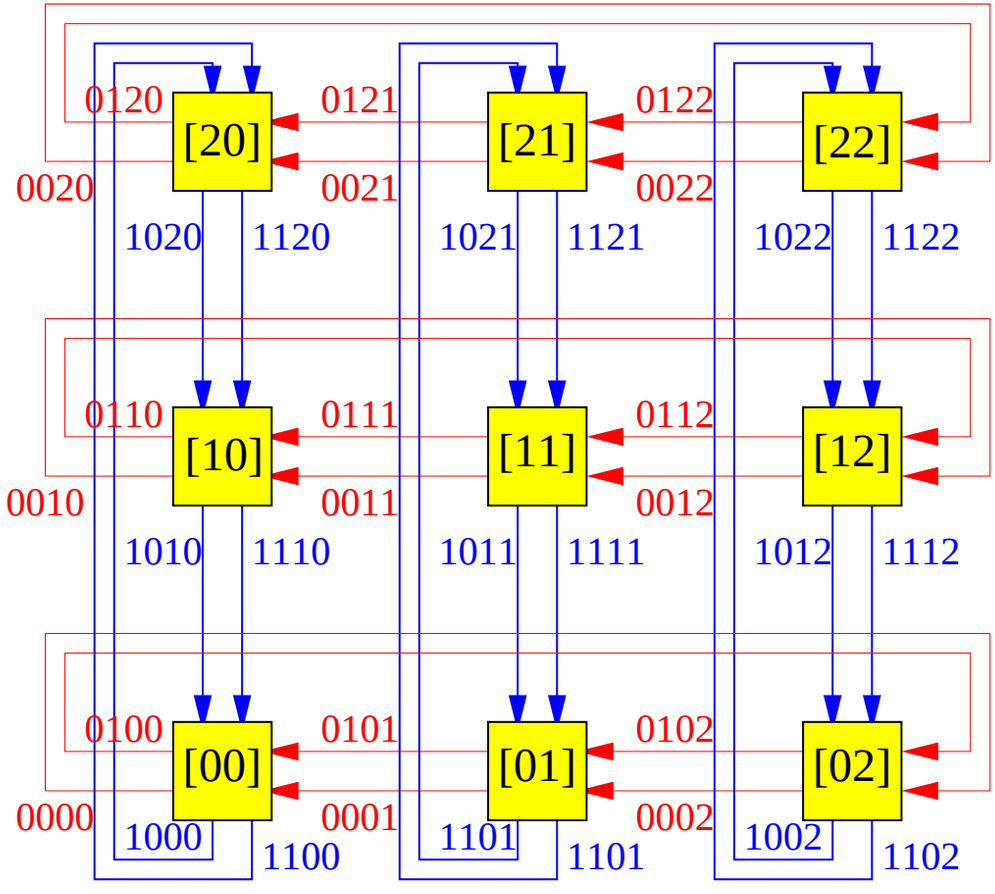
- Každý 1-D toroid má cyklický graf kanálových závislostí Z.
- Řešení: dva virtuální kanály na 1 fyzický.



- Uvažujme $K = K(z_1, \ldots, z_n)$ s minim. směrováním a s plně duplex. kanály.
- Každý $u = [a_1, \ldots, a_i, \ldots, a_n]$ má v dimenzi i
  - 1 decr kanál jdoucí do uzlu $[a_1, \ldots, a_i \ominus_{z_i} 1, \ldots, a_n]$
  - a 1 incr kanál jdoucí do uzlu $[a_1, \ldots, a_i \oplus_{z_i} 1, \ldots, a_n]$.
- Uzel u má n fyzických *decr* kanálů c_{iu} označ. $(n+1)$-znak. řetězcem iu, kde $i = 0, \ldots, n-1$ je číslo dimenze.
- Každý c_{iu} je rozdělen na 2 virtuální kanály: horní c_{i1u} a dolní c_{i0u}.
- Definujeme lexikografické uspořádání *decr* virtuálních kanálů.
- Totéž definujeme pro *incr* kanály.



(a)



(b)

Značení (a) fyzických a (b) virtuálních *decr* kanálů v 2-D toroidu $K(3,3)$, kde dim. $X = 0$ a $Y = 1$. Např. $c_{0120} > c_{0022} > c_{0021}$

Neblokující směrování v toroidech

Definice 7. R'_{d} = *směrovací fce, která používá virtuální* decr *kanály v striktně klesajícím pořadí a virtuální* incr *kanály v striktně rostoucím pořadí.*

Legální *decr* trasy v $K(z)$:

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