BPC-HWS

Vypracované otázky k SZZ

Text: ChezZ, Revival, mReasyLife
Korektura: ChezZ, Revival, mReasyLife
Formátování: –

7. března 2022

Obsah

1 Dílčí kroky procesu směrování uvnitř směrovače.

- Zpracování příchozích paketů
  - 1. krok
    - Príjem rámca (napr. Ethernet)
    - Na vstupnej časti sieťového rozhrania na úrovni linkovej vrstvy
      - interpretace položek v hlavičce protokolu linkové vrstvy
      - detekce začátku a konce rámce
      - nakonec - zpracována zapouzdřená datová jednotka (identifikace začátku paketu, identifikace hlavičky paketu)
    - Spracovanie na urovni linkovej vrstvy
      - Vyjmutí hlavičky rámce
      - Vytvorenie kontextu paketu (datová struktura využívaná pro řízení dalších operací ve směrovači)
      - Vyplnění příslušných polí kontextu paketu
      - Vyjmutí zapouzdřeného paketu a jeho zpracování
    - Zpracování informací síťové vrstvy (L3)
      - Nalezení hlavičky IP
      - Ověření konzistence hlavičky IP
      - Přečtení obsahu IP hlavičky
      - Zápis do kontextu paketu
  - 2. krok
    - Předání paketu i kontextu do bloku přepojování
  - 3. krok
    - Dočasné uložení paketu ve vyrovnávací paměti
    - Vyhledávání v přepojovací tabulce
    - Na základě informací v kontextu paketu
    - Nalezení správného výstupního rozhraní
    - Zápis do kontextu paketu
  - 4. krok
    - Předání paketu i kontextu základní desce
    - Plánování přenosu paketu
    - Zohlednění priority paketu
  - 5. krok
    - Přenos paketu k odchozímu síťovému rozhraní
    - Postup zpracování na výstupu
- Zpracování odchozích paketů
  - 6. krok
    - Předání paketu i kontextu do výstupní linkové karty
    - Uložení paketu ve vyrovnávací paměti

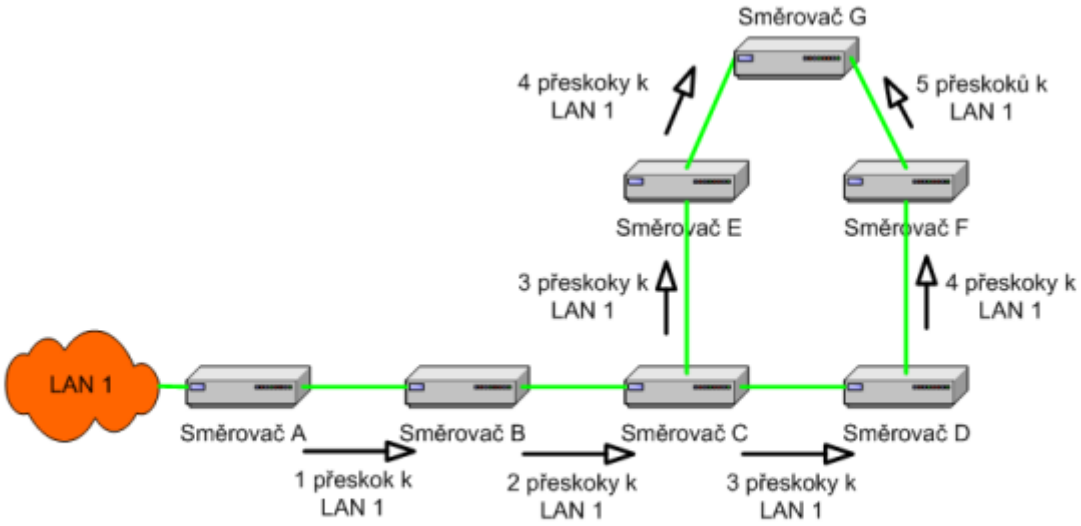
    - Aktualizace adresy v kontextu paketu
- 7. krok
    - Předání kontextu Správě front
    - Kontrola priority paketu (Určené blokem přepojování)
    - Vložení kontextu paketu do odpovídající fronty
    - Naplánování odesílání paketu (Zpravidla podle propustnosti přidělené dané frontě)
    - Zaplnění fronty (Způsobeno zahlcením sítě,riesene Řízenym zahazováním paketů)
- 8. krok
    - Předání paketu i kontextu správě provozu
    - Identifikace uživatele z kontextu
    - Kontrola případného omezení rychlosti pro uživatele
    - Tvarování provozu (Pozdržení paketu, Zahození paketu)
- 9. krok
    - Předání paketu odchozímu síťovému rozhraní
    - Aktualizace informací síťové vrstvy (L3) (Nová hodnota TTL, Výpočet zabezpečení CRC)
    - Vygenerování nové hlavičky linkové vrstvy (L2)
- 10. krok
    - Posledním krokem zpracování paketu je jeho fyzické odeslán

2 Směrovací protokol Distance-Vector, směrovací protokol RIP

2.1 Distance-Vector

Tento typ směrovacích protokolů je založen na distribuovaném výpočtu, v rámci kterého každý směrovač spočítá svoji „nejlepší“ cestu k dostupným sítím. Směrovače provádí výpočet nezávisle na ostatních. Po skončení výpočtu směrovač informuje svoje sousedy o své „nejlepší“ cestě. Na základě těchto informací pak sousední směrovače mohou přehodnotit své znalosti o „nejlepších“ cestách. Když sousední směrovač zjistí, že našel „lepší“ cestu, než ta původní, tak o tom také informuje svoje sousedy včetně směrovače, od kterého přišla zpráva inicializující výpočet. Z popisu je vidět, že stanovení „nejlepší“ cesty je iteračním procesem, který probíhá v několika krocích. Mezi klíčové požadavky na tento iterační proces patří:

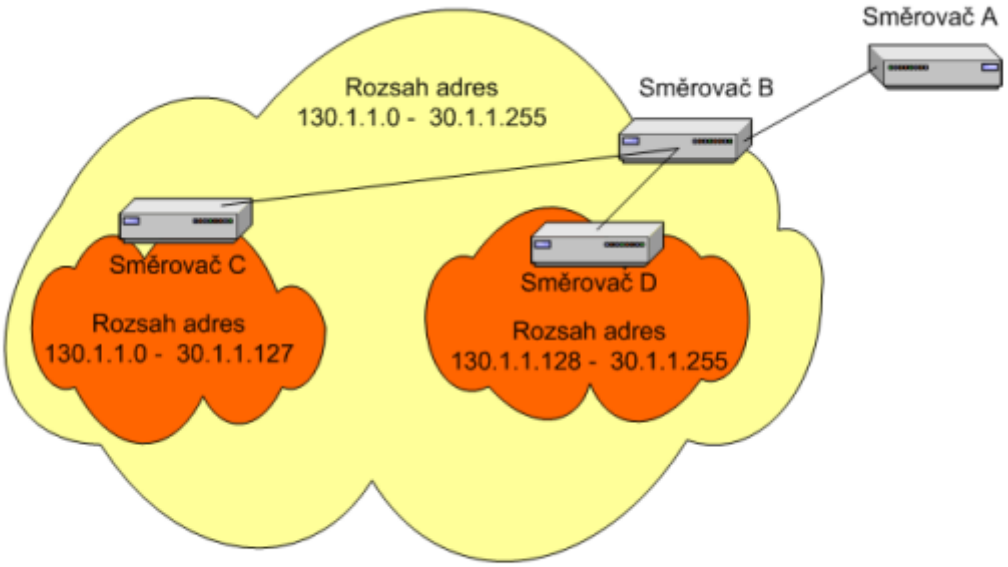
- Stabilita, tj. aby se proces po konečné době ustálil a nedostal do stavu, kdy si směrovače mezi sebou vzájemně cyklicky vyměňují směrovací zprávy.
- Rychlá konvergence, aby doba potřebná k ustálení byla co nejkratší.



Obr. 2.5 Šíření směrovacích informací u směrovacích protokolů distance-vector

„Nejlepší“ cestou v případě směrovacích protokolů distance-vector je považována cesta, která obsahuje nejmenší počet přeskoků. Výhodou směrovacích protokolů distance-vector je jejich jednoduchá konfigurace, údržba i řešení případných problémů. Nevýhodou tohoto typu směrovacích protokolů je, že počet přeskoků je často omezen na poměrně malou hodnotu (např. 15). Když cílová síť je ve větší vzdálenosti, tak je považována za nedostupnou. Dále změna v topologii sítě může způsobit záplavové šíření směrovacích informací. Částečnou ochranu proti záplavovému šíření směrovacích informací může zajistit omezení rozhraní, přes která směrovač může rozeslat směrovací informace či omezení, přes která

rozhraní mohou být jaké sítě avizovány. Tato pravidla však musí být nakonfigurována ručně. Podporuju agregaciu adries pre lepsiu priehladnost zaznamov v smerovacich tabulbach (spaja suvisle adresne priestory niekolkych podsieti do jedneho velkeho adresne priestoru ktory dalej siri).



Obr. 2.6 Agregace adres

2.2 RIP

Konkrétním představitelem směrovacích protokolů distance-vector je Routing Information Protocol (RIP), který je určen především pro využití v lokálních sítích. Iterační proces protokolu je založen na algoritmu označeném jako Ford – Fulkerson či Bellman – Ford. RIP šíří informace o sítích, které zná a dále tyto informace doplňuje o svoji vzdálenost od těchto sítí. Maximální vzdálenost je 16 přeskoků. Vzdálenější síť je označena jako nedostupná. Vzdálenost přímo připojené sítě je 1 přeskok. Směrovač rozesílá směrovací informace po 30 s. Ak neobdrzi spravu z daneho smeru:

- Po 180 s -> oznaci ako nepouzitelnu
- Po 240 s -> odstrani zo smerovacej tabulky vsetky info vztahujuce sa k sietam ktore boli dostupne danym smerom

RIP v1 nepodporovala masku siete -> neumoznovala delenie adresneho priestoru. RIP v2 uz podporuje. Potenciální problémy, které mohou nastat u tohoto směrovacího protokolu, jsou posílání paketů do neplatného směru nebo dlouhá doba konvergence. Tyto problémy se mohou projevit např. při kritických událostech, jako je výpadek linky. Tieto problemy sa riesia:

- Prvním řešením je informovat sousedy o výpadku spojení okamžitě místo využití pravidelných aktualizací rozesílaných po 30 s

- Split Horizon, zakazuje vysílání aktualizací o dané síti po rozhraní, po kterém byla informace o této síti přijata. Tato modifikace vyřeší výše popsaný problém, ale nedokáže eliminovat problémy v případě smyček v síťové topologii
- Split Horizon with Poison Reverse, které do směru, odkud byla přijata aktualizace o dané síti, vysílá směrovací informace o této sítí se vzdáleností 16 (síť je nedostupná), už pracuje správně i v případě smyček.

3 Směrovací protokoly Link-State, směrovací protokol OSPF.

3.1 Link-State

Tento typ směrovacích protokolů představuje pokročilejší způsob hledání „nejlepší“ cesty. Místo počítání přeskoků umožňuje přiřadit linkám váhové koeficienty, tzv. metriky, a pak hledat cestu s nejmenší celkovou metrikou. Směrovač informuje své sousedy o přiřazených metrikách pomocí zpráv Link State Agreement (LSA). Zpráva o metrice se neposílá zpět do směru, odkud byla přijata. V případě, že metrika cesty k dané síti v přijaté LSA zprávě je menší, než aktuální hodnota ve směrovací tabulce, tak si směrovač aktualizuje svůj záznam. V opačném případě neprovádí žádnou změnu. Dále pak rozešle svoji zprávu LSA svým sousedům. Směrovač je povinen předávat LSA zprávy na všech portech kromě portu, přes který byla zpráva LSA přijata. Ve zprávě LSA je uvedeno, která část informace od kterého směrovače pochází. K přijatým informacím každý směrovač přidá svoje znalosti o sítích a metrikách. Proto v ustáleném stavu všechny směrovače znají celou topologii sítě a mají stejné informace k rozhodování. Aby nedošlo k nekonečnému rozesílání LSA zpráv, tak musí být směrovač schopný rozpoznat, že danou zprávu už jednou viděl a podruhé ji musí ignorovat. Eliminace vícenásobného zpracování LSA zpráv může být zajištěna pomocí pořadového čísla zprávy, využitím pole time-to-live či využitím identifikace zprávy např. podle hodnoty kontrolního součtu.

3.2 OSPF

OSPF, konkrétní představitel rodiny link-state, patří mezi směrovací protokoly IGP. Podporuje dělení sítí na podsítě s využitím masek. Schopný eliminovat smyčky v topologii sítě.Pro přenos paketů využívá linky, které mají nejmenší celkovou metriku, ale v případě výpadku linky se dokáže automaticky přizpůsobit změnám a najít alternativní cestu využitím záložních tras. Metrika linek může být odvozena od zpoždění, fyzické vzdálenosti, zabezpečení, atd., ale nejčastěji je určena přenosovou rychlostí linky. Metriku cesty lze nastavovat velmi jemně, protože hodnota metriky se pohybuje v intervalu 1-65535. Celková metrika cesty není omezena. Součástí podpory alternativních linek je i podpora rozložení zátěže, kdy je provoz rozložen do více alternativních cest. Šíření směrovacích informací je automatické a provádí se formou záplav (flooding). I přes znalost celkové topologie sítě protokol OSPF pouze stanoví následující přeskok směrem k cílové síti, kam má být paket doručen. OSPF zprávy jsou zapouzdřeny přímo do IP paketů (nevyužívá se transportní protokol) a konzistence obsahu je zabezpečena algoritmem MD5. Jsou definovány tři základní typy zpráv, které souvisí s dílčími procesy směrovacího protokolu:

- Zpráva Hello – vysílá se pravidelně po všech rozhraních směrovače a slouží pro detekci a následnou pravidelnou kontrolu dostupnosti sousedních směrovačů.
- Zpráva s popisem databáze (OSPF Database Description Message) – slouží pro výměnu informací o topologii sítě mezi sousedními směrovači po fázi navázání spojení pomocí Hello zpráv. Výměna popisu databáze zajišťuje, aby směrovací informace byly ve všech směrovačích autonomní oblasti konzistentní. Protože databáze může být rozsáhlá, tak je popis často rozdělen do více zpráv. Zprávy vysílá nadřazený směrovač a podřízený je potvrzuje

- Zprávy popisující stav linky využívané pro dílčí aktualizaci vybraných záznamů databáze (zpráva OSPF Link Status Request), příp. pro šíření informací o nových přímo připojených linkách směrovače (zpráva OSPF Link Status Update).

4 Základní funkční bloky mechanismů pro zajištění kvality služeb.

- Klasifikace paketů
- Značkování paketů
- Dohled nad síťovým provozem
- Měření provozu
- Řízené odesílání paketů

4.1 Klasifikace paketů

Klasifikace je proces řazení paketů do skupin podle předem stanovených pravidel. Provádí se zpravidla na základě informací uložených do hlavičky datové jednotky.

- sloučené vyhodnocení (Behaviour Aggregate – BA) - vybírá pakety podle jediného identifikátoru (v hlavicke IP -DSCP). Paket, přicházející ke směrovači, byl již dříve označen v jiném síťovém prvku.
- vícepoložkové třídění (Multi-Field Calssification – MF)- vybírá pakety na základě jedné nebo více položek v hlavičce protokolu IP, příp. TCP/UDP, jako jsou: zdrojová adresa, cílová adresa či typ, zdrojový nebo cílový port transportního protokolu, resp. jejich kombinace.

4.2 Značkování paketů

Slouží pro označení příslušnosti datové jednotky ke své třídě. Realizováno nastavením hodnoty určitého pole v hlavičce IP datagramu (napr. IP adresa zdroje, IP adresa cílového uzlu, nebo jejich kombinace). Pokud byl paket již označen, daný směrovač ho může přeznačit (napr. pri prechode zo site do siete s inymi pravidlami znackovania, alebo když paket vybočuje z předem sjednaných parametrů přenosu).

- Internet Protocol Type of Service - IP ToS
  - V hlavičce protokolu IP
  - Osmibitové pole u IPv4
    - 3 bity – priorita podle požadavku zdroje
    - 3 bity – požadavky na pĜenos - 3. bit (D) – Zpoždění (Delay) , 4. bit (T) – Propustnost (Throughput) , 5. bit (R) – Spolehlivost pĜenosu (Reliability)
    - posledne 2 bity sa nepouzivaju
- Differentiated Service CodePoint (DSCP)
  - Technologie diferencovaných služeb (DiffServ) - mechanizmus na zaistenie kvality sluzieb
  - Prvních 6 bitů
    - Označení požadovaného způsobu zacházení - Per-Hop Behavior (PHB)
    - Určí okrajový směrovač pri vstupu paketu do sítě
  - posledne 2 bity sa nepouzivaju (CU - currently unused)

4.3 Dohled nad síťovým provozem

Dohled nad síťovým provozem zajišťuje, aby se datový tok vstupující do sítě pohyboval v mezích dohodnutých mezi zákazníkem a poskytovatelem připojení. Dohled se skládá z měření provozu a na základě výsledků měření se stanoví další způsob zpracování paketů datového toku, viz Obr. 3.5. Zvolený způsob zpracování může vést k zachování původně přidělené značky, k přeznačení paketů jinou značkou, či k zahození paketu.

4.4 Měření provozu

Založen na kontrole provozu přicházejícího na vstupní porty. Nejčastěji se ověřují následující dva parametry provozu:

- garantovaná průměrná přenosová rychlost (Committed Information Rate - CIR) - specifikuje dlouhodobou průměrnou rychlost dat, jejichž přenos je zaručen uživateli v rámci dohody SLA. Parametr CIR se měří v bytech/s. Pakety su prenasane v burstoch –> CIR < PIR
- maximální okamžitá přenosová rychlost (Peak Information Rate - PIR) - určuje maximální povolený počet bitů odeslaných v jednom okamžiku, předem sjednaný mezi poskytovatelem připojení a uživatelem v dohodě o úrovni služeb (Service Level Agreement - SLA)

4.4.1 Mechanizmus Token Bucket

Mechanismus TB lze popsat dvěma parametry: rychlostí doplňování tokenů r a velikostí nádoby b. Největší povolený shluk přicházejících paketů tedy odpovídá objemu nádoby b a dlouhodobá průměrná rychlost zpracování příchozích dat odpovídá rychlosti doplňování tokenů do nádoby r. Dlouhodobý průměr rychlosti přicházejících dat tedy nesmí překročit rychlost doplňování tokenů a krátkodobé špičky nesmí překročit velikost nádoby, jinak může dojít k zahození nebo jinému alternativnímu způsobu zpracování paketů.

4.5 Řízené odesílání paketů

Klíčem k zajištění odlišného zacházení různých datových toků ve směrovačích je řazení paketů do oddělených front a diferencovaný způsob odesílání paketů z těchto front. Kromě samotného odesílání paketů podle příslušného prioritního mechanismu je dalším důležitým úkolem řízení odesílání, dohled nad dostupnými síťovými prostředky, především nad šířkou pásma odchozího portu. Plánování odesílání paketů provádí každý výstupní port samostatně. Na základě informací ve směrovací tabulce jsou příchozí pakety nejdříve přeneseny na požadovaný výstupní port. Každý výstupní port provede klasifikaci paketů a řadí je do odpovídající fronty. Následně pak blok řízení určí, ze které fronty bude odeslán paket na výstup. Mezi nejběžnější metody řízeného odesílání paketů patří:

- fronta typu First-In-First-Out (FIFO),
- prioritní systém front (Priority Queuing - PQ),
- systém front se spravedlivou obsluhou (Fair-queuing – FQ),
- systém front s váženou cyklickou obsluhou (Weighted Round Robin - WRR),

- systém front s váženou spravedlivou obsluhou (Weighted Fair Queuing – WFQ),
- systém front založený na třídách s váženou spravedlivou obsluhou (Class-Based Weighted Fair Queuing – CB WFQ).

5 Způsob využití základních typů rámců technologie WiFi. Účel a dílčí kroky jednotlivých fází připojení klienta do sítě WLAN (skenování, autentizace, asociace, stav připojení, odpojení, roaming).

5.1 Způsob využití základních typů rámců technologie WiFi

- Rámce pro management
  - Beacon – Základní rámec WLAN sítě. AP ho posílá klientským stanicím. Obsahuje: základní parametry použité technologie (např. Číslo kanálu DSSS), SSID, Informace o podporovaných přenosových rychlostech, synch.času, Identifikátor nových dat pro stanice nacházejících se v režimu spánku (Traffic Indication Map TIM = Z režimu spánku se stanice probudí pouze na příjem rámce beacon).
  - Association request – Inicializuje klient vysláním žádosti tj. request. Po asociaci klient může posílat data přes WLAN
  - Association response – Odpoveď AP na žádost, buď ji přijme nebo odmítne.
  - Reassociation request – Obnovení spojení po krátkodobém výpadku spojení. Inicializuje klient vysláním žádosti
  - Reassociation response – Odpoveď AP na žádost reasociace
  - Disassiociation request/response – Vybudování nového spojení po změně AP v rámci handoveru (Podpora předávání klientských stanic mezi AP). Klient ruší spojení s aktuálním AP. Pak klient požádá o asociaci nový AP
  - Authentication – První krok připojení klienta do bezdrátové datové sítě. Inicializuje klient zasláním žádosti k Acess Pointu (AP). Na žádost může přímo odpovědět AP, nebo ji může předat dál k autentizačnímu serveru
  - Deauthentication – odmietnutie Authentication
- Řídicí rámce
  - Request to send (RTS) – zajistí stanici nerušený přístup ke sdílenému médiu. Přístupovou metodu inicializuje přímo stanice, která vyžaduje výhradní přístup. Je určeno pro řešení mimořádných událostí v síti (extrémně zatížené, se skrytým uzlem). Jedná se o rozšíření přístupové metody CSMA/CA. Stanice, která chce posílat rámec pomocí RTS, musí nejdříve vyhrát soutěžení o přístup k médiu. Když získá přístup, může odeslat rámec RTS a tím zahájit rezervaci média
  - Clear to send (CTS) – Potvrzení rezervace RTS.
  - Acknowledgement (ACK) – Je ním potvrzen úspešný prenos rámce a tím je ukončena i rezervace média.
  - Power Save Poll (PS Poll) – Rámec módu se sníženou spotřebou

  - Connection-Free End (CF End) – Konec řízení prístupu , vysíla jej AP v rámci metody PCF (Centralizovaná koordinační funkce - Point Coordiantion Fuction), co je přístupová metoda bez soutěžení. AP se pravidelně dotazuje stanic a zjišťuje, zda nemají data k vysílání. Okrem nej existuje DCF – distribuovaná, využíva súťaženie o médium.
  - CF End + ACK – konec řízení prístupu a zároveň odpověď na úspešný prenos rámce když to nestihl spávou ACK před ukončením.

- Datové rámce

5.2 Účel a dílčí kroky jednotlivých fází připojení klienta do sítě WLAN (skenování, autentizace, asociace, stav připojení, odpojení, roaming)

Připojení k bezdrátové sítí se odehrává ve dvou stupních. První stupeň se nazývá **autentizace** a představuje proces ověření totožnosti bezdrátového klienta. Druhým stupněm připojení je proces **asociace**, po kterém už klient může posílat data přes WLAN. Klient vysílá žádost association request a čeká na odpověď association response.

5.3 Skenování

Proces skenování se dělí na dva typy **aktivní** a **pasivní**. V případě **pasivního** skenování klient poslouchá na každém kanálu po určitou dobu a čeká na beacon rámec. V případě, že na daném kanále přijme klient beacon rámec, vyhledá v něm SSID. V případě více stanic se stejným SSID vybere bod s nejsilnějším signálem a s nejnižší bitovou chybovostí.

Při **aktivním** skenování klient sám generuje testovací rámce označené anglickým názvem probe request. Testovací rámec může obsahovat konkrétní SSID bezdrátové sítě, kam se chce připojit nebo broadcast SSID. Když je uvedeno konkrétní SSID odpovídá pouze dané zařízení. V případě broadcast odpovídá každé zařízení, které přijalo beacon rámec.

5.4 Autentizační metody

Probíhá ověření, zda má uživatel dostatečné oprávnění pro vstup do lokální sítě. Metod je několik, důležité je aby je podporoval přístupový bod.

Open System authentication Vychází ze standardu 802.11 jako výchozí metoda pro bezdrátové sítě. Není nutná znalost hesla, či klíče. Povoluje přístup do sítě všem klientům, u kterých je nastavené SSID shodné s SSID přístupového bodu. Metodu lze použít spolu se zabezpečovacím algoritmem **WEP**. V případě této metody nedochází k ověření wep klíče. Klíče využívá pro zakódování přenášených dat až po procesech autentizace a asociace.

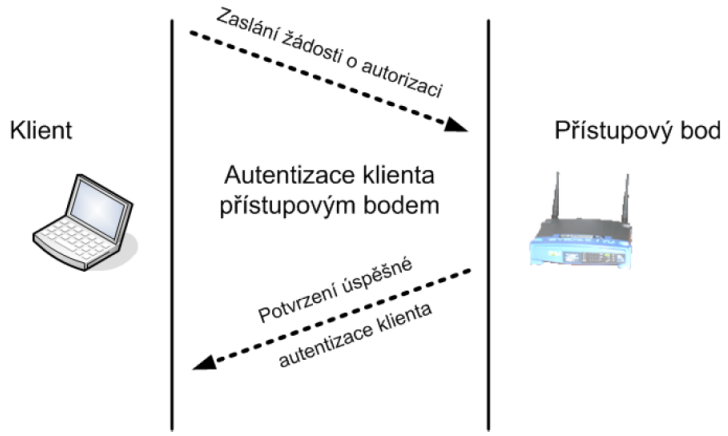
Shared Key Authentication Metoda musí pracovat s algortimem WEP, která využívá klíče na straně klienta i na straně přístupového bodu. Pro správnou funkci musí být klíče stejné na obou stranách.

Proces autentizace

Klient pošle přístupovému bodu žádost o autentizaci.

AP vygeneruje náhodný text a zašle ho klientovi.

Klient zašifruje přijatý řetězec svým WEP klíčem.



Obrázek 1: Autentizace klienta přístupovým bodem

– V zašifrovaném tvar ho zašle zpět přístupovému bodu.

AP dešifruje řetězec svým WEP klíčem.

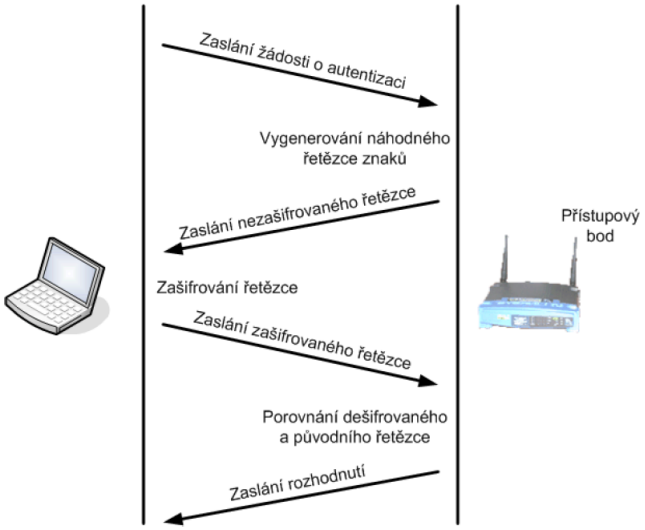
– Vyhodnotí shodu původního a dešifrovaného řetězce.

AP může ověřit, zda klient používá správný WEP klíč.

– Když ano, proces ověření klienta skončí úspěšně.

– Když ne, žádost klienta je odmítnuta.

Oznámení o výsledku rozhodnutí: Authentication response.



Obrázek 2: Autentizační metoda Shared Key Authentication

Při **Shared Key Authentication** je možné odvodit WEP klíč díky zasílání otevřeného a zašifrovaného řetězce. Zabezpečení přenosu dat nebude účinné. Při **Open System Authentication** je obtížnější odhalení WEP klíče, který je použit pouze pro přenos dat.

5.5 Asociace

- Druhý krok připojení klienta od WLAN
- Po asociaci klient může posílat data přes WLAN
- Inicializuje klient vysláním žádosti: association request
- AP žádost přijme nebo odmítne: assicoation response.

5.6 Stav připojení

Řízení přístupu CSMA/CA metodami: PCF a DCF. Přenost dat.

5.7 Reasociace

Obnovení spojení z důvodu ztráty spojení nebo kvůli velkému provozu v síti.

Obnova spojení
Klient vyšle zprávu reassociation request.

Přístupový bod odpovídá zprávou reassociatino response.

Vybudování nového spojení
Klient ruší spojení s aktuálním AP: disassociation request.

Klient požádá o asociaci nový AP reassociation request – reassociation response.

5.8 Roaming

Přecházení uživatelů mezi přístupovými body. Probíhá na druhé nebo třetí vrstvě. Na druhé vrstvě se uživatel přesunuje od jednoho přístupového bodu k druhému v rámci téže IP sítě / podsítě. Tj. jenom fyzické předání uživatelů mezi dvěma přístupovými body (handover). Pokud se přesouvá do jiné IP sítě, nemůže nadále používat nadále na venek používat původní IP adresu. K tomu je potřeba Mobile IP. Řeší tento problém pomocí domácí adresy, která je neměnná a cizí adresy, kterou si klient půjčuje v navštívených sítích.

6 Deterministické a náhodné přístupové metody využívané v sítích WLAN.

6.1 CSMA/CA

- Kolize
  - Sdílené medium
  - Náhodný přístup
  - U WLAN sítí je obtížné detekovat kolizi, proto se jí snaží jen vyhnout
- Přístupová metoda
  - Využívá zpětné potvrzování přijatých datových rámců
  - Snižuje efektivitu systému
  - Využívá náhodné čekací doby v případě, že medium není volné

6.2 Náhodna přístupova metoda

6.2.1 Distribuovaná koordinační funkce (DCF – Distributed Coordination Function)

- Specifikována v standardu 802.11
- Lze využít v BSS, ESS i IBSS
- Využívá náhodnou přístupovou metodu
- Stanice soutěží o přístup k médiu:
  - stanice si vygenerují náhodné číslo
  - každá stanice postupně snižuje svoje náhodné číslo každým hodinovým impulzem a na konci hodinového impulzu zkontrolují obsazenost média
  - Po uplynutí náhodně generovaného čekacího intervalu = Dosažení hodnoty 0 => zahájení vysílání
  - Ostatní stanice zjistí, že medium už není volné a přestanou odpočítávat. Zapamatování aktuální hodnoty u ostatních stanic - Využití během následujícího soutěžení
  - Po příjmu cílový uzel musí zkontrolovat přijatý rámec a zpětně potvrdit správný příjem rámcem ACK

6.3 Deterministicka přístupova metoda

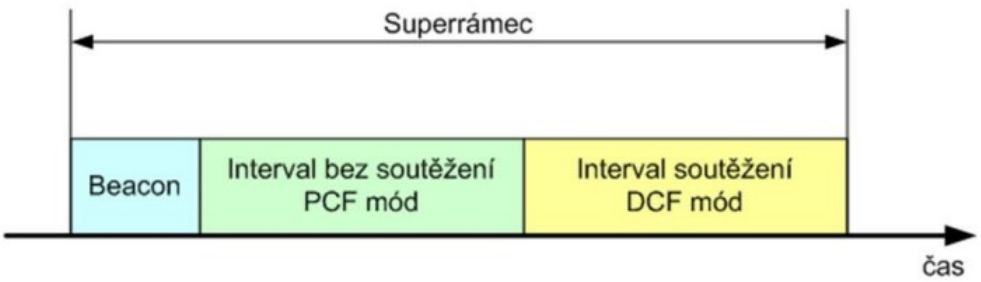
6.3.1 Centralizovaná koordinační funkce (PCF - Point Coordiantion Fuction)

- Přístupová metoda bez soutěžení – řízená tj. deterministická
- AP pravidelně dotazuje stanic a zjišťuje, zda nemají data k vysílání
- Vhodné pro zajištění QoS

Obr. 6.31 Ukázka komunikace v módu DCF

- Vyžaduje přítomnost AP
- Před využitím PCF musí klient nejdĜív sdělit AP, že umí odpovídat na dotazy
- Musí pracovat v kombinaci s DCF = Využíva Superrámec
- Průběh komunikace PCF:
  - Proces je zahájen vysláním rámce „beacon“
  - Interval bez soutěžení (Contention-Free Period - CFP)
    - Řízení přístupu PCF – dotazování
    - Zprávy CF-Poll – CF-ACK
    - Ukončí AP zprávou CF-End
  - Nasleduje Interval soutěžení (Contention Period - CP) podle DCF
    - Využívá DCF (soutěžení stanic o přístup k médiu)
    - První stanice, která získá přístup k médiu odešle svůj datový rámec
    - Cílová stanice, resp. přístupový bod počká SIFS a pošle rámec ACK
    - Příjem ACK znamená, že data byla doručena bez problémů
    - Následuje další komunikace, zahájena příslušnou mezirámcovou mezerou

- Superrámec
  - Existuje pouze v případě, že se využívá PCF



7 Vlastnosti základních algoritmů výběru buněk (PIM, iRRM, SLIP, DRRM).

7.1 PIM algoritmus

Algoritmus Paralell Iterative Matching – PIM využívá mechanizmus náhodného výběru. Buňky vstupující do propojovacího uzlu jsou uloženy do front VOQ a výběr proběhne iteračním procesem. Každý iterační krok je složen ze tří fází. Na začátku není žádný ze výstupních a výstupních portů označen a jen výstupní a výstupní porty, které budou neoznačeny, mohou vstoupit do následujícího iteračního kroku. Zmíněné tří fáze iteračního kroku jsou následující:

- Inicializace
- Požadavky
- Potvrzení
- Výěr
- Přidělení páru vstup – výstup

Inicializace

Žádný ze vstupních a výstupních portů není označen.

Neoznačené vstupní a výstupní porty mohou vstoupit do iteračního procesu.

Požadavky

Žádost od neoznačených vstupních portů.

Všem výstupním portům, pro které mají buňky.

Potvrzení

Neoznačený výstupní port přijme více žádostí.

Vyhoví pouze jedné z nich.

Náhodný výběr – stejná pravděpodobnost výběru pro všechny žádosti.

Oznámení výsledku výběru vstupnímu portu (= samo potvrzení).

Výběr

Vstupní port může dostat více potvrzení.

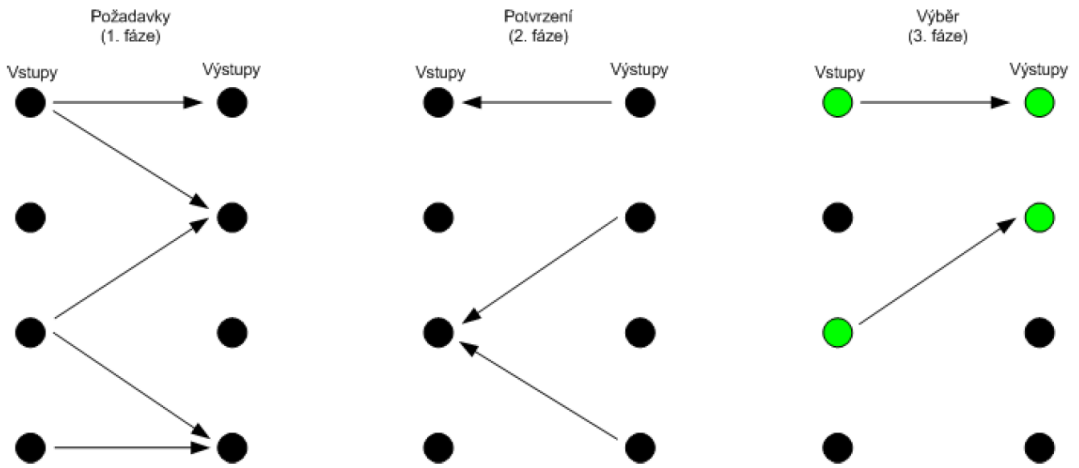
Vybere výstup náhodně.

Přidělení páru vstup – výstup

Vybrané vstupní a výstupní porty se stanou označenými.

Nebudou zahrnuty do dalšího iteračního kroku.

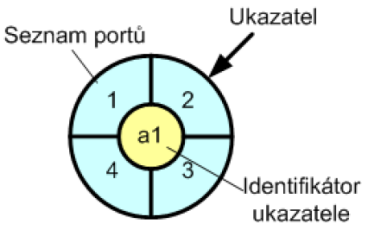
Průběh jednoho iteračního kroku je znázorněn na obrázku níže. V každém iteračním kroku je pak označeno průměrně 75% potvrzení. Iterace konverguje poměrně rychle, během $O(logN)$ iteračních kroků. Nevýhoda je náročná implementace rychlých náhodných funkcí. V případě rovnoměrného provozu dosahuje algoritmus PIM propustnost 63% pro 1 a 100% pro N iteračních kroků.



Obrázek 3: Jeden iterační krok algoritmu PIM

7.2 iRRM algoritmus

Iterative Round – Robin Matching, podobný algoritmu PIM. Místo náhodného výběru využívá mechanizmu round – robin a to jak u výstupního, tak u vstupního portu. Každý arbiter (řídící model) spravuje ukazatel ukazující na port, který má v daném časovém okamžiku nejvyšší prioritu. U vstupního portu se ukazatel nazývá **accept pointer** a u výstupního portu se nazývá **grant pointer**.Na obrázku níže je znázorněn ukazatel u 4 – portového propojovacího uzlu.



Obrázek 4: Ukazatel algoritmu round – robin

7.2.1 Popis procesu

Inicializace

Žádný ze vstupních a výstupních portů není označen.

Iterace

Žádost od neoznačených vstupních portů.

- Všem vstupním portům, pro které mají buňky.

Neoznačený výstupní port přijme více žádostí

- Vybere žádost od vstupu s nejvyšší prioritou.

- Příp. od nejbližšího následujícího portu v seznamu.

Oznámení výsledku výběru.

- Výstupy informují vybrané vstupní porty.
- Informují také nevybrané vstupní porty.

Nastavení ukazatele arbitru.

- Na port následující hned za právě vybraným.
- Výstupy, které nedostaly žádost – ukazatel zůstane v původním stavu.

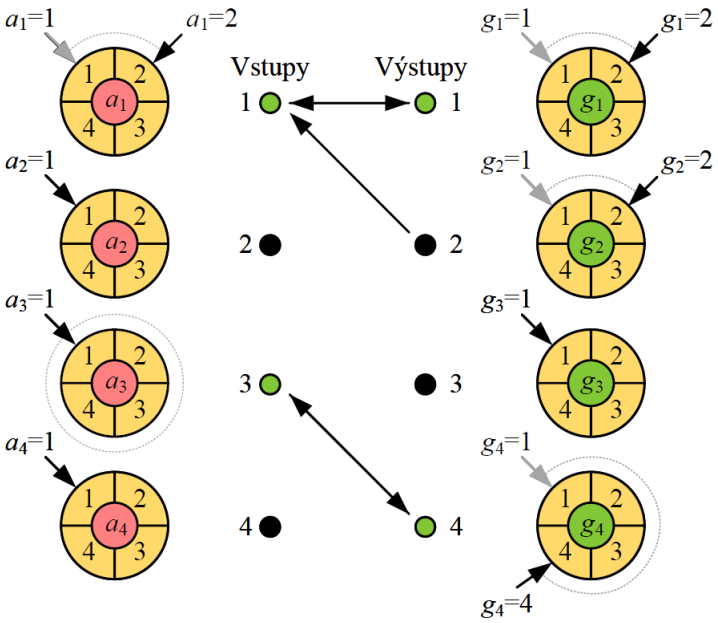
Pokud vstupní port obdrží více potvrzení

- Výběr potvrzení.
- Od výstupního portu s právě největší prioritou.
- příp. od nejbližšího následujícího portu v seznamu.

Aktualizace ukazatele vstupního arbiteru

- Na port následující hned za právě vybraným.
- Vstupy, které neposlaly žádost, ukazatel nemění.

7.2.2 Příklad algoritmu iRRM



Obrázek 5: Příklad algoritmu iRRM

Inicializace

$$a_i = g_i = 1, pro \;\; i = 1, 2, 3, 4 \tag{1}$$

Žádost od neoznačených vstupních portů

Oznámení výsledku výběru

$$g_1 = 1 + 1, g_2 = 1 + 1, g_3 = 1 \;\; a \;\; g_4 = 3 + 1 \tag{2}$$

Výběr potvrzení a aktualizace ukazatele vstupního arbiteru

$$a_+ = 1 + 1, a_3 = mod_4(4 + 1) = 1 \tag{3}$$

7.3 iSLIP algoritmus

Zdokonalení algoritmu iRRM, došlo ke aktualizaci výstupního portu (grant pointer) a to pouze v případě, pokud je dané potvrzení akceptováno. Vybraná dvojice vstupního a výstupního portu bude mít vždy nejnižší prioritu. Spravedlivé přidělování šířky pásma datovým tokům a 100 % propustnost.

7.4 Kroky iteračního procesu

Inicializace
Žádný vstupní ani výstupní port není označen.

Žádost od neoznačených vstupních portů
Všem výstupním portům, pro které mají buňky.

Neoznačený výstupní port přijme více žádostí
Vybere žádost od vstupu s nejvyšší prioritou.

Příp. od nejbližšího následujícího portu v seznamu.

Oznámení výsledku výběru
Výstupy informují vybrané vstupní porty.

Informují také nevybrané vstupní porty.

Výstup zatím neaktualizuje ukazatele arbiteru.

Pokud vstupní port obdrží více potvrzení
Výběr potvrzení.

Od výstupního portu s právě největší prioritou.

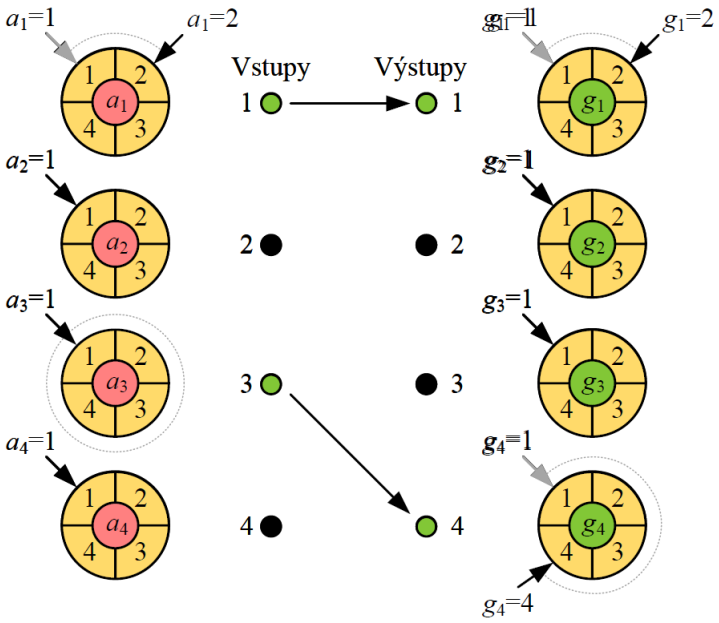
Příp. od nejbližšího následujícího portu ze seznamu.

Aktualizace ukazatele a_i vstupního arbitru
Na port následující hned za právě vybraným výstupem.

Pouze v prvním iteračním kroku.

Aktualizace ukazatele g_i výstupního arbitru
Na port následující hned za vstupním portem v seznamu, který akceptoval potvrzení.

Pouze v prvním iteračním kroku.



Obrázek 6: Příklad algoritmu iSLIP

7.5 Příklad iSLIP

Inicializace

$$a_i = g_i = 1, pro \;\; i = 1, 2, 3, 4 \tag{4}$$

Žádost od neoznačených vstupních portů

Oznámení výsledku výběru

Výběr potvrzení a aktualizace ukazatele vstupního arbiteru

$$a_+ = 1 + 1, a_3 = mod_4(4 + 1) = 1 \tag{5}$$

Aktualizace ukazatele výstupního arbiteru

$$g_1 = 1 + 1, g_4 = 3 + 1 \tag{6}$$

7.6 DRRM algoritmus

Dual Round – Robin Matching implementuje mechanizmus Round – Robin a využívá systém virtuálních front VOQ. Vstupní port se skládá z VOQ front a vstupního arbitru. Výstupní port obsahuje pouze výstupní arbitr.

Výběr žádosti

Řídí vstupní arbiter.

Využitím mechanizmu Round – Robin.

Výběr VOQ fronty, která obsahuje buňky k odeslání.

Zaslání žádostí
Každý vstup na příslušný výstupní port.

Výběr jednoho požadavku
Provede výstupní arbiter.

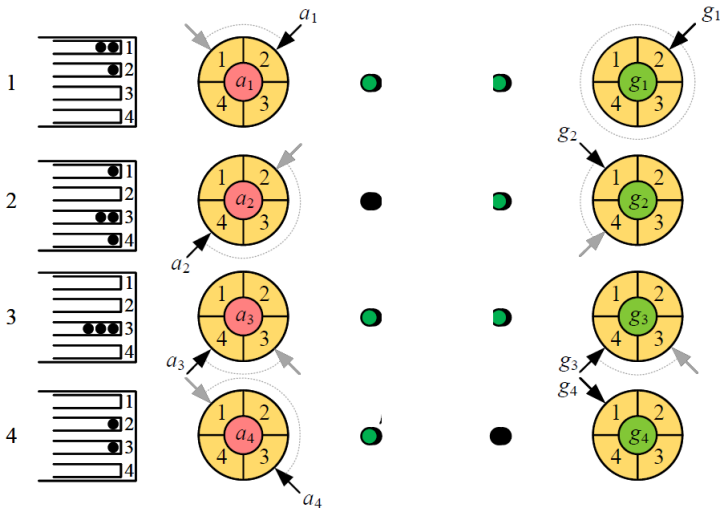
Využije mechanizmus Round – Robin.

Zaslání potvrzení

Aktualizace ukazatele vstupních portů
Puouze z vybraných vstupů.

Aktualizace ukazatele výstupních portů
Pouze u dotázaných.

7.7 DRRM příklad



Obrázek 7: Příklad algoritmu DRRM

Odeslání žádostí

Potvrzení a vytvoření párů vstup – výstup

Aktualizace vstupních a výstupních arbitrů

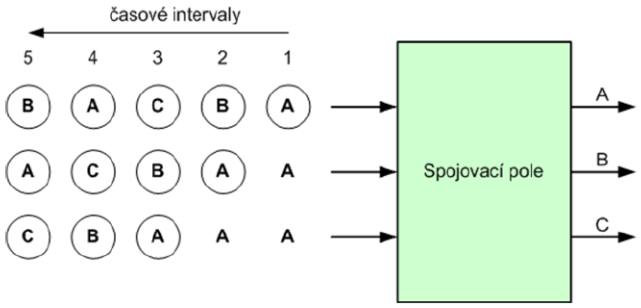
$$a_1 = 2, a_2 = 4, a_3 = 4, a_4 = 3 \tag{7}$$

$$g_1 = mod_4(2 + 4) = 2, g_2 = mod_4(4 + 1) = 1, g_3 = 4, g4 = g_4 \tag{8}$$

7.8 Desynchronizační efekt

Desynchronizační efekt:

- Vstupní arbitry žádají o různé výstupy
- Větší propustnost
- Příklad – jsou zobrazeny pouze první buňky z vybrané VOQ fronty



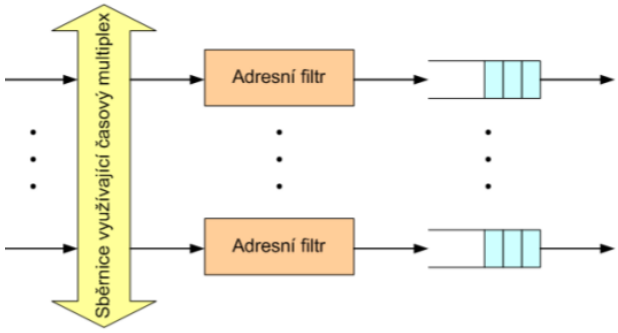
Obrázek 8: Desynchronizační efekt

Projevuje se u DRRM i iSLIP, DRRM je výhodnější – kratší výměna informací - větší rychlost.

8 Vlastnosti a struktura přepínače se sdíleným mediem a se sdílenou pamětí.

8.1 Prepináč so zdieľaným médiom

Vstupné porty prepínačov tohto typu sú pripojené k zdieľanému médiu pomocou časového multiplexu. V prípade, že N je počet portov, časový interval, zodpovedajúci **bunkovej perióde**, je rozdelený na N mini intervalov, z ktorých každý prislúcha jednému z vstupných portov. Daný port počas tohto mini intervalu môže zapísať svoje dáta na zdieľané médium (zbernica alebo kruh). Pretože je médium zdieľané medzi N portami, musí byť šírka pásma zdieľaného média N-krát vyššia, než je šírka pásma vstupného rozhrania. Z toho vyplýva, že priepustnosť zdieľaného média určuje kapacitu prepínača. Charakteristická architektúra takého prepínača je na Obr.12. Každá výstupná linka je pripojená na zdieľané vysokorýchlostné médium cez adresný filter a vyrovnávaciu pamäť typu FIFO. K bunke je bežne pripojená ešte **hlavička**, ktorá obsahuje bitovú mapu výstupných portov. Nastavením bitu príslušného odchádzajúceho portu je potom signalizované, že bunka je určená pre tento odchádzajúci port. Adresný filter overuje túto bitovú mapu buniek objavujúcich sa na zdieľanom médiu, a v prípade splnenia kritérií filtrovania, **bunku skopíruje do vyrovnávacej pamäte**.
Ďalším obmedzením môže byť rýchlosť zápisu do vyrovnávacej pamäte, kedy vyrovnávacia pamäť musí byť schopná zapísať až N buniek za jeden časový interval (z každého vstupu smeruje prevádzka na jeden výstup).Riadenie prepínača je **decentralizované** a preto každý výstupný port môže pracovaťnezávisle na ostatných. S nezávislosťou jednotlivých portov súvisí **menšia efektivita** využívania hardvérových prostriedkov, ktoré nie sú zdieľané medzi odchádzajúce porty. Môže tak vzniknúť situácia, že jeden z výstupných portov už zahadzuje bunky, pretože má preplnenú vyrovnávaciu pamäť, pričom ostatné vyrovnávace pamäte môžu byť úplne prázdne. Prepínače so **zdieľaným médiom** sa často využívajú aj v telekomunikačných systémoch, kde je **šírka pásma** vstupných liniek pomerne malá.



Obrázek 9: Architektúra prepínača so zdieľaným médióm

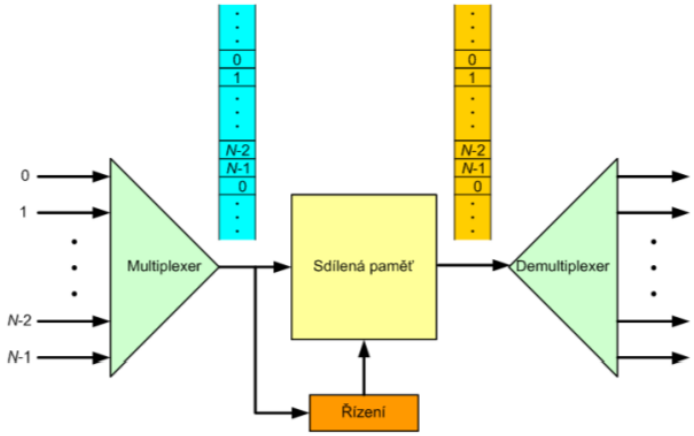
8.2 Prepínač so zdieľanou pamäťou

Pri tomto type prepínača sú prichádzajúce bunky radené do **časového multiplexu** a sú postupne zapisované do zdieľanej pamäťovej oblasti. Prepájanie je potom realizované čítaním buniek vo forme multiplexovaného toku dát, ktorý je potom demultiplexovaný a vedený k jednotlivým výstupom. Riadenie zápisu a čítania z pamäte vykonáva modul riadenia na základe informácií získaných z hlavičky buniek. Ukážka architektúry prepínača so zdieľanou pamäťou je na Obr.13.Prepínače so zdieľanou pamäťou vykazujú **najlepšie využitie prostriedkov**, pretože všetky vstupné aj výstupné porty zdieľajú jednu pamäťovú oblasť. Toto riešenie tiež umožňuje jednoduché prispôsobenie veľkosti pamäte k požiadavkám na stratovosť.Rýchlosť operácií s pamäťou je rovnako N-krát vyššia, než rýchlosť linkových rozhraní. Preto rýchlosti pamäte priamo obmedzujú kapacitu prepínača. Ďalšou nevýhodou je zložité riadenie zápisu a čítania z pamäte.

Existujú **dve metódy zdieľania pamäťového priestoru medzi portami**:

1. Úplne rozdelenie – pamäťový priestor je rozdelený na N rovnakých častí, kde každá dielčiačasť je priradená jednému zo vstupných portov a bunka je následne zapísaná do pamäte podľa toho, pre ktorý výstupný port je určená

2. Plné zdieľanie – celý pamäťový priestor je zdieľaný všetkými vstupnými portami, čo je efektívnejším využitím pamäte, musia však existovať algoritmy, ktoré skúmajú využitie pamäte jednotlivými portami a zabraňuju monopolizácií



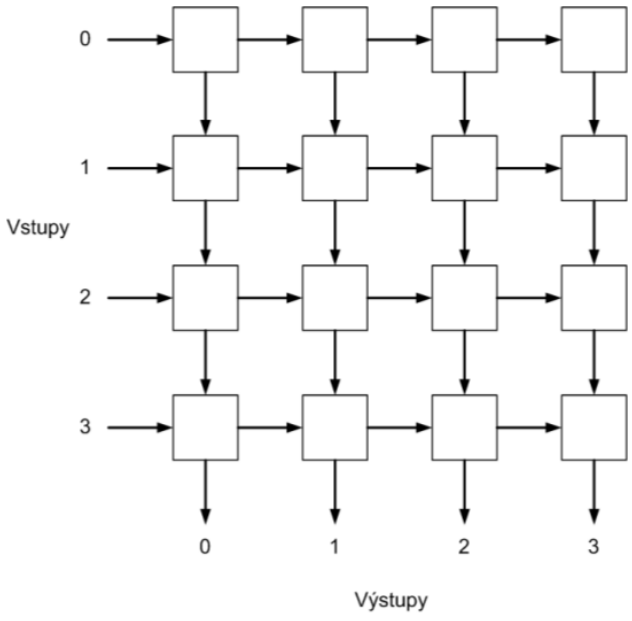
Obrázek 10: Architektúra prepínača so zdieľanou pamäťou

9 Struktura spojovacích polí s prostorovým dělením kanálu (křížový přepínač, plně propojená struktura, Banyan).

Pri tejto skupine prepináčov existuje **viac ciest** medzi prichádzajúcimi a odchádzajúcimi portami. Je možné ich využiť na súčasný prenos viacerých buniek. Tým je dosihanuté, že **priepustnosť prepojovacieho uzla** je násobkom šírky pásma ciesty a počtu ciest, ktoré súčasne prenášajú bunky. Prakticky je kapacita prepínača obmedzená faktormi, ktoré súvisia s fyzickou implementáciou, napr. vývod súčiastok či problém so synchronizáciou. Viaccestné prepínače sú odolnejšie vočí poruchám, jednocestné zase jednoduchšie na implementáciu.

9.1 Krížový prepínač

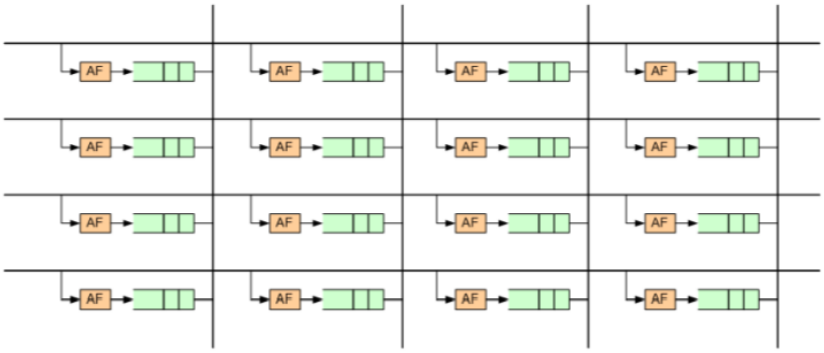
Prepojovací uzol vyžaduje krížový prepináč so štruktúrou $N * N$, obsahujúci N^2 samostatne riadených spínacích prvkov,každý odpovedá dvojici vstup–výstup. Každý spínací prvok má **dva stavy**, a to, rozopnutý stav , ktorý je továrenský a zopnutý stav. Spojenie medzi vstupným portom a výstupným je dosiahnuté uvedením spínacieho prvku do zopnutého stavu. Je možné spúšťať u každého spínacieho prvku uvedenie do zopnutého stavu automaticky príchodom bunky. Proces prebieha nezávislo na ostatných bunkách. Pole sa tak stáva **samosmerovacím** – riadiace funkcie sú distribuované medzi spínacie prvky.Výhodou tejto architektúry je nemožnosť vnútorného blokovania, jej modulárnosť a jednoduchosť. Obmedzením je však narastanie spínacích uzlov kvadraticky. Rozhodovanie o výbere bunky pre daný port a časový interval sú častým úzkym miesto väčšieho spojovacieho poľa.Na obrázku 14 je znázornený krížový prepínač,kde vodorovné linky značia vstupy a zvislé výstupy.



Obrázek 11: Krížový prepínač

9.1.1 Vyrovnávacia pamäť v spínacích prvkoch

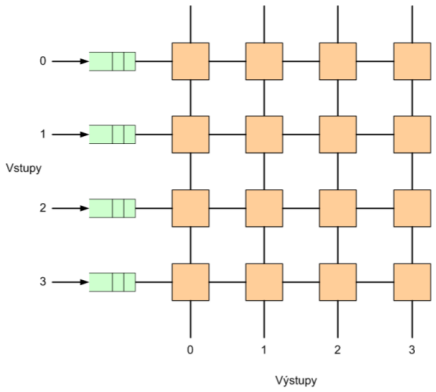
V každom spínacom bode sa nachádza adresný filter, ktorý bunky s cieľovou adresou odpovedajúcej príslušnému výstupnému portu prepúšťa do vyrovnávacej pamäte. Z buniek, ktoré čakaú vo vyrovnávacej pamäti je nutné vybrať, ktorá bunka v ktorom časovo okamihu môže byť prenesená na výstup. Architekrúra netrpí blokovaním fronty. Výstupný buffer portu je rozdelený na N menších pamätí. Nie je využité zdieľanie pamätí, preto je nutné využitie viac pamätí pre danú veľkosť stratovosti. Nevýhodou je taktiež technologické riešenie,pretože pamäte musia byť integrované spolu s riadiacou logikou spojovacieho poľa do jedného čipu, tým je obmedzený počet spínacích prvkov.



Obrázek 12: Krížový prepínač s vyrovnávacou pamäťou v spínacích prvkoch

9.1.2 Vyrovnávacia pamäť na vstupe

Prijatá bunka je najprv uložená do vstupnej vyrovnávacej pamäte a čaká, aby mohla vstúpiť do spojovacieho poľa. V prípade konfliktov o prístup na rovnaký výstupný port ,ktoré sú riešené distribuovane sa algoritmus výberu robí zvlášť v každom spínacom bode. Keď sa bunka dostane k spínaciemu miestu, pre daný vstupný port je to oznámené blokačným signálom. Následne výstupný port preruší vysielanie bunky a nechá ju vo vyrovnávacej pamäti. V prípade centralizovaného výberu buniek pre každý výstupný port zvlášť prebehne výber tak, aby bola pre každý port len jedna bunka.



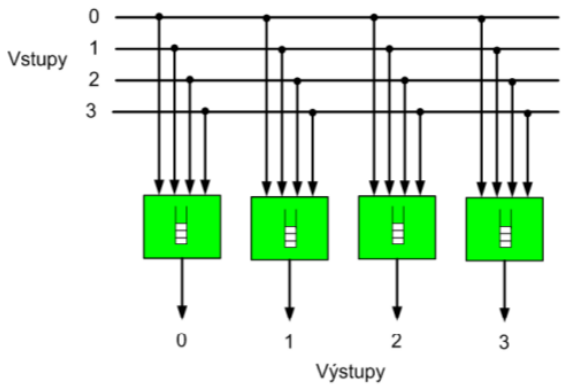
Obrázek 13: Krížový prepínač s vyrovnávacou pamäťou na vstupe

9.1.3 Vyrovnávacia pamäť na výstupe

Veľmi podobné vlastnosti, ako v štruktúre s výrovnávacou pamäťou v spojovacom uzle.Zarovnanie pamäte portu vytvára jednu oblasť, čo prináša efektívnejšie využitie pamäte.

9.2 Prepínač na plne prepojené štruktúry

Tieto prepínače sa vyznačujú tým, že každý vstup ma samostatné spojenie s každým výstupom. Toto spojenie je dosiahnuté pomocou N zberníc, kde každa zubernica vedie of konkrétneho vstupného portu ku všetkým výstupným portom. Spojovacie pole vyžaduje N samostatných vyrovnávacích pamätí na výstupných portoch. Bunka je rozoslaná na každý vstupný port. Bunky od rôznych vstupných portov sú súčasne privedené na rovnaký výstupný port. Výstupný port tak robí filtrovanie buniek a ich ukladanie do vyrovnávacej pamäte. Každy port má vlastný filter buniek a vyrovnávaciu pamäť. Štruktúra je jednoduchá a neblokujúca.

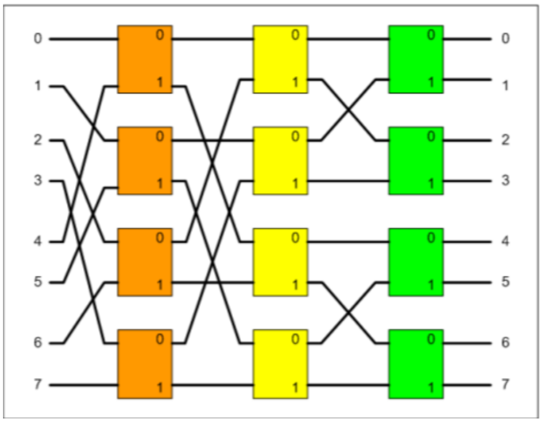


Obrázek 14: Prepínač na plne prepojené štruktúry

9.3 Prepínáč na spojovacie pole typu Banyan

Jedná sa o jednocestný a samosmerovací prepínač zostavený zo spínacích elementov 2x2.Poznáme 3 topológie – Banyan, Delta, Omega.

Počet spínacích elementov a ciest v tejto rodine spojovacích polí narastá $N\log N$-krát, čo je v prípade veľkých spojovacích polí výrazne lepšie. Samosmerovací algoritmus nevyžaduje prídavný riadiaci modul a ako všetky spojovacie polia so zdieľaným médiom umožňuje paralelný prenos viacerých buniek. Samozrejme, táto štruktúra má aj niekoľko nevýhod. Najvýznamnejšia z nich je, že môže nastať vnútorné blokovanie a priepustnosť spojovacieho poľa rapídne klesá s nárastom veľkosti. Pokles priepustnosti je možné kompenzovať využitím spínacích prvkov MxM, kde M>2 namiesto prvkov 2x2. Napriek zvýšenej priepustnosti ale spojovacie pole zostane stále blokovacím.

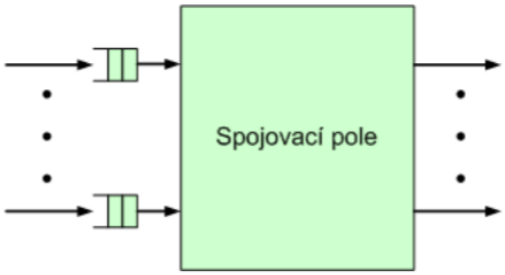


Obrázek 15: Prepínáč na spojovacie pole typu Banyan

10 Výhody a nevýhody architektur pro přepojovací uzly využívající: vstupní vyrovnávací paměti, výstupní vyrovnávací paměti, sdílenou paměť, a virtuální výstupní fronty.

10.1 Vstupná vyrovnávacia pamäť

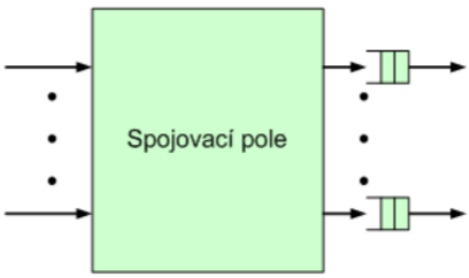
Najväčšou nevýhodou je možnosť blokovania rady, ktorá spôsobuje pokles priepustnosti v prípade veľkého prepináča až na hodnotu 58,6 %. Pre zvýšenie priepustnosti je možné využiť systém okna, kedy je zo vstupného vyrovnávacieho portu vybraných viacero buniek určených pre potenciálne odoslanie. Prenesená však bude maximálne len jedna. Počet potencionálne prenositelných buniek určuje veľkosť okna. Zvýšením veľkosti okna na dve sa zvýši priepustnosť na 70 %. Výhodou je ukladanie buniek vo vyrovnávacej pamäti.



Obrázek 16: Štruktúra prepojovacieho uzla so vstupnou vyrovnávacou pamäťou

10.2 Výstupná vyrovnávacia pamäť

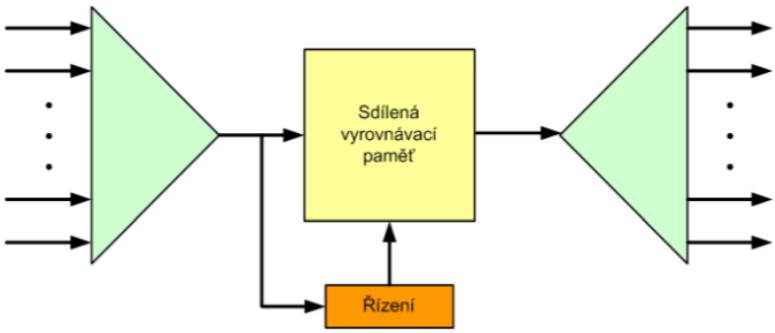
Prepojovací uzol s výstupnou vyrovnávacou pamäťou umožňuje, aby každá bunka, ktorá vstúpi do spojovacieho poľa, bola doručená na požadovaný výstupný port, tj. priepustnosť tejto štruktúry je 100 %. Výstupný port však môže prijať v jednej bunkovej perióde až N buniek, čo kladie veľké nároky na rýchlosť spracovania a ukladanie buniek do vyrovnávacej pamäte portu.



Obrázek 17: Štruktúra prepojovacieho uzla s výstupnou vyrovnávacou pamäťou

10.3 Zdieľaná pamäť

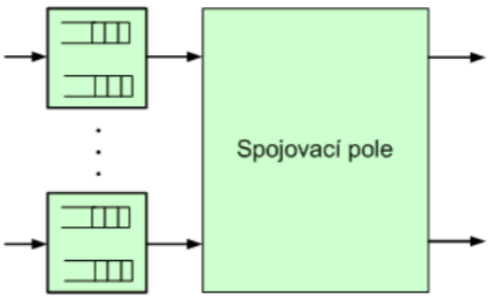
Výhodou je vysoká priepustnosť, nízke oneskorenie a veľmi efektívne využitie pamäte hlavne pre menšie prepojovacie uzly.V prípade viacvrstvových štrukúr spojovanie viacerých spojovacích polí do jednej štruktúry vedie k zníženiu efektivity. Uloženie bunky do viacerých vyrovnávacích pamäti môže viesť k zmene poradia, čo je nutné eliminovať zložitým a drahým prídavným zariadením.



Obrázek 18: Štruktúra prepojovacieho uzla so zdieľanou vyrovnávacou pamäťou

10.4 Virtuálne výstupné rady

Výhodou tohto uzla je predovšetkým eleiminácia blokovania rady, čo zabezpečuje zväčšenie priepustnosti celého uzla. Vstupná vyrovnávacia pmäť je rozdelená do N logicky rozdelených rád a každá odpovedá jednému z výstupných portov. Nevýhodou je však zložité riadenie rád a potreba inteligentného výberu bunky pre vstup do spojovacieho uzla. Systém musí pracovať s $N * N$ virtuálnymi radami v každej bunkovej periódou.



Obrázek 19: Štruktúra prepojovacieho uzla s virtuálnou výstupnou radou