**11. Přepínání sítě**

**Switch — Princip činnosti**

* Aktivní prvek — Přepínač
* Přepíná provoz mezi více připojenými segmenty současně
* Přepínání je řešeno hardwarovými prostředky s vysokým stupněm paralelizace (ASIC obvody)

**Základní funkce**

* Přepíná rámce → Rozhoduje kam půjdou
* Switch obsahuje MAC tabulku → Mapuje cílovou adresu MAC na konkrétní port
  + Díky ní pak rozhoduje, na jaký port má rámec zaslat
* Naopak od hubu, **nezasílá všechny přijaté rámce na všechny porty** - zvyšuje bezpečnost sítě (a taky tím zabraňuje zbytečný přenos)
* Pracuje na spojové vrstvě (L2)

**Přepínání rámců**

1. Switch přijme rámec na ingress portu (příchozí port)
   1. Pokud zdrojovou MAC nezná, rovnou si ji zapíše do MAC tabulky
2. Podle cílové adresy v rámci a záznamech v MAC tabulce se rozhodne, na jaký egress port rámec přepne (odchozí port)
   1. Pokud cílovou MAC zná a má ji namapovanou na port → Přepne rámec na tento port
   2. Pokud cílovou MAC nezná → Zašle přijatý rámec broadcastově a čeká na odpověď
      1. Až odpověď přijme, zapíše si zdrojovou MAC rámce do MAC tabulky

<aside> 💡 Tím se efektivně naučil MAC adresu dalšího počítače

**Metody přepínání**

**Store-and-forward**

* Přijme celý rámec (**vyšší zpoždění)**
* Ověří délku a kontrolní součet
* Nevyhovující rámce zahodí

<aside> 💡 Například když kontrolní součet se nerovná či se jedná o moc malý / velký rámec

* Uloží si MAC adresu odesílatele, pokud ji ještě nezná
* Přepne rámec na příslušný port
  + Pokud cílovou MAC má namapovanou na port → Zašle to na ten port
  + Pokud ne → Odešle rámec na ostatní aktivní porty

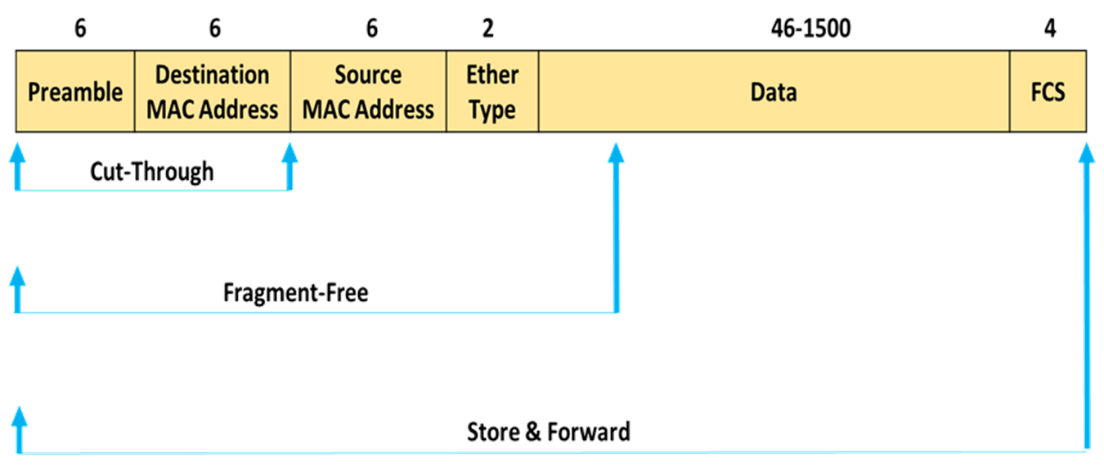
**Cut-through**

* Přijme alespoň cílovou MAC adresu rámce
* Již zahájí jeho zpracování — Příjem rámce pokračuje
  + Nekontroluje se kontrolní součet ani délka — Takto se můžou dostat invalidní rámce dál do sítě (nejsou zkontrolované)
* Přepíná rámec na příslušný port
  + Zase → Pokud zná cílovou MAC, přepne to na namapovaný port, pokud ne tak odešle rámec broadcastově

**Tato metoda má ještě dvě varianty:**

1. **Fast-forward** 
   * Zahájí odesílání rámce ihned po přijetí cílové MAC
   * Nejrychlejší způsob přepínání, ale vůbec se nekontroluje, zda se jedná o vyhovující rámec
2. **Fragment-free** 
   * Zahájí odesílání rámce až po přijetí 64 bytů (kde nastává nejvíce kolizí/chyb)

<aside> 💡 Fragment-free nabízí kompromis mezi integritou a zpožděním.



**Virtuální sítě (VLAN)**

* Vytváří se na switchých
* Umožňuje rozdělit fyzickou síť na logickou
  + Zajišťuje lepší správu sítě a její zabezpečení
* VLAN dokážou oddělit a izolovat provoz různých zařízení na jedné fyzické síti
  + Tím omezí přístup ke zdrojům na základně přirazení access portů do jednotlivých VLAN
* Každá VLAN je svá broadcatová doména

**Vlastnosti VLAN**

1. Bezpečnost
   * Izolace datového provozu do virtuálních sítí
   * Tím počítače vidí jen data, která opravdu patří do jejich VLAN
2. Snadná správa
   * Správa sítě je zdjednodušena vytvořením “skupin”
   * Lépe se identifikují chyby, např. pokud se nachází na VLAN 10 ale ne na VLAN 20, tak je něco špatně na VLAN 10
3. Lepší využití síťových prostředků
   * Ve výsledku se můžeme dostat do bodu, kdy broadcastový rámec nerozešle po velké síti, ale jen v právě té VLAN, do které patří

**Realizace VLAN**

* Vytvoříme VLAN na jedno (či více switchů)
* Jednotlivé VLAN přiřadíme buďto
  + Access portům — Porty vedoucí k jednomu uzlu, např. uživatelský počítač
  + Trunk portům — Porty vedoucí k dalšímu síťovému aktivnímu prvku, např. switch

**Druhy VLAN**

* Data VLAN — Pro obyčejný uživatelský provoz
* Default VLAN — VLAN 1, nelze smazat či ji přejmenovat
* Native VLAN — Přirazena trunk portu z důvodu zpětné kompatibility switchů bez podpory VLAN
  + Doporučeno nastavit Native VLAN jinou než VLAN 1
* Management VLAN — Vytvořený námi, slouží k správě aktivních prvků
  + Odříznutý od data VLAN — Uživatelé nomohou čmuchat náš management provoz

**Access port**

* Předpokládáme, že připojený uzel do tohoto portu je např. uživatelský počítač, tiskárna, atd.
* Po přijetí rámce od počítače se tzv. otaguje → Přidá se dot1q tag určující, z jaké VLAN rámec přišel
* Pomocí toho pak switch a další aktivní prvky mohou určit, do jaké VLAN rámec patří

**Trunk port**

* Předpokládáme, že připojený uzel do tohoto portu je např. switch či router
* Umožňuje přenos vícero rámců, které patří do jiných VLAN
  + Např. mezi Switchem A a Switchem B se udělají trunk porty, které mají povolené takové VLANy, které jsme nastavili na access portech

<aside> 💡 Trunk port můžeme přesně označit jako **IEEE 802.1q trunk port**

</aside>

**Tagování rámců**

* IEEE 802.1q
* 802.1q tag je vložen za zdrojovou MAC adresou v rámci

**Best practicies při realizace VLAN**

* Porty, které nevyužijeme přesuneme do jiné než výchozí VLAN 1
  + Také je samozřejmě vypneme, atd.
* Oddělení management provozu od uživatelského provozu
  + Vytvořením “management” VLAN → VLAN přes kterou budeme moct přistupovat k prvkům
* Vytvoření nativní VLAN, která bude jiná než 1

<aside> 💡 Nativní VLAN slouží k zpětné kompatibilitě s prvky, které VLAN nepodporují

</aside>

**Limitace na Cisco switchých 2960 a 3560 series**

* Podporují až 4096 VLAN
  + Normal range VLANs
    - VLAN 1 až VLAN **1005**
    - Uloženy do vlan.dat ve flash paměti switche
  + Extended Range VLANs
    - VLAN 1006 až VLNA **4096**
    - Uloženy do running-configu v NVRAM