

Address Resolution Protocol

- Konverze MAC (např. Ethernet) a síťových (např. IP) adres
- Neznámé adresy se zjišťují broadcastovou výzvou:

Ethernet=1	IP=0x0800		ARPreq=1
Sender MAC		Sender IP	
FF:FF:FF:FF:FF:FF		Target IP	

- Výsledky se ukládají na uzlu do ARP *cache*
- Unicastová odpověď (odpovídající si nejprve musí přidat informace o tazateli do svojí ARP tabulky)
- Nelze ověřit správnost odpovědi (RFC 826!)
- Gratuitous ARP: nevyžádané ARP (rychlejší změny, riziko)
- Výpis ARP tabulky: `arp -a`
- Omezení na linkový segment, mezi sítěmi je v činnosti OSI 3

Úvod do počítačových sítí (2020)
S/SAL 154

Address Resolution Protocol je pomocný technický protokol, který představuje spojovací článek mezi síťovou a linkovou vrstvou. Umožňuje uzlům v síti zjišťovat linkové (MAC) adresy odpovídající konkrétním síťovým adresám. Je to obecný protokol, může sloužit pro jakékoliv síťové a linkové adresy, my ale budeme demonstrovat jeho použití na Ethernetových a IP adresách.

Už víme, že linková vrstva dostane od síťové vrstvy požadavek doručit data po přímo připojeném segmentu sítě, a to buďto cílovému stroji nebo nejbližšímu routeru, tedy cílovému uzlu na úrovni linkové vrstvy. Potřebuje tedy správně vyplnit do svého záhlaví MAC adresu cíle odpovídající určité IP adrese. Aby tuto MAC adresu zjistil, vyšle cílovému uzlu rámec s ARP dotazem. Cílovou MAC adresu ovšem zatím nezná, a proto použije **broadcastovou** MAC adresu (FF:FF:FF:FF:FF:FF). Díky tomu se rámec doručí na všechny uzly na daném linkovém segmentu, ale všichni kromě držitele poptávané IP adresy ho ignorují.

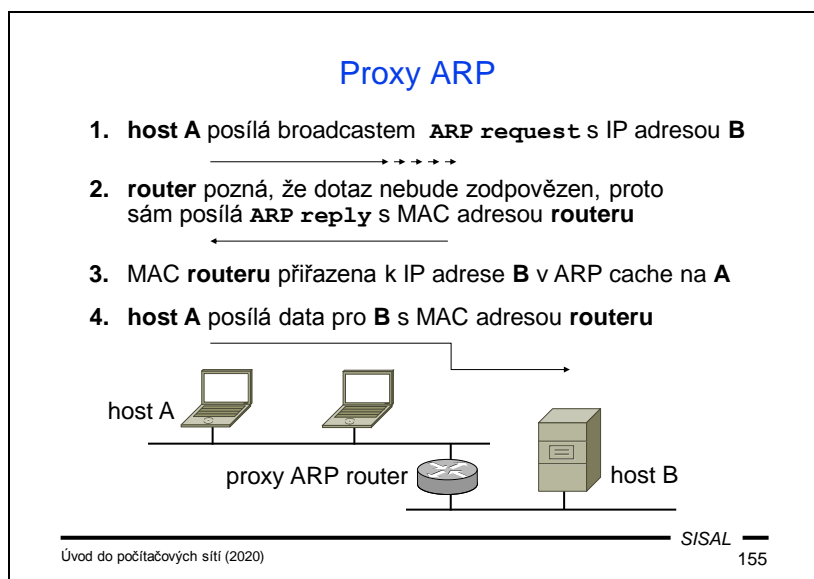
Hledaný uzel (tj. vlastně ARP server) na dotaz zareaguje unicastovou ARP odpovědí s požadovanou MAC adresou. Tazatel (ARP klient) si přiřazení IP a MAC adresy pro další použití uloží do **ARP cache**, ve které odpověď vydrží po konfigurovatelnou dobu (řádově minuty). Ovšem stejnou operaci udělá i server. Předpokládá se totiž, že právě zahájená komunikace bude pokračovat a server by musel záhy sám adresu klienta dohledávat. Proto si patřičné přiřazení klientovy IP a MAC adresy zanechá ve své cache. A proto také můžeme v naší ARP cache vidět i stroje, s nimiž jsme my sami nekomunikovali – stačí, aby nám poslaly ARP dotaz. Obsah ARP cache si můžeme vypsat příkazem **arp -a**.

ARP protokol (a tím i obsah ARP cache) je omezen rozsahem lokální linkové (OSI 2) sítě – pro komunikaci mimo linkový segment už potřebujeme vystoupat o vrstvu výš.

Mimochodem to také znamená, že ve stejné LAN lze používat karty se stejnou MAC adresou, pokud na sebe „nevidí“, tj. jsou odděleny routerem. Často to dokonce bývá tak, že router má sice pochopitelně do všech svých sítí různé IP adresy, ale jednu jedinou MAC adresu.

Problémem ARP je opět jeho chybějící zabezpečení a tedy to, že broadcastový dotaz dorazí všem, takže odpovědět nám může kterýkoliv uzel sítě. Ba co hůř, potenciální útočník ani nemusí čekat na náš dotaz. Existuje totiž také varianta **nevyžádané** ARP zprávy (gratuitous ARP), což je vlastně odpověď bez předchozího dotazu. Používá se např. pro **klastrová** řešení – důležité stroje v síti mohou běžet redundantně, přičemž oba sdílejí stejnou IP adresu, ale ten z dvojice, který je momentálně aktivní, informuje pomocí gratuitous ARP ostatní stroje v síti o svojí MAC adrese jako té, kterou mají v této chvíli používat pro sdílenou IP adresu.

Určitého zvýšení bezpečnosti v jistých typech sítí lze dosáhnout tak, že některým důležitým uzlům (serverům, routerům) **zakážeme** používat ARP protokol a nastavíme jim obsah ARP cache napevno v konfiguraci.



Ve složitých LAN se může správa sítě rozhodnout, že nebude stanicím v síti sdělovat detaily o jejich subnetu a správné směrování nechá čistě na směrovačích. Představme si zjednodušenou situaci jako na obrázku: v síti jsou dvě podsítě, ale stanicím se tato informace zatají a posílá se jim síťová maska, která odpovídá **celé** síti. Pokud počítač A chce komunikovat s počítačem B, podle síťové masky se domnívá, že jsou ve stejné síti, a pošle tedy ARP dotaz. Jak už ale víme, broadcastový dotaz se šíří pouze po linkovém segmentu, a tedy **podsíti**, kde je počítač A, takže k B nikdy nedorazí. Aby celá síť fungovala, musíme na routeru, který síť odděluje, spustit **ARP proxy**. To je služba, která ARP dotaz zachytí, a protože pozná, že by se tazatel nikdy nedočkal odpovědi, pošle mu odpověď místo stroje B a jako hledanou adresu uvede **svoji MAC** adresu. Klient si toto přiřazení uloží do ARP cache a pro další komunikaci s počítačem B bude používat MAC adresu routeru.

Může se to zdát divné, ale když si to pečlivěji rozmyslíme, klient se vlastně bude chovat **úplně stejně**, jako by měl správnou informaci o síťové masce a o routeru by věděl – posílal by pakety pro B rovněž s MAC adresou nejbližšího routeru!

Uživatel na stanici by mohl rozdíl pozorovat, kdyby se podíval do ARP cache. Našel by tam totiž **více IP** adres se **stejnou MAC** adresou. Pokud máme v síti ARP proxy, je to stav očekávaný. Ale pokud tomu tak není, může vícenásobný výskyt stejné MAC adresy v ARP cache signalizovat **pokus o napadení** naší ARP cache pomocí falešných ARP zpráv.

Linková vrstva (OSI 2)

- Dělí se na dvě podvrstvy:
 - Logical Link Control (LLC) umožňuje různým protokolům síťové vrstvy přístup ke stejnému médiumu (multiplexing)
 - Media Access Control (MAC) řídí adresaci uzlů a přístup k médiumu: kdo, kdy a jak může data odesílat a jak je přijímat
- TCP/IP už se touto vrstvou („síťového rozhraní“) nezabývá
- Síťový segment (fyzická síť):
 - množina uzlů sdílející stejné médium
- PDU na linkové vrstvě: rámec (frame)
 - liší se podle použitého média
 - obecně obsahuje: synchronizační pole, hlavičku (adresy, typ, příp. řídící data), datové pole a patičku (Frame Check Sequence - detekce chyb)

Protokol ARP pro nás představoval přesun od síťové vrstvy k **linkové** a tím také přesun mimo TCP/IP.

Linková vrstva představuje velmi důležitý mezník, dalo by se říci, že to je vlastně krok od software k hardware. Na rozdíl od některých jiných vrstev OSI modelu na ni zbylo velké množství práce a ve skutečnosti ji proto dělíme na dvě podvrstvy:

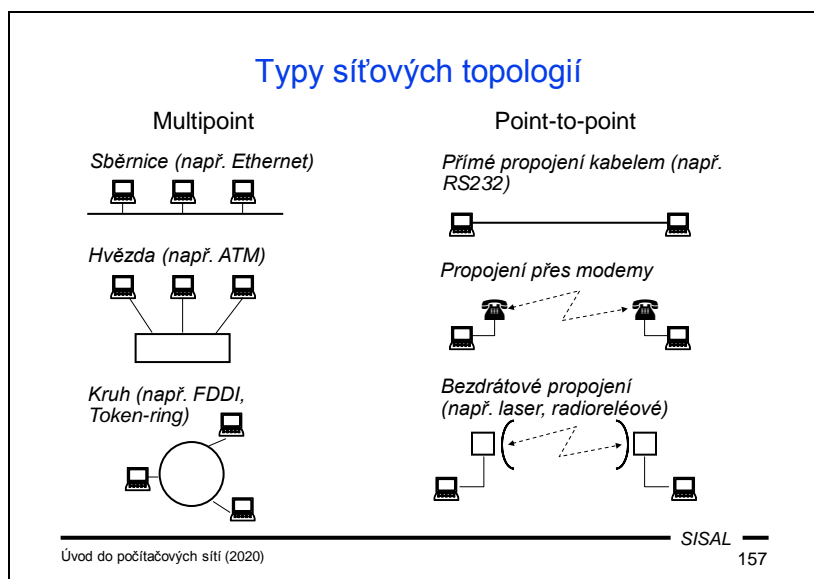
- Horní podvrstva se nazývá *Logical Link Control* (LLC) a má na starosti **multiplexing**. Zodpovídá tedy za to, že data různých síťových protokolů se správně uloží a identifikují tak, aby je přijímající linková vrstva dokázala předat software odpovídajícího síťového protokolu.
- Spodní podvrstva se nazývá *Media Access Control* (MAC) a má na starosti jednak **adresaci** a jednak **řízení přístupu** jednotlivých uzlů k fyzickému médium, které sdílejí v rámci stejného linkového (fyzického) segmentu sítě.

Podvrstva MAC je přímo závislá na technologii pod ní ležící fyzické vrstvy a my se budeme zabývat především dvěma z nich, Ethernetem (který dnes dominuje kabelovým sítím) a bezdrátovou technologií WiFi.

Datová jednotka linkového protokolu se nazývá **rámec** (frame) a jeho přesný formát se pro jednotlivé protokoly linkové vrstvy může lišit, obecně ale obvykle obsahuje:

- *Synchronizační pole* – speciální sekvenci bitů, která má „probudit“ cílovou stanici, odlišit následující data od „šumu“. Toto pole se obvykle nepočítá do skutečného obsahu rámce.
- *Hlavičku* – úvodní část rámce, která obsahuje přinejmenším MAC adresy příjemce a odesílatele a řídící informace LLC.
- *Data* (payload) nadřazeného protokolu.

- *Patičku* – závěrečnou část rámce, která obvykle obsahuje tzv. Frame Check Sequence (FCS), hodnotu, jež slouží ke **kontrole správnosti** doručení. Linková vrstva je totiž poslední vrstva pracující s daty nad fyzickou vrstvou, takže je žádoucí, aby se po doručení na cílový uzel zkontrolovalo, zda fyzická vrstva data přenesla v pořádku.



Jedním z kritérií, jak třídit různé technologie spodních dvou vrstev OSI jsou možnosti jejich topologie (uspořádání uzlů). Topologie může být studována buďto z pohledu **fyzického**, tj. jak jsou doopravdy uzly propojeny (např. kabelem), anebo **logického** (jak spolu uzly komunikují).

Technologie, které umožňují propojit v rámci linkového segmentu více uzlů (**multipoint**) obvykle používají některou z následujících topologií:

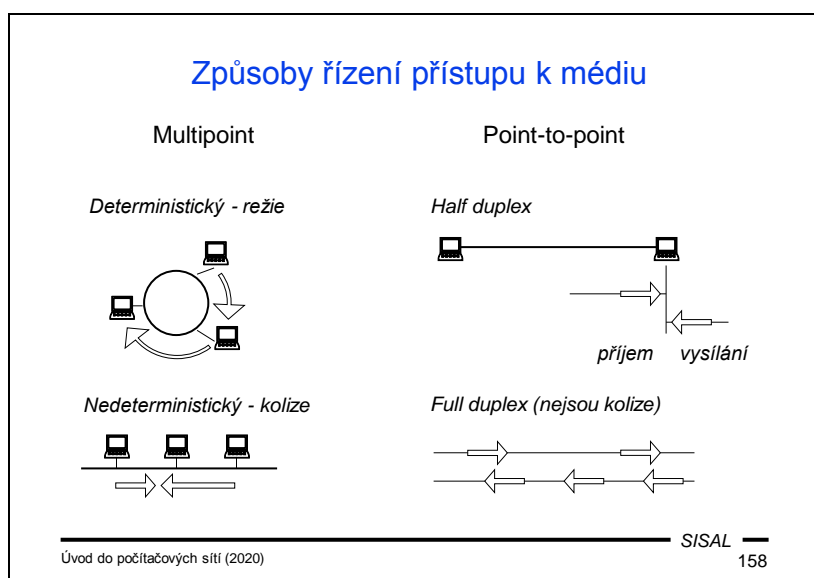
- **Sběrnice** – všechny uzly jsou připojeny sériově na stejné médium. Všechny mohou (víceméně) zároveň přijímat signál, který se šíří po médiu (což je v pořádku), ale všechny se mohou zároveň pokoušet nějakou svoji zprávu odeslat. Tomuto stavu se říká **kolize** a protokol ho musí nějak řešit. Fyzická sběrnicová topologie Ethernetu realizovaná koaxiálním kabelem byla ve své době velmi populární, měla velkou výhodu v tom, že přidání nového počítače do sítě bylo možné udělat prakticky kdykoliv a kdekoliv, ale její hlavní nevýhoda spočívala v tom, že pokud se kabel kdekoli přerušil, rozpadla se **celá** síť.
- **Hvězda** – segment obsahuje centrální prvek a jednotlivé další uzly jsou vázané na tento centrální prvek. Toto je nejobvyklejší fyzická topologie, kterou se v současnosti realizuje Ethernetová síť. Často se používá termín **strukturovaná kabeláž** a stanice se obvykle připojují UTP kabelem. Ovšem logicky zůstává topologie Ethernetu i nadále **sběrnicová**, sběrnice je ale koncentrována do zmíněného centrálního prvku, kde je dostatečně fyzicky chráněna, zatímco kabeláž náchylná k poškození propojuje pouze koncové stanice, takže v případě problému je nefunkční pouze dotčená stanice.
- **Kruh** – jednotlivé uzly jsou propojeny do kruhu. Tímto způsobem pracují např. technologie Token-ring a FDDI.

Alternativou k multipoint technologiím jsou technologie používající **point-to-point** topologii, např. RS 232. U nich se mohou propojit navzájem jen dva uzly, takže adresace na linkovém protokolu může být výrazně zjednodušená. Pokud ale budeme hovořit o fyzické point-to-point topologii, kdy spolu propojíme např. Ethernetovým kabelem pouze dva uzly, pořád se bude používat stejný protokol jako v multipoint zapojení.

Dosah point-to-point technologií je možné prodloužit tím, že se část linky nahradí jinou technologií. Možným řešením je zapojit k počítači zařízení (*modem*), které **moduluje** datový provoz tak, aby se dal přenést pomocí telefonního spojení.

Poslední skupinou technologií používající point-to-point model jsou bezdrátová propojení s velkým dosahem na bázi laseru nebo radiových vln. Nicméně to, co si dnes obvykle člověk představí pod pojmem „bezdrátová síť“, je jiná technologie, která pracuje s hvězdicovou topologií.

Poznámka: Centrálním prvkem hvězdy ve strukturované kabeláži bývá switch a jednotlivé stanice se k němu připojují kabelem zastrčeným do jedné zásuvky. Zásuvce se běžně říká **port**. Pozor – nezaměňujte si tento termín s termínem port na transportní vrstvě, jde o homonymum! Od této chvíle budeme v naší části přednášky používat toto slovo pouze v novém významu.



Topologie do značné míry určuje, jaké možnosti má daná technologie z hlediska řízení přístupu uzlů k médiu, tak aby se dokázala vypořádat s paralelními požadavky na přenos.

U multipoint topologií se používají dva základní přístupy:

- Při *deterministickém* způsobu řízení problémy nenastávají. Někdo nebo něco deterministicky určuje, kdo smí v dané chvíli vysílat. Nevýhodou je to, že pokud uzel, který „je na řadě“, zrovna nemá co vysílat, jeho vysílací frekvence zůstane nevyužita, což zvyšuje režii a snižuje kapacitu sítě.
 - V některých sítích (např. Token-ring) roli řídicího prvku hraje zvláštní paket postupující po síti (*token*) – pokud uzel chce vysílat, počká na token a pošle po síti místo něj svoje data; příjemce datový paket odstraní a pošle do sítě zase token.
 - V některých technologiích existuje v síti zvláštní řídicí uzel, který posílá ostatním uzlům signál v okamžiku, kdy mají právo vysílat.
- Při *nedeterministickém* řízení (např. u Ethernetu) nikdo neomezuje uzly ve vysílání, takže se následně musí řešit **kolize** (viz další slajd).

U point-to-point topologií rozlišujeme, zda uzel dokáže současně přijímat i vysílat.

- Pokud to uzel neumí, bude pracovat v tzv. *half-duplex* režimu. Pokud takto zapojíme Ethernet, budou na něm normálně nastávat kolize.
- Pokud oba uzly zvládnou vstup i výstup současně, je možné oběma nastavit *full-duplex* režim práce. Např. na Ethernetu se tím pro daný segment zbavíme kolizí, čímž radikálně zvýšíme propustnost.

Řešení kolizí

- CSMA (Carrier Sense with Multiple Access)
 - uzel poslouchá „nosnou“, a pokud není volno, čeká
- CSMA/CD (Collision Detection), např. Ethernet
 - během vysílání uzel současně detekuje případnou kolizi
 - při kolizi stanice zastaví vysílání, upozorní ostatní, počká určitou (náhodnou!) dobu a pokus opakuje, obvykle se postupně prodlužuje interval čekání (*exponenciální čekání*)
 - podmínka: doba vysílání rámce > doba šíření po segmentu (*kolizní okénko*); limituje max. délku segmentu a min. velikost rámce
- CSMA/CA (Collision Avoidance), např. WiFi
 - když je volná nosná, vysílá se celý rámec a čeká se na ACK
 - pokud není volná nosná nebo nedorazí ACK, zahájí se exponenciální čekání

Sítě, které umožňují vícenásobný přístup, musejí nějakým způsobem řešit, když více uzlů najednou hodlá začít vysílat. Základním principem je, že uzel zkontroluje „**nosnou**“, tedy přenosové médium, jestli na něm neprobíhá aktuálně nějaký přenos. Tomuto kroku se říká kontrola nosné („carrier sense“). Pokud nosná není volná, uzel čeká. Pokud je nosná volná, může začít vysílat. Je ovšem jasné, že takováto kontrola negarantuje, že nedojde k vícenásobnému přístupu („multiple access“). Proto existují různá rozšíření, která se pokoušejí vzniklé **kolize** nějak řešit.

Ethernet používá metodu *Collision Detect*. Uzel během vysílání zároveň kontroluje nosnou, takže je schopen **detekovat kolizi**. Pokud k ní dojde, uzel zastaví vysílání, upozorní ostatní stanice v síti na vzniklou kolizi a zahájí čekání na nový pokus. Doba čekání je nutné volit náhodně z určitého intervalu, aby se snížilo riziko, že oba uzly, které kolizi detekovaly, čekají stejně dlouho a opakované vysílání skončí kolizí opět. Zároveň je nutné opakovanými pokusy síť nezahltit, a proto se používá čekání **exponenciální**, tzn. že střední hodnota intervalu, ze kterého se vybírá doba čekání, se s každým dalším pokusem zdvojnásobuje. Podmínkou úspěšnosti této metody je ale to, že doba vysílání rámce musí být delší než doba šíření rámce z jednoho konce segmentu na druhý (tzv. *kolizní okénko*). Pokud tato podmínka není splněna, mohlo by se stát, že některý uzel stihne odvysílat celý rámec dříve, než z druhého konce segmentu dorazí kolidující rámec, uzel kolizi nedetekuje, nepokusí se o opakované odeslání a rámec bude pro většinu ostatních uzlů ztracen. Kolizní okénko určuje jednak maximální délku síťového segmentu (příliš dlouhé segmenty se musí rozdělit) a minimální velikost rámce (příliš krátké rámce se doplňují „vatou“).

WiFi používá metodu *Collision Avoidance*. Využívá přitom hvězdicové topologie – jako centrální prvek hvězdy funguje tzv. *access point* (AP), ke kterému jsou připojeny jednotlivé stanice. Jakýkoliv přenos je tedy v danou chvíli vlastně point-to-point

operací mezi stanicí a AP a lze tu implementovat potvrzování doručení. Uzel kolizi rozpozná tím, že nedorazí potvrzení o doručení, a zahájí exponenciální čekání.

Ethernet

- Historie:
 - první pokusy o realizaci LAN ve firmě Xerox
 - standardizaci převzalo IEEE (únor 1980 → IEEE 802)
 - dva nejběžnější formáty Ethernet II, IEEE 802.3
- Momentálně vůdčí technologie pro lokální sítě
 - dokáže pružně reagovat na progresivní vývoj HW
 - přizpůsobí se širokému spektru přenosových médií
- Řízení přístupu metodou CSMA/CD
 - při detekci kolize uzel vysílá „jam signal“
 - exponenciální čekání končí po 16 pokusech chybou
- Adresy:
 - 3 byty prefix (výrobce, multicast...), 3 byty adresa
 - dříve „vypálená“ v kartě, dnes nastavitelná

Ethernet vznikl ve firmě Xerox a je trochu překvapivé, že za nejúspěšnější síťovou technologií stojí firma na kopírky. V únoru 1980 převzala standardizaci protokolu organizace IEEE (Institute of Electrical and Electronics Engineers) a traduje se, že toto datum je ve skutečnosti skryto za číslem 802 v oficiálním označení standardu (IEEE 802). Tento krok znamenal rozštěpení vývoje, jehož důsledkem je mimo jiné odlišný formát dvou nejběžnějších verzí protokolu. V počáteční fázi vývoje měl Ethernet řadu soupeřů, určitý čas nad ním měl třeba převahu IBM Token-ring, ale během desítek let se ukázalo, že Ethernet měl nejlepší předpoklady držet krok s vývojem hardware.

K řešení kolizí na multipoint segmentech a half-duplexních point-to-point segmentech se používá metoda CSMA/CD tak, jak jsme ji popsali na minulém slajdu.

Ethernetové adresy mají délku 6 bajtů, z toho první tři tvoří prefix výrobce, druhé tři vlastní číslo karty. Adresa je výrobcem uložena do síťové karty, dříve napevno, v současnosti ji lze softwarově změnit. Pokud tak ale nikdo neučinil, dá se z adresy poznat výrobce. Kdysi dávno platilo pravidlo, že výrobci garantují jedinečnost adres, to už ale dnes neplatí. Může se klidně stát, že pokud koupíte dvě karty od stejného výrobce, budou mít stejnou adresu. Takové karty lze použít v jedné LAN současně jen tehdy, pokud je odděluje router. Jinak je třeba alespoň jedné z nich adresu změnit. Kromě prefixů výrobců existují ještě další zvláštní hodnoty prefixů pro broadcasty a multicasty.

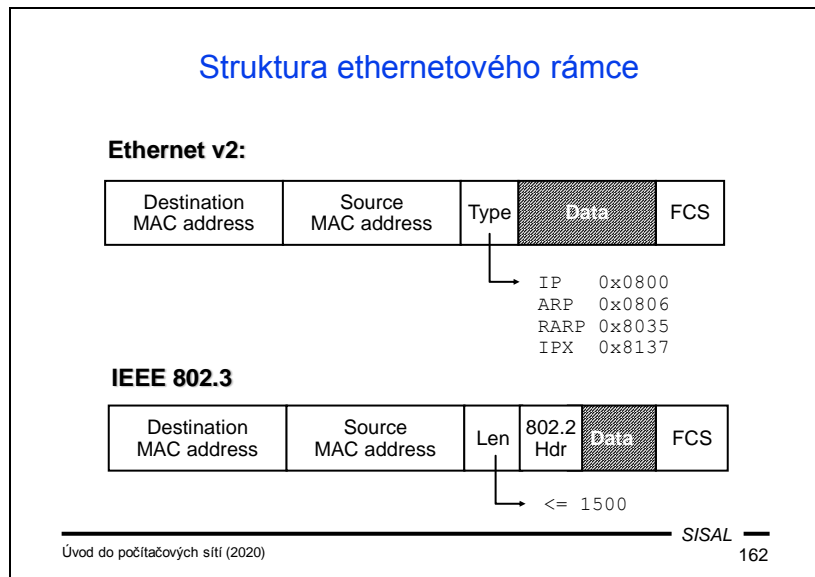
Standardy IEEE 802.3

Standard	Rok	Označení	Rychlost	Médium
802.3	1983	10BASE5	10 Mbit/s	tlustý koaxiální kabel
802.3a	1985	10BASE2	10 Mbit/s	tenký koaxiální kabel
802.3i	1990	10BASE-T	10 Mbit/s	kroucená dvoulinka (UTP)
802.3j	1993	10BASE-F	10 Mbit/s	optický kabel
802.3u	1995	100BASE-TX,FX	100 Mbit/s	UTP nebo optický kabel
802.3z	1998	1000BASE-X	1 Gbit/s	optický kabel
802.3ab	1999	1000BASE-T	1 Gbit/s	kroucená dvoulinka
802.3ae	2003	10GBASE-SR,...	10 Gbit/s	optický kabel
802.3an	2006	10GBASE-T	10 Gbit/s	kroucená dvoulinka
802.3ba	2010	100GBASE-SR	100 Gbit/s	optický kabel

Na rozdíl od RFC jsou normy IEEE vázány licencí.

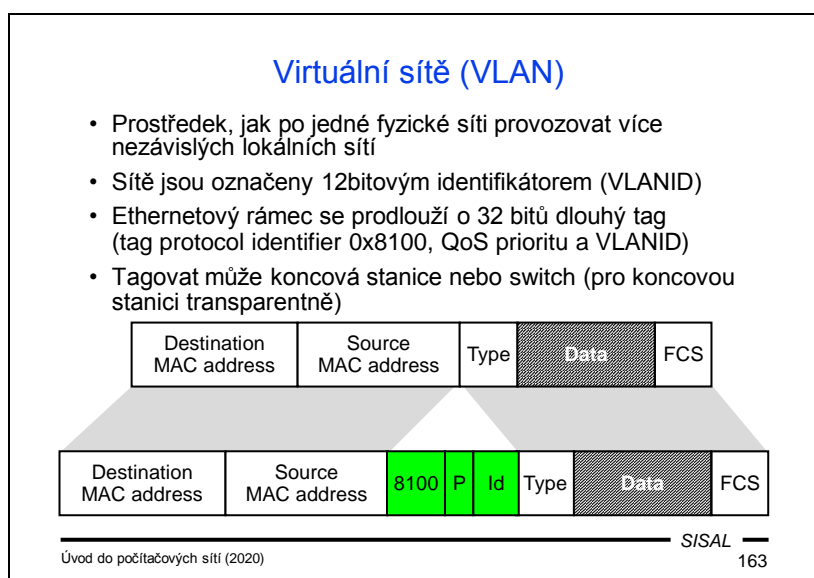
Tato tabulka ukazuje, jak se Ethernet postupně měnil s tím, jak se posouvaly možnosti hardware. Zkoušet se nebude.

Důležité upozornění ohledně otevřenosti protokolů: normy IEEE na rozdíl od RFC **nejsou veřejné**. Musíte si je koupit. Anebo půjčit od někoho, kdo si je koupil...



Dlouhý a bouřlivý vývoj Ethernetu znamenal také strukturu rámce. Zatímco v původním konceptu hlavička obsahuje za MAC adresami dvoubajtový typ síťového protokolu, ve verzi od IEEE je na místě typu délka a teprve za ní je speciální přidané LLC záhlaví (IEEE 802.2).

Povšimněte si také toho, že na rozdíl od hlavičky IP zde předchází adresa příjemce adrese odesílatele. Důvodem je to, aby hardware dokázal co nejrychleji poznat, zda přicházející rámec je určen danému uzlu nebo ne.



Struktura Ethernetového rámce umožnila zavedení důležitého nástroje pro budování sítí, a to **virtuálních sítí** (VLAN, Virtual LAN). Používá se v situaci, kdy jednu fyzickou síť je třeba rozdělit na více logických sítí (např. když ji sdílí více subjektů, anebo chceme oddělit segmenty s různým stupněm ochrany).

Podstatou metody je vsunutí 4B úseku (tzv. *VLAN tagu*) do rámce, za MAC adresy, čímž dojde ke změně **typu** rámce na speciální typ VLAN, a vložený úsek nese informaci o konkrétním čísle virtuální sítě (VLANID). Důležitou vlastností ale je to, že tato operace se může odehrát **transparentně**, bez vědomí koncové stanice. Ta může být připojena do portu switche, který je nakonfigurován jako součást sítě s určitým VLANID – v tom případě se z pohledu stanice bude síť tvářit jako obyčejná síť: stanice odesílá normální rámce, switch do rámce vsune VLAN tag se správným VLANID a pošle ho dál; naopak když na switch dorazí rámec pro tuto stanici, switch z něj VLAN tag odebere a stanice obdrží rámec, jako by přišel z obyčejné sítě bez VLAN. Stanice přitom nemá přístup k rámcům, které se přenášejí po síti, ale mají VLAN tag s jiným VLANID. Tím dojde k logickému oddělení provozu v různých virtuálních sítích. Nicméně v síti obvykle existují uzly, které potřebují mít přístup k rámcům ze všech virtuálních sítí (např. centrální router). Pro ně se pak přípojný bod konfiguruje jako tzv. **trunk** a switch v tom případě nedělá s rámcem žádné operace a veškerá obsluha VLAN tagů je ponechána až na koncovém uzlu.

Drobná komplikace spočívá v tom, že rámec se přidáním tagu prodlouží, takže všechna síťová zařízení, kterými otagované rámce procházejí, musí být schopna pracovat s rámcem delšími než je povolené maximum. Jinou možností je informovat všechny stanice v síti o tom, že maximální povolený rámec je o 4B kratší, než je standard.

Cyklický kontrolní součet (CRC)

- CRC (Cyclic Redundancy Check) je hashovací funkce široce používaná pro kontrolu konzistence dat (např. FCS)
- Posloupnost bitů je považována za koeficienty polynomu (ve dvojkové soustavě)

$$\boxed{\dots \ 1 \ 1 \ 0 \ \dots} \quad \Rightarrow \quad \dots + 1 \cdot x^{26} + 1 \cdot x^{27} + 0 \cdot x^{26} + \dots$$

- Ten se vydělí tzv. *charakteristickým polynomem* (např. pro CRC-16 je to $x^{16} + x^{15} + x^2 + 1$)
- Zbytek po dělení se převede zpět na bity a použije jako hash
- Jednoduchá implementace (i pomocí HW)
- Velká síla, n -bitový CRC detekuje:
 - na 100% chyby s lichým počtem bitů, chyby kratší než n bitů
 - s vysokou pravděpodobností i delší chyby

Už několikrát jsme se při popisu formátu dat setkali s poli obsahujícími nějakou kontrolu obsahu. Příkladem může být třeba kontrolní součet IP hlavičky, anebo Frame Check Sequence v patičce rámce. Většina těchto kontrolních mechanismů používá hashovací funkci **cyklického kontrolního součtu** (CRC, Cyclic Redundancy Check). Tato funkce je založena na dělení polynomů, což je docela zajímavé, protože se může zdát, že takový výpočet bude dost složitý, a přesto ho lze velmi snadno realizovat pomocí hardware.

Základní myšlenkou je to, že převedeme posloupnost bitů na polynom s binárními koeficienty, které odpovídají jednotlivým bitům, a s řádem, který odpovídá počtu bitů. Tento polynom poté vydělíme tzv. *charakteristickým polynomem*. Tento polynom musí mít takový stupeň, kolik bitů má kontrolní pole. Polynom, který vyjde jako zbytek po dělení, se opět převede na posloupnost bitů, a tím dostáváme výsledek funkce s pevnou velikostí odpovídající řádu charakteristického polynomu.

Navzdory jednoduché implementaci má metoda překvapivě velkou sílu. Dokáže detekovat všechny chyby s lichým počtem bitů, všechny chyby kratší než počet bitů kontrolního pole a i u chyb jiného rozsahu má poměrně vysokou úspěšnost.

WiFi

- Bezdrátová síť, jiný název: WLAN (wireless LAN)
- Mnoho různých variant pod souhrnným označením IEEE 802.11 (802.11a, b, g, n, y,...):
 - různá pásma (2,4 až 5 GHz)
 - různé rychlosti (2 až 600 Mbps)
- WiFi zařízení dnes prakticky v čemkoliv
- Struktura sítě:
 - ad-hoc peer-to-peer síť
 - infrastruktura přístupových bodů (access pointů)
- SSID (Service Set ID): řetězec (až 32 znaků) pro rozlišení sítí
- Problém: **zabezpečení!**

Termín WiFi označuje skupinu protokolů IEEE 802.11, jež se používají pro bezdrátovou komunikaci v bezlicenčních frekvenčních pásmech 2,4 a 5 GHz. Název původně neměl znamenat nic, ale časem se z něj stala slovní hříčka parodující Hi-Fi. Jiný používaný název je rovněž Wireless LAN (WLAN). Na rozdíl od protokolů IEEE 802.3 se ovšem jednotlivé protokoly této skupiny od sebe výrazně odlišují a tvoří souvislou řadu.

Společnou vlastností je použití CSMA/CA a rovněž hvězdicová topologie sítě. Ačkoliv je možné síť používat i v ad-hoc peer-to-peer variantě, obvykle se používá taková infrastruktura, kdy ve středu jednotlivých hvězd jsou *přístupové body* (AP, access pointy), k nimž se připojují v rámci jejich dosahu jednotlivá koncová zařízení. Plánování pokrytí nějakého prostoru nebo budovy WiFi signálem je složité, protože jednotlivé AP musejí buďto vysílat na nepřekrývajících se frekvenčních kanálech (kterých je jen 13), anebo musejí mít centrální řízení, které je poměrně drahé.

Problémem WiFi sítí je bezpečnost. Vzhledem k tomu, že potenciální útočník nepotřebuje fyzický přístup k síti (stačí stát před budovou), je třeba v privátních sítích maximálně dbát na všechny možné zabezpečovací prvky. Každá síť je identifikována speciálním identifikátorem SSID (Service Set Identifier), ten ale neslouží k zabezpečení, pouze k rozlišení různých sítí.

Fyzická vrstva (OSI 1)

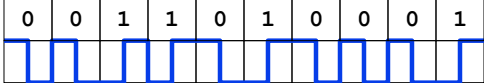
- Funkce vrstvy:
 - přenos dat po konkrétním fyzickém médiu
 - převod digitální informace na analogovou a obráceně
- Různé typy médií
 - metalické: elektrické pulzy
 - optické: světelné pulzy
 - bezdrátové: modulace vln

Poslední vrstvou OSI modelu je vrstva **fyzická**. Jejím úkolem je přenos fyzického signálu po konkrétním médiu. Pro něj je nutné nejprve převést digitální informaci (bity) na analogovou (elektrické pulzy na metalických kabelech, světelné pulzy na optických kabelech nebo různé modulace rádiových vln) a při příjmu je nutný opačný proces.

Druhy přenosu dat

- Analogový vs. digitální
 - ve skutečnosti je vše analogové (přenáší se např. proud)
 - digitální: rozhoduje, zda hodnota signálu spadá do nějakého intervalu (menší vliv zkreslení)
 - převody: D→A a zpět *modem* (modulator/demodulator), A→D *codec* (coder/decoder)
- Baseband vs. broadband
 - baseband přenáší přímo signál a kóduje ho, Ethernet používá tzv. Manchester:

0	0	1	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---


 - broadband přenáší základní signál a moduluje ho (fázi, amplitudu, frekvenci)

SISAL 167

Úvod do počítačových sítí (2020)

Vysvětlíme si nejprve některé pojmy týkající se variant přenosu, které mohou být poněkud matoucí.

Analogový vs. digitální přenos.

- Ve skutečnosti je každý přenos analogový, protože svět kolem nás je analogový. Ať už je použito jakékoliv médium, jeho fyzikální podstata je analogová. Co tedy znamená **digitální přenos**? V podstatě jde pouze o to, že přijímací zařízení se nesnaží interpretovat libovolnou příchozí hodnotu signálu, ale pouze hodnoty, které leží v určitém rozsahu. Některý rozsah je interpretován jako hodnota 1, jiný jako hodnota 0, a signál mimo tyto intervaly je ignorován. Vysílající zařízení se naopak snaží minimalizovat čas, kdy je signál mimo patřičné intervaly. Důsledkem těchto vlastností je fakt, že digitální přenos je více odolný vůči rušení.

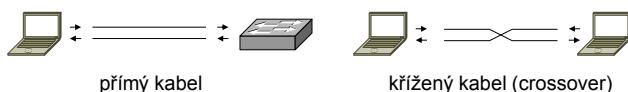
Baseband vs. Broadband přenos.

- Název **baseband** napovídá, že v tomto případě pro zakódování digitální informace používá přímo hodnota signálu. Příkladem může být šíření elektrického signálu po metalickém vedení. Laicky bychom asi mohli očekávat, že nejsnazší by bylo prostě jedničku kódovat jako vysoký stav signálu a nulu jako nízký. Při tomto způsobu by ale mohlo dojít k tomu, že hodiny na straně odesílatele by se začaly rozcházet s hodinami na straně příjemce a odesílané a přijímané bity by si přestaly odpovídat. Je tedy třeba přenášet ještě i „tikot hodin“, a proto třeba Ethernet používá kódování „Manchester“, kdy v každém „tiku“ se musí změnit hodnota signálu, přičemž nula je změna stavu z vysokého na nízký a jednička obráceně.
- Název **broadband** se nedá zcela přesně sémanticky ztotožnit s jevem, který označuje. Název vychází z toho, že se používá pro přenosy v širokém pásmu. Ale podstata tkví v tom, že se data se kódují pomocí určité **modulace** základníhoho

signálu. Používá se buďto posun fáze, změna amplitudy (známe z rozhlasu jako AM) nebo frekvence (FM).

Nestíněná kroucená dvoulinka (UTP)

- Dnes standardní prostředek strukturované kabeláže
- 4 páry Cu vodičů navzájem pravidelně zakroucené
 - zakroucení snižuje vyzařování i příjem elektromagnetického záření (nižší rušení)
- 100Mb Ethernet používá jen dva páry (je možno rozdělit)
- Konektory: RJ 45
- Při propojení je třeba zohlednit povahu zařízení
 - dnes obvykle už autodetekce MDI/MDIX



- Alternativa: kabel s kovovým stíněním (STP)

Pro připojení stanic v lokální síti se dnes používá většinou nestíněná kroucená dvoulinka (UTP, Unshielded twisted pair). Název je opět poněkud matoucí, protože v kabelu je dohromady **osm** vodičů. Podstata názvu spočívá ve **způsobu ochrany** před rušením. Je založen na tom, že pokud jsou dva vodiče navzájem zakroucené s pravidelným stoupáním, vytváří se při průchodu elektrického proudu elektromagnetické pole, které je přirozenou obranou proti běžné úrovni rušení. Navíc se do obou vodičů vysílá stejný signál, ale s opačnou polaritou, takže případné rušení lze na straně příjemce „odečíst“. Pokud ovšem potřebujeme kabeláž provozovat v prostředí se silným rušením, používá se kabel **STP** (Shielded Twisted Pair), který má kolem vodičů ještě dodatečné kovové **stínění**.

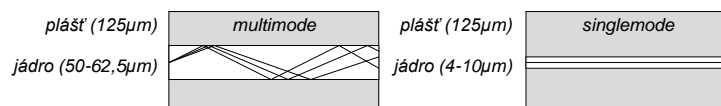
Poněkud zarážející zajímavostí je, proč je v kabelu osm vodičů, když Ethernet až po 100Mbps používá pouze čtyři. Toho se dá s výhodou využít a za pomoci speciálních rozdvojek propojit jedním kabelem **dvě** dvojice počítačů, kdy každé spojení používá různé čtyři vodiče.

Při osazování konektorů RJ 45 na UTP kabel je třeba respektovat to, že výstupní pin na jedné straně musí být propojen na vstupní pin na straně druhé. Aby bylo možné použít kabely, kde stejný vodič je na jedné straně vstupní a na druhé výstupní (tzv. **přímé**), používají koncové stanice (počítače) jiné zapojení pinů než switche. To ale zároveň může působit problémy, když potřebujeme propojit dvě **stejná** zařízení (dva počítače nebo dva switche). Pro takové propojení bychom měli použít kabel, kde jsou odpovídající vodiče zapojeny obráceně, tj. tzv. **křížený** kabel (crossover). Naštěstí jsou dnes síťové karty už schopné při připojení kabelu se s protistranou domluvit, který vodič bude která strana používat pro vstup a který pro výstup, takže v běžném případě, pokud obě strany tuto (tzv. MDI/MDIX) autodetekci podporují, není třeba typ použitého kabelu zkoumat. Autodetekce ovšem určitou dobu trvá, takže se tím

prodlužuje doba nutná k zahájení komunikace, což může pro některé kritické real-time aplikace být nepřijatelné.

Optická vlákna

- Signál se šíří jako viditelné světlo vláknem z SiO_2
 - vysoké frekvence, velká šířka přenosového pásma
 - nízký útlum, žádné rušení
- Nevýhody:
 - vyšší cena, náročnější manipulace, **nekoukat do kabelu**
- Druhy vláken:
 - jednovidová (singlemode): svítí se laserem => jeden paprsek, větší dosah, šířka pásma („rychlost“, ne rychlost), cena
 - mnohovidová (multimode), svítí se i LED



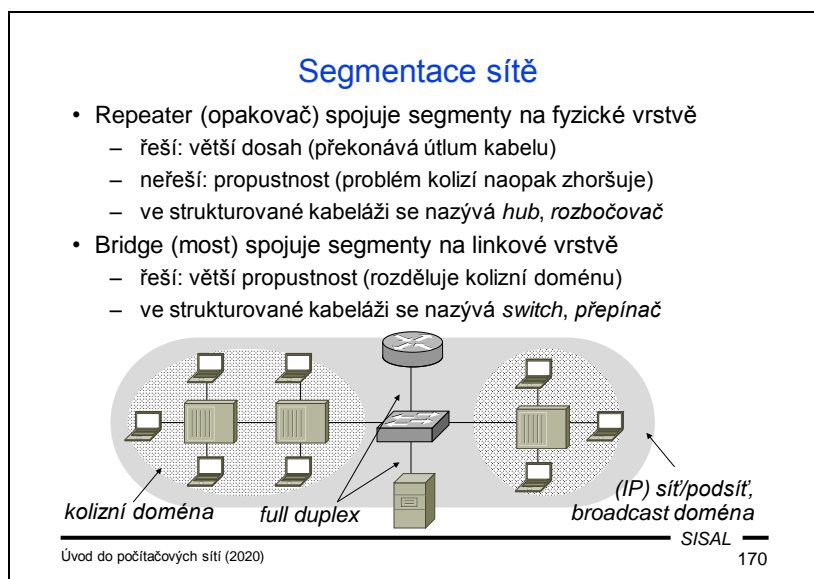
Úvod do počítačových sítí (2020)

S/SAL 169

Tam, kde narážíme na fyzikální omezení šíření elektrického signálu po metalickém kabelu, používá se kabeláž optická. Informaci šíří světelné pulzy, které se vysílají křemíkovým vláknem. Světelný paprsek nemá problém s rušením, má velmi nízký útlum a velkou šířku přenosového pásma („rychlost“). Nevýhodou je vyšší cena a náročnější manipulace (optický kabel má řádově větší **minimální poloměr ohybu**).

Rozeznáváme dva druhy optických kabelů:

- *Jednovidová (singlemode)* vlákna mají křemíkové jádro užší a jako světelný zdroj se používá laser. Díky tomu je výrazně omezen lom světelných paprsků, čímž se zásadně zvětšuje dosah i přenosová kapacita. A rovněž cena. Proto se tato vlákna obvykle používají pro přenosy na velké vzdálenosti.
- U *mnohovidových (multimode)* vláken je křemíkové jádro širší a jako světelný zdroj se mohou používat i LED diody. Paprsky se zde více lámou, takže dosah i přenosová kapacita jsou menší a používají se nejčastěji pro páteřní rozvody LAN.



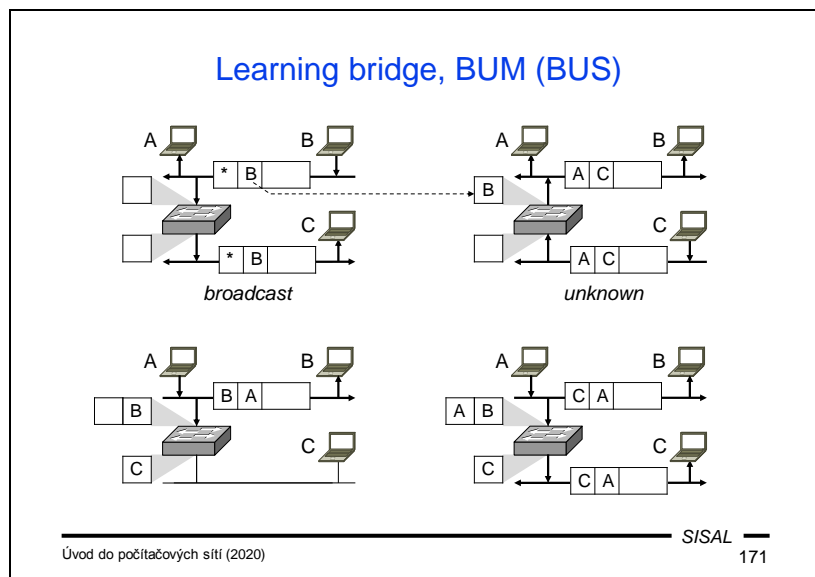
Podívejme se nyní na příklad propojení malé lokální sítě, na němž si vysvětlíme některé další pojmy. Z důvodů většího dosahu i větší propustnosti sítě do ní budeme potřebovat zapojit několik síťových zařízení.

Některé stanice jsou daleko od centra a pro jejich připojení můžeme do sítě přidat zařízení typu *repeater* (český termín „opakovač“ se skoro nepoužívá). Ve strukturované kabeláži se zařízení obvykle označuje „hub“ s českým ekvivalentem „rozbočovač“. Repeater pracuje na **fyzické vrstvě**, takže jeho úkolem je jen distribuovat patřičný fyzický signál, aniž mu jakkoliv rozumí. Řeší se tím zvětšení **dosahu** signálu, protože se eliminuje útlum. Nijak se tím ale neřeší otázka propustnosti sítě resp. počtu kolizí. Signál se od zdroje musí rozšířit všem stanicím, takže pokud by došlo ke kolizi bez repeateru, dojde k ní i s ním. Repeater nerozděluje tzv. *kolizní doménu*. Naopak pravděpodobnost kolizí se s repeaterem ještě zvětšuje, protože repeateru nějakou dobu trvá, než rámec přenese ze vstupního na výstupní rozhraní, čímž se doba šíření rámce po segmentu prodlouží.

Pokud chceme řešit i otázku **propustnosti**, musíme se posunout výš, na **linkovou vrstvu** a použít *bridge* (český termín „most“ se rovněž moc nepoužívá). Ve strukturované kabeláži se obvykle používá termín „switch“ (přepínač), který už jsme poznali dříve. Bridge už rozumí MAC adresám v přenášených rámcích (ačkoliv jeho vlastní MAC adresy zde nijak nefigurují) a dokáže rámce posílat pouze tam, kam je to nutné. Každý port bridge tak odděluje jednu kolizní doménu, čímž se počet kolizí rapidně sníží. Zároveň se tím řeší propustnost a do jisté míry i bezpečnost, protože do každého segmentu směřuje pouze provoz, který tam patří podle cílové MAC adresy. Samozřejmě taková ochrana není postačující, protože potenciální útočník může zfalšovat ARP protokol a přesvědčit naši stanici, aby své rámce posílala jemu místo skutečného příjemce.

V centrálním switchi jsou v naší síti na obrázku připojeny ještě dva uzly s výjimečným postavením – je to centrální server sítě a centrální router. Oba dva budou připojeny nejspíše samostatně, a to linkou v režimu **full duplex**, který jim zabezpečí maximální propustnost.

Celá síť oddělená jedním routerem představuje jednu IP síť (příp. podsíť) a také jednu *broadcast doménu* (oblast, po níž se šíří limited broadcasty).



Přepínače obvykle pracují v režimu **learning bridge**. Znamená to, že si pro každý port udržují tabulky MAC adres stanic, které jsou za tímto portem připojené, a udržují si je samy monitorováním provozu.

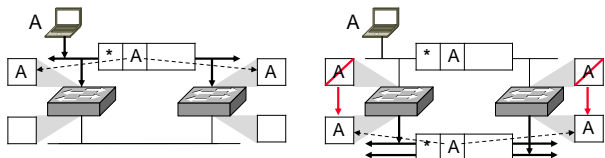
Na obrázku vidíme switch, který čerstvě nabootoval a má tabulky prázdné.

- Řekněme, že jako první se objeví broadcast rámeček poslaný stanicí B z horního segmentu. Vzhledem k tomu, že se jedná o **broadcast**, switch ho musí rozeslat na všechny porty. Zároveň se podívá na MAC adresu odesílatele a do tabulky MAC adres horního segmentu si zaneše adresu stanice B.
- Druhý rámeček bude poslaný stanicí C z dolního segmentu stanicí A na horním segmentu. Tuto informaci ovšem v této chvíli switch ještě nemá, umístění stanice A je **neznámé**. Proto i tento rámeček musí poslat na všechny porty. Současně se do tabulky pro dolní port dostane MAC adresa stanice C.
- Pokud nyní stanice A z horního segmentu pošle rámeček směřující ke stanici B, switch už dokáže zjistit, že stanice B leží na stejném segmentu sítě (resp. portu switchu), a proto není třeba rámeček už **nikam posílat**! Do tabulky MAC adres přibude poslední záznam o stanici A a od této chvíle už switch má celou mapu sítě přiřazenou ke svým portům.
- Proto když následně stanice A pošle rámeček stanici C, switch ho **přepośle pouze** na port, kam patří. Žádný jiný port resp. segment sítě už tento provoz nebude zatěžovat.

Od kroku 3 bude switch posílat do všechny rámce do správných portů, s výjimkou broadcastů, neznámých unicastů a multicastů, které bude nadále posílat všem. Proto se tomuto chování někdy říká **BUS** (broadcast and unknown service) nebo **BUM** (broadcast, unknown and multicast).

Spanning Tree Algorithmus

- Pokud by dva segmenty propojoval druhý switch, síť se zahltí přeposíláním rámců a learning bridge selže



- Důvod: graf je cyklický
- Řešení: najít acyklickou podmnožinu, kostru (spanning tree)
- Switche se musejí dohodnout, který z nich bude mít potlačeno forwardování a bude pouze monitorovat provoz
- Protokol (STP) má nezbytné timeouty, start portů je pomalý
 - obvykle lze STA na portu potlačit („faststart“), nutno zvážit

S/SAL

Úvod do počítačových sítí (2020) 172

Důležité body resp. propojení v síti se někdy kvůli robustnosti zálohují pomocí redundantních switchů. To ovšem přináší jeden zásadní problém. Pokud by pracovaly oba switche, síť bude zaplavena přeposíláním rámců a metoda learning bridge selže.

Podívejme se opět na příklad na obrázku. Řekněme, že stanice A vyšle broadcast. Když dorazí na switche, oba si přiřadí MAC adresu A ke správnému portu a **oba** rámec odešlou do druhého segmentu. Poznamenejme, že switche na rozdíl od routerů do rámce **nijak nezasahují**. Proto oba rámce po dolním segmentu dorazí ke druhému switchi. Na to ovšem oba switche zareagují tak, že provedou změnu přiřazení MAC adresy. Tím stanici A „přesunou“ na dolní segment a učiní ji na čas nedostupnou. Zároveň ale rámec opět poslušně přepošlou do horního segmentu a tím uzavřou nekonečnou smyčku.

Pokud si síť představíme jako graf, kde jednotlivé segmenty jsou vrcholy a switche hrany, je příčinou problému **kružnice**, kterou jsme přidáním druhého switche vytvořili. Problém, jak graf zredukovat a zbavit se kružnice, je známý, je to problém **nalezení kostry grafu** (kostra = spanning tree). V počítačových sítích se používá distribuovaná verze algoritmu za pomoci protokolu STP (Spanning Tree Protocol). Vypuštění hrany je v praxi realizováno jako převedení (některých portů) switche do režimu **blocking**, kdy se nepřeposílají rámce a switch pouze monitoruje, zda nedošlo k výpadku druhého switche, který je v režimu **forwarding**.

STP ovšem pro svoji práci potřebuje určitý čas. Pokud zapojíme stanici do běžného portu switche, musí nejprve proběhnout STP a stanice je ještě několik vteřin nedostupná. To může někdy vadit a v takovém případě lze obvykle určité porty switche nakonfigurovat jako „faststart“ a použití protokolu potlačit. Tím administrátor **přebírá zodpovědnost** za to, že dalšími propojeními v síti nevytvoří cyklus.

Souhrn 8

- Popište účel a bezpečnostní rizika protokolu ARP.
- Popište účel vrstev LLC a MAC.
- Jaký je rozdíl mezi fyzickou a logickou topologií sítě?
- Co je kolize, kde se vyskytuje a jak se řeší?
- K čemu slouží a jak se realizuje VLAN?
- K čemu slouží CRC a jak pracuje?
- Popište způsob přenosu dat v závislosti na médiu.
- Popište typy metalických a optických kabelů.
- Co je learning bridge a STP?