

Měření latence síťových prvků

Michal Krumnikl

Obsah

1	Úvod	2
1.1	Přepínání	2
1.2	Úkol mostu a switche	2
1.3	Mosty vs. switche	3
1.4	Mechanismus fungování switche	3
1.5	Store-and-forward vs. cut-through	4
1.6	Fragment Free	5
2	Konfigurace switche Cisco Catalyst 1900	5
3	Měření latence síťových prvků	5
4	Naměřené hodnoty	10

1 Úvod

Text této kapitoly byl převzat z archivu článků a přednášek Jiřího Peterky a upraven pro potřeby této práce. Originální text je možno najít na adrese <http://www.earchiv.cz/>.

1.1 Přepínání

Switching je obecně řečeno variantou přepojování paketů na úrovni spojové vrstvy (data link), tedy jednou z forem přepojování rámců. To, co tuto variantu charakterizuje, je samozřejmě konkrétní mechanismus fungování - pro správné pochopení switchingu je ale vhodnější vyjít spíše z motivace, která ke vzniku této techniky vedla. Jelikož ale příslušná motivace spadá spíše do oblasti vzájemného propojování sítí, vezmeme to jen stručně, srovnáme posílání mostu (jako „klasického“ zařízení zajišťujícího přepojování rámců na úrovni linkové vrstvy), a tzv. switch-e (který také zajišťuje přepojování rámců na úrovni linkové vrstvy, ale na bázi switchingu).

1.2 Úkol mostu a switche

Hlavním úkolem mostu je propojovat mezi sebou dílčí části jedné a téže lokální sítě, nejčastěji jednotlivé kabelové segmenty (souvislé úseky kabelů). Typicky jde o takové sítě, které jsou budovány na koaxiálním kabelu, mají tudíž fyzicky sběrníkovou topologii, a důvody pro jejich rozdělení na více segmentů jsou spíše technické (například maximální možná délka souvislého kabelu). V praxi pak bývá takovýchto dílčích segmentů v rámci jedné sítě několik málo (například dva, tři, ...), a most má za úkol propojovat tyto segmenty takovým způsobem, aby byl možný přenos dat mezi nimi (k tomu se využívá jeho schopnost předávat data z jednoho segmentu do druhého, kterou označujeme jako tzv. forwarding). Kromě toho se ale most snaží chovat i dostatečně inteligentně a nedělat některé zbytečné věci - pokud například zjistí, že určitý rámec je adresován příjemci ve stejném segmentu, ve kterém se nachází i jeho odesílatel, domyslí si, že příslušný rámec nemá smysl šířit i do dalších segmentů, a tak to nedělá. Výsledkem je pak to, že většina „provozu“ se odehrává lokálně v jednotlivých segmentech sítě, a vzájemná komunikace mezi segmenty (a jejich vzájemné ovlivňování) je minimalizována na nejnížší nutnou míru.

1.3 Mosty vs. switche

Naproti tomu vznik tzv. switchů souvisí se snahou optimálně využít vlastností a možností nových druhů kabeláže, které již nemají sběrníkovou topologii - zejména kabeláže na bázi kroucené dvoulinky, ve které každý jednotlivý uzel může mít jen a jen pro sebe svou vlastní přípojku k nejbližšímu přepojovacímu uzlu. Představme si nyní takovýto stav, dovedený až do dokonalosti (například v prostředí klasického stomegabitového Ethernetu) - tedy stav kdy každý jednotlivý uzel má svou vlastní přípojku k nejbližšímu switchi, neboli co jeden segment, to jeden uzlový počítač. Pak ovšem nemůže existovat žádný „lokální“ provoz, který by mohl zůstat lokální v určitém segmentu a nemusel být propouštěn do dalších segmentů - veškerý provoz mezi kterýmikoli dvěma uzly naopak musí procházet přes zmíněný switch. Na druhé straně ale tento provoz nemusí být propouštěn do dalších „jednouzlových“ segmentů. Každé dva komunikující uzly pak v důsledku toho mají k dispozici plnou přenosovou kapacitu Ethernetu (který v tomto příkladu předpokládáme), tedy „100 megabitů full-duplex“, a nemusí se o ni dělit s jinými dvojicemi komunikujících uzlů (jako by tomu bylo v případě sdílených segmentů s více uzly). A právě to je hlavní cíl techniky, označované jako switching.

V praxi ovšem nemusí nastávat výše popsáný ideální stav, kdy každý jednotlivý uzel má svou vlastní přípojku, kterou nemusí s nikým sdílet. Místo toho mohou switche propojovat i takové segmenty, které jsou „zabydleny“ více počítači (tj. jsou jimi sdíleny), a mají tedy určitou míru čistě lokálního provozu. Ovšem čím více „zabydlený“ každý segment bude, tím menší budou výhody přinášené technikou switchingu, a tím více se bude chování switche blížit chování klasického mostu.

1.4 Mechanismus fungování switche

Můžeme říci, že tzv. switching, neboli princip fungování switch-e, není principiálně odlišný od způsobu fungování běžného mostu. Rozdíl zde není v kvalitě, jako spíše v kvantitě - typické switche mají více rozhraní (portů) pro připojení jednotlivých segmentů, a vyšší (rychlejší) bývají i jejich přepojovací schopnosti. Má-li totiž switch sloužit zejména tomu, aby komunikující uzly měly k dispozici maximální možnou přenosovou kapacitu, bylo by nevhodné tuto jejich komunikaci zpomalovat či jinak zdržovat.

Přepojování na úrovni linkové vrstvy (které provádí switch) je relativně jednodušší (oproti směrování), a může tudíž být i dostatečně rychlé. Zde se přitom úspěšně využívá možnosti realizovat celé toto přepojování na úrovni linkové vrstvy přímo technickými prostředky, hardwarem optimalizovaným pro takovýto účel (zatímco směrování se ještě obvykle řeší programovými

prostředky). Dnes se proto mohou běžně vyrábět a prodávat switche, které mají opravdu velmi vysoké přepojovací schopnosti a slibují výrazné zvýšení celkové propustnosti sítě (při vcelku únosné ceně). Ovšem jejich skutečný přínos je vždy ovlivněn dalšími faktory - zejména tím, kolik uzlů je v jednotlivých segmentech, a jaký je charakter jejich vzájemné komunikace. Někdy může nasazení switche opravdu významně zrychlit komunikace v dané síti, zatímco v jiné síti může být celkový přínos třeba i nulový.

1.5 Store-and-forward vs. cut-through

Principu store-and-forward je používán u síťových mostů - neboli každý jednotlivý blok dat je v každém síťovém uzlu (realizovaným mostem) nejprve celý načten a teprve pak zpracován. Most zkontroluje CRC kód celého rámce a pokud souhlasí, odešle jej na odpovídající port dle přepínací tabulky. Některé síťové mosty jsou schopny zpracovávat jen rámce delší 64 bytů a kratší 1518 bytů. Tomuto kritériu vyhovuje standardní provoz na přepínané síti, některé mosty umožňují přeposílat i rámce větších velikostí. V případě switchů, kterým dost záleží na rychlosti, nemusí být tato metoda načítání celých rámců zrovna nejvhodnější. Proto některé switche používají ke svému fungování poněkud jiný princip, kterému se říká cut-through (zatímco jiné zůstávají u původního principu store-and-forward).

Princip cut-through spočívá v tom, že přepojovací uzel (switch) nečeká na načtení celého přenášeného bloku dat, a snaží se jej zpracovat co možná nejrychleji. V konkrétním případě (předpokládejme Ethernetové rámce) si počká jen na načtení hlavičky Ethernetového rámce, ze které již dokáže poznat, co má s celým blokem dat udělat a okamžitě také své rozhodnutí naplní. To pak v praxi znamená, že příslušný rámec začne průběžně odesílat v příslušném výstupním směru ještě v době, kdy jej z jiného směru sám teprve přijímá. Odsud také zřejmě pochází označení celého principu - anglické „through“ znamená „skrz“, a celý switch se pro přenášený rámec v jistém smyslu stává „průchozí skrz“ (i když s jistým malým zpožděním, odpovídajícím době nutné pro načtení hlavičky rámce).

Hlavní výhodou principu cut-through je garance, že rámec projde přestupním uzlem (switchem) za pevně danou dobu (která je navíc hodně malá), zatímco v případě principu store-and-forward nic takového zaručeno není. Princip cut-through je proto výhodnější pro přenos různých multimediálních formátů a obecně pro přenos dat v reálném čase, kde časové závislosti jsou velmi důležité. Mezi nevýhody naopak patří skutečnost, že switch fungujícím na principu cut-through musí přenášet i poškozené rámce (který by přenášet nemusel). Důvodem je to, že poškození lze poznat až ze zabezpečovacího údaje, který se typicky nachází až na konci datového rámce - ale

switch fungující na principu cut-through se musí rozhodnout a začít přenášet rámeček ještě dříve, než sám tento zabezpečovací údaj přijme.

1.6 Fragment Free

Principem fragment-free je kombinace obou výše jmenovaných postupů. Vychází z předpokladu, že veškeré kolize mohou vzniknout jen během prvních $51.2\mu s$ (platí pro 10Mbit síť), což odpovídá prvním 512 bitům rámce. Toto pozorování vyplývá z norem pro strukturovanou kabeláž (CSMA/CD rozděljuje čas do slotů, pro nejdelší možnou délku (2,5 km + 4 repeatery) má slot délku $51.2\mu s$). Switch v tomto režimu nejdříve přijme 64 bytů rámce, zkontroluje zda nedošlo ke kolizi a teprve poté jej začne přeposílat na odpovídající port, tak jak tomu bylo v případě režimu cut-through.

2 Konfigurace switche Cisco Catalyst 1900

Switch Cisco Catalyst 1900 umožňuje nastavit režim přepínání paketů. Použitím globálního konfiguračního příkazu `switching-mode` můžeme vybrat, který z režimů přepínání bude použit. Switch podporuje tyto režimy: store-and-forward a fragment-free. Výchozím režimem pro přepínání je fragment-free.

Syntaxe příkazu je následující:

```
switching-mode { store-and-forward | fragment-free }  
    nastavuje zvolený režim  
no switching-mode  
    nastaví výchozí hodnotu
```

Switche Catalyst 29xx nemají možnost měnit režim přepínání. V dokumentaci¹ Cisco uvádí, že tyto switche mohou přepínat jen metodou store-and-forward.

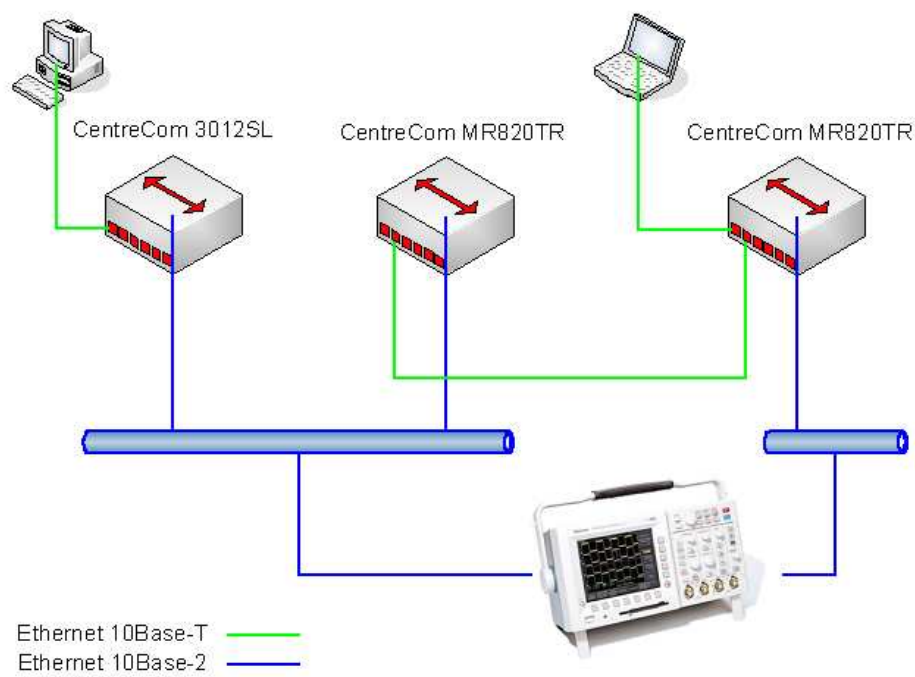
3 Měření latence síťových prvků

Měření zpoždění síťových prvků bylo realizováno pomocí dvoukanálového digitálního osciloskopu, kde jednotlivé kanály byly připojeny ke koaxiálním

¹Catalyst 2900 Series XL Installation and Configuration Guide, March 1998 : A Catalyst 2900 switch always forwards packets by using the store-and-forward mode: complete packets are stored and checked for errors before transmission.

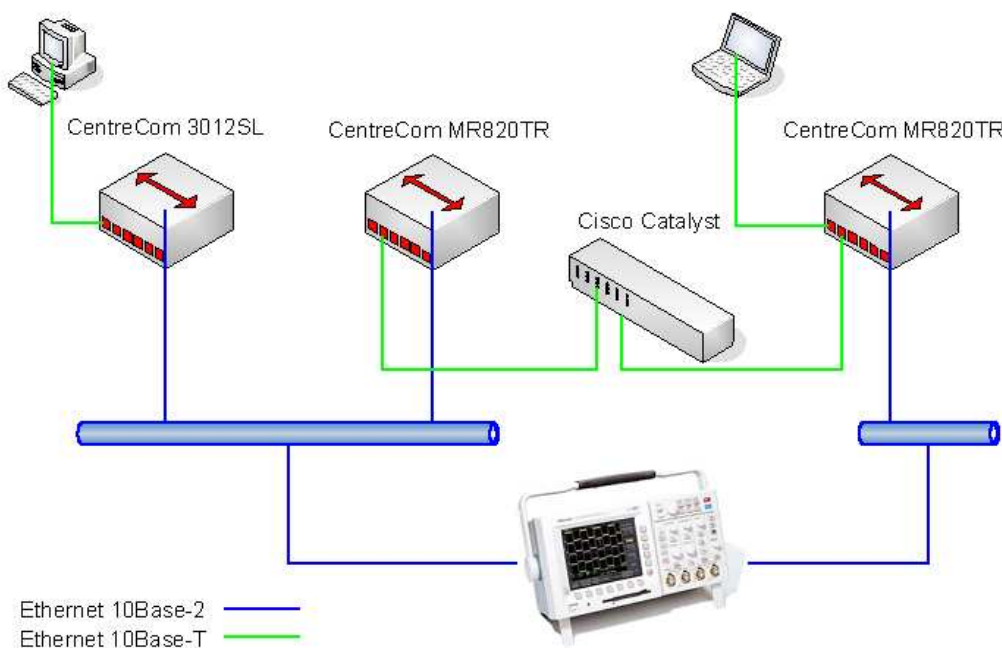
kabelům kabeláže 10Base-2. Zdrojová stanice vysílající pakety je připojena pomocí twisted pair kabeláže do HUBu, který funguje jako převodník 10Base-T na 10Base-2. Zpoždění toho HUBu se v následném měření nepromítne, protože první kanál osciloskopu je připojen až na segment 10Base-2 vedoucí z tohoto HUBu. Segment 10Base-2 s připojeným osciloskopem je připojen do druhého HUBu, který funguje jako převodník zpět na 10Base-T, který je zapojen do měřeného prvku. Z měřeného prvku je 10Base-T převeden na 10Base-2 pomocí třetího HUBu. Segment 10Base-2 je připojen k druhému kanálu osciloskopu.

Měření bylo prováděno na 10Base-2, tedy na vedení realizovaném koaxiálním kabelem, z několika důvodů. Tím prvním je snadnější měření signálu. Signál je veden středovým vodičem koaxiálního kabelu a okolní úplet tvoří stínění, tedy zem signálu. Digitálním osciloskopem tedy měříme napětí středového vodiče proti zemi. Specifikace 10Base-T využívá dva páry kroucené dvoulinky, jeden pár je použit pro vysílání a druhý pro příjem dat. Logika je dána rozdílovým napětím mezi dvěma vodiči odpovídajícího směru. Měření na tomto vedení by byla problematická z důvodů chybějícího zemnění a nutností současně měřit v obou směrech. Dalším důvodem je snaha o co nejmenší ovlivnění přenosové soustavy. Měření na koaxiálním kabelu ovlivňuje přenosovou soustavu jen vnitřní impedancí osciloskopu. Na obr. 1 je zobrazeno zapojení, které je použito pro změření latence obou HUBů a pro pozdější eliminaci jejich zpoždění.

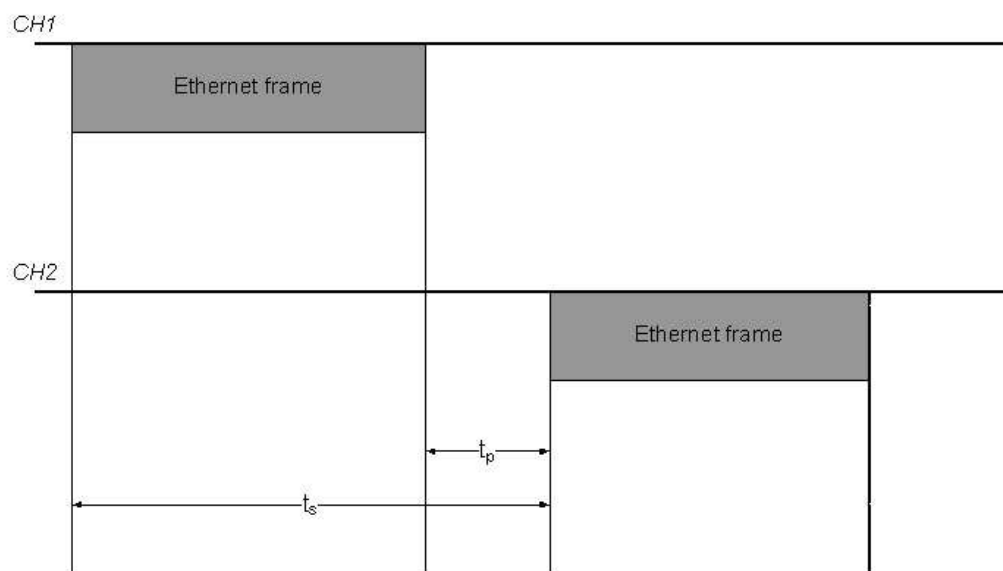


Obrázek 1: Měření zpoždění HUBů

Na obr. 2 je znázorněno zapojení již s měřeným síťovým prvkem. Vzhledem k tomu, že zpoždění HUBů je řádově menší než zpoždění měřených síťových prvků (viz. tab 1), můžeme je tedy zanedbat. Na obr. 3 jsou znázorněné průběhy signálů na jednotlivých kanálech. Měřena byla doba mezi prvním bitem rámce přicházejícího na vstup měřeného prvku a prvním bitem téhož rámce z prvku vystupujícího. Tento interval je dále označován jako t_s . Interval označený jako t_p odpovídá době, kterou daný prvek potřebuje na to, aby již přijatý rámec začal přeposílat na odpovídající port. Měřené intervaly jsou graficky zachyceny na obrázku 3.



Obrázek 2: Měření zpoždění síťových prvků (switche Cisco)



Obrázek 3: Časové intervaly měření zpoždění rámců

4 Naměřené hodnoty

Tabulka 1 zachycuje zpoždění HUBu zapojeném dle obrázku 1. Vidíme, že zpoždění rámce není závislé na velikosti rámce. HUB má také nejmenší latenci ze všech použitých prvků, na druhou stranu ale nesmíme zapomínat na jeho hlavní nevýhodou, kterou je společná kolizní doména.

Délka rámce	Δt_s	Δt_p
98B	$0.680\mu s$	–
546B	$0.680\mu s$	–
1492B	$0.680\mu s$	–

Tabulka 1: Zpoždění na hubu CentreCom MR820TR

V tabulce 2 vidíme latenci switche Cisco Catalyst 1900, který je nakonfigurován na přepínání metodou fragment free. Z naměřených časů vidíme, že v tomto režimu jsou rámce odesílány se zpožděním přibližně $71.6\mu s$, což odpovídá principu této metody. Switch musí přijmout minimálně prvních $51.2\mu s$ rámce, kdy se přenesou prvních 512 bitů. Samotné přepnutí tedy trvá méně než $20\mu s$. Dokumentace k Cisco Catalyst 1900 uvádí zpoždění FIFO v režimu fragment free $70.0\mu s$.

Délka rámce	Δt_s	Δt_p
98B	$71.6\mu s$	–
546B	$71.6\mu s$	–
1492B	$71.6\mu s$	–

Tabulka 2: Zpoždění na přepínači Catalyst 1900 v režimu „fragment free“

Situace se rapidně změní, pokud změníme konfiguraci a nastavíme režim store and forward. Tomuto odpovídá tabulka 3. Vidíme zde, že zpoždění rámce již závisí na jeho délce. Je to způsobeno tím, že switch musí přijmout celý rámec, uložit jej do vyrovnávací paměti, zpracovat a nakonec odeslat odpovídajícím portem.

Délka rámce	Δt_s	Δt_p
98B	$96.0\mu s$	$7.0\mu s$
546B	$456.0\mu s$	$7.0\mu s$
1492B	$1210.0\mu s$	$7.4\mu s$

Tabulka 3: Zpoždění na přepínači Catalyst 1900 v režimu „store and forward“

V tabulkách 4 a 5 jasně vidíme, že switche Cisco Catalyst 2950 a 2900 používají metodu store and forward. Rychlosti zpracování rámce se neliší od typu Catalyst 1900 v režimu store and forward.

Délka rámce	Δt_s	Δt_p
98B	100.0 μs	11.0 μs
546B	456.0 μs	12.0 μs
1492B	1200.0 μs	12.0 μs

Tabulka 4: Zpoždění na přepínači Catalyst 2950 v režimu „store and forward“

Délka rámce	Δt_s	Δt_p
98B	101.0 μs	14.0 μs
546B	461.0 μs	14.0 μs
1492B	1210.0 μs	15.0 μs

Tabulka 5: Zpoždění na přepínači Catalyst 2900 v režimu „store and forward“

V tabulce 6 jsou zachyceny doby zpoždění na switchi L3 (layer 3 switching) Cisco Catalyst 3550. Vidíme, že má z výše uvedených prvků největší latenci. Je to ale dáno tím, že na rozdíl od předchozích modelů, které pracovaly jen na 2. vrstvě, tedy jen s hlavičkou ethernetového rámce, tento switch pracuje i s obsahem. Zpracování obsahu paketu si vyžádá více strového času, než zpracování pouze hlavičky rámce. Tomuto odpovídá i větší zpoždění při průchodu rámců.

Délka rámce	Δt_s	Δt_p
98B	118.0 μs	28.0 μs
546B	516.0 μs	70.0 μs
1492B	1380.0 μs	168.0 μs

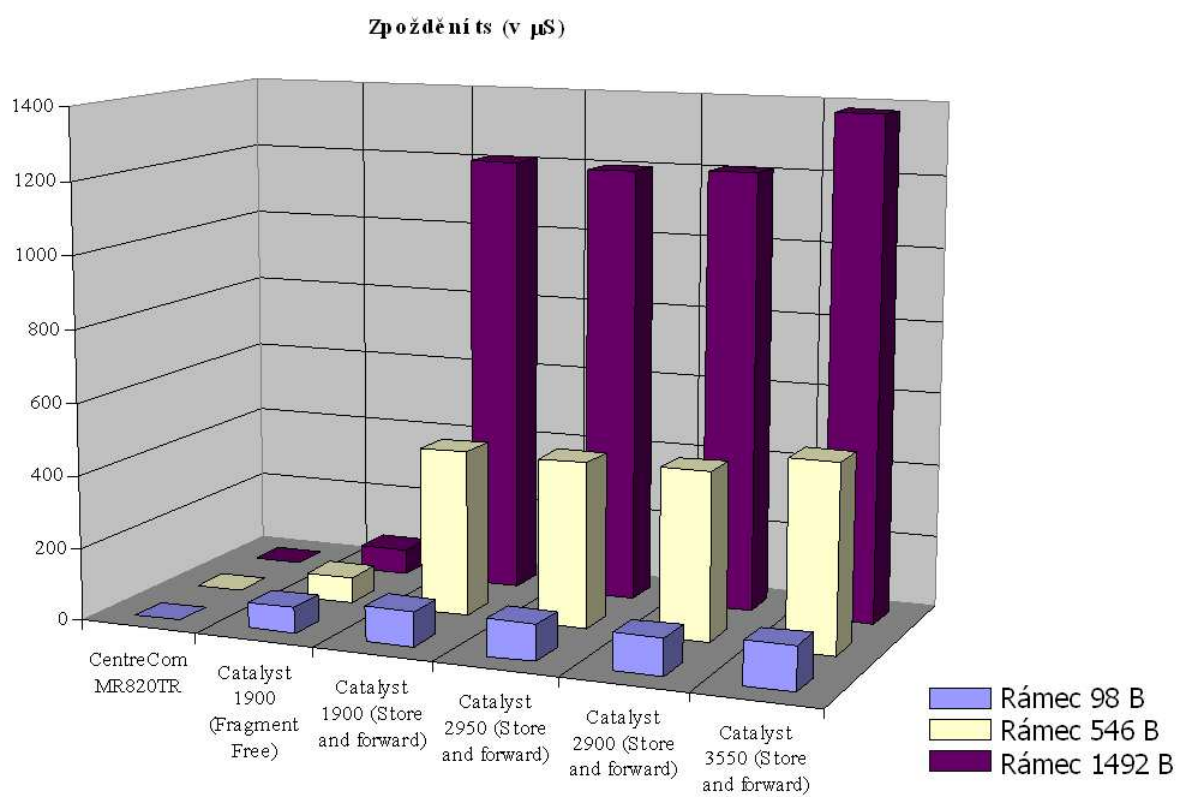
Tabulka 6: Zpoždění na přepínači Catalyst 3550 v režimu „store and forward“

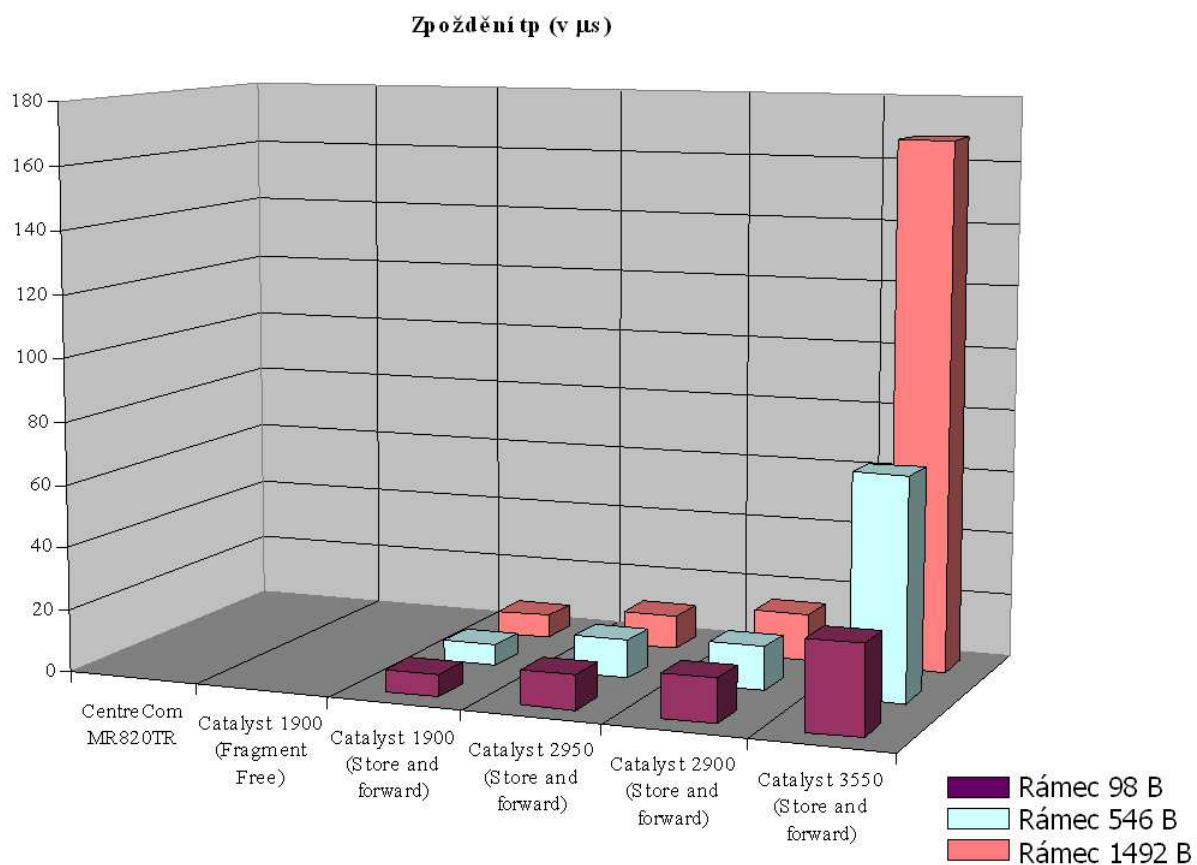
Tabulka 7 shrnuje naměřené časy pro všechny měřené síťové prvky. Grafy 4 a 5 jsou grafickým vyjádřením uvedené tabulky.

Označení prvku	Typ prvku	Velikost rámce					
		98B		546B		1492B	
		Δt_s	Δt_p	Δt_s	Δt_p	Δt_s	Δt_p
CentreCom MR820TR	hub	0,68	–	0,68	–	0,68	–
Catalyst 1900	switch ^a	71,6	–	71,6	–	71,6	–
Catalyst 1900	switch ^b	96,0	7,0	456,0	7,0	1210,0	7,4
Catalyst 2950	switch ^b	100,0	11,0	456,0	12,0	1200,0	11,0
Catalyst 2900	switch ^b	101,0	14,0	461,0	14,0	1210,0	15,0
Catalyst 3550	switch ^b L3	118,0	28,0	516,0	70,0	1380,0	168,0

^a fragment free, ^b store and forward

Tabulka 7: Shrnutí zpoždění prvků (v μs)

Obrázek 4: Srovnání zpoždění (t_s)



Obrázek 5: Srovnání zpoždění (t_p)

Reference

- [1] Jiří Peterka : eArchiv (<http://www.earchiv.cz/a97/a704k150.php3>)
- [2] Cisco: Catalyst 2820 Series and Catalyst 1900 Series Command Reference
- [3] Cisco: Catalyst 1900 Installation and Configuration Guide