OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

1 ÚVOD

V polovině devatenáctého století se svět začal měnit díky bezprecedentnímu rozvoji technologie v různých oblastech obzvláště telekomunikacích. Tehdy bylo vynalezeno, díky vědci *Alexanderovi Grahamu Belloni* – 1870, nové zařízení, jenž dokáže elektricky přenášet řeč. Tento a další vynálezy otevřely lidem a umožnily, jim získat zcela odlišný úhel pohledu. Vývoj v oblasti telekomunikací postupně směřoval až ke vzniku internetu na konci 20. století.

Internet nepochybně významně ovlivňuje všechny aspekt našeho každodenního života. Dostali jsme se do situace, kdy někteří z nás bez Internetu nedokážou normálně žít. Řada lidí na internetu postavila svoji pracovní i osobní komunikaci. Podle webové stránky *Middle East Online* se *Bill Gates* domnívá, že se lidí s technologií reagují více přirozeně, když používají zařízení umožňující *chating* a *touch*. Pro podnikatele proto bylo důležité, aby bylo zprostředkováno lidům lepší a snadný způsob ke spojení (obzvláště komunikace přes zvuku – *Messenger* nebo VoIP) což vede k dobrému byznysu.

Tato bakalářská práce se zabývá problematiku kvality VoIP hovoru mezi na síti. V dalších odstavcích bude diskutováno o důležitosti implementace QoS pro VoIP, použité parametry QoS a její navržené služby (budou se v této zprávě nazývat „mechanizmy“), jako např. Služba *best effort*, integrované a diferencované služby. Poté budou stručně probrány základy síťových protokolů TCP/IP, úvod do problematiku QoS – způsoby změření kvality hlasu jako třeba MOS a E-Model. Na základě těchto poznatků bude pomocí uvedených softwarů navržena laboratoř kvality služby. Tato laboratoř bude zaměřena na testování kodeků z hlediska kvality přenášeného zvuku s ohledem na šířku pásma, jitter, zpoždění a ztrátovost paketů. Řešení spočívá v tom, které kodeky jsou pro danou službu vhodné. Dalším řešením může být větší šířka pásma – finančně nákladné.

2 TCP/IP

V networking TCP/IP protokoly hrají nejdůležitější roli v rámci přenosu pakety, zaručení doručování paketů do cílové stanice, ohlašování existence chyb při přenosu a řada dalších služeb. Síťových protokolů existuje celá řada. V Internetu se používají síťových protokoly TCP/IP. Síťový protokol je norma napsaná na papíře (resp. textovým editorem na počítači). V Internetu se používají normy nazývané *Request For Comments* – zkratkou RFC, které se číslují průběžně od jedničky. V současné době jich jsou necelé tři tisíce. Mnohé však postupem času zastaraly, takže z první tisícovky jich je aktuálních jen několik[6].

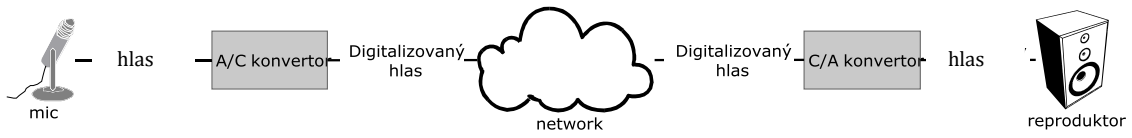
TCP/IP model byl vytvořen v roce 1970 od agentury DARPA *Spojené státy Ministerstvo národní obrany*, často také označovaný jako Internetový model, jenž popisuje proces při zpracování informací na základě vrstev. Na tomto modelu nejvýznamnější protokoly jsou TCP a IP, které dohromady tvoří základní části tohoto modelu, proto se symbolizuje za tímto kódem TCP/IP. TCP/IP model a jeho související protokoly jsou udržované od *Internet Engineering Task Force* (IETF). Je to v podstatě zjednodušení ISO/OSI modelu, jenž standardizoval mezinárodní standardizační úřad (ISO). Jde o modelu *Open System Interconnection* (OSI). Model uvádí všeobecné principy sedmivrstvé síťové architektury. Popisuje vrstvy, jejich funkce a služby. Nejsou zde zařazeny žádné protokoly, které by vyžadovaly zbytečně mnoho detailů. Každá ze sedmi vrstev objasňuje skupinu pevně definovaných funkcí potřebných pro komunikaci. Pro svou činnost využívá služeb své sousední nižší vrstvy. Své služby pak poskytuje sousední vyšší vrstvě. Nevýhodou ISO/OSI modelu je jeho přílišná složitost. Pochopení TCP/IP modelu je důležité pro správnou konfiguraci sítě či analýzu a řešení problémů při komunikaci mezi počítači nebo aplikacemi.

Implementace TCP/IP modelu je rozdělena do tří částí. Nejnižší část, vrstva fyzického rozhraní, je implementována v síťové kartě a jejím ovladači. Vyšší vrstvy, internetová a transportní, jsou součástí síťových modulů operačních systémů (TCP/IP stack), které bývají implicitně nainstalovány. Poslední vrstva, aplikační, je implementována buď přímo v aplikacích (webový prohlížeč) nebo jako systémové služby (např. DNS klient). Tab. 1.1 ilustruje architektury obou těchto modelů, názvy vrstev a použité protokoly pro příslušnou vrstvu. ***Application Layer*** zajišťuje komunikaci na nejvyšší úrovni, tedy komunikaci mezi samotnými procesy a aplikacemi, které běží na počítači. Tato vrstva také řeší reprezentaci dat (vhodné kódování aplikačních dat pro přenos, převod dat do tohoto kódování a zpět) a řízení dialogu. ***Transport Layer*** vytváří logické spojení mezi koncovými body. Transportní protokoly rozdělují aplikační data na menší jednotky tzv. pakety, které jsou poté posílány po síti. ***Internet Layer*** vytváří logické spojení mezi počítači. Protokoly internetové vrstvy směrují zaobalené pakety tzv. datagramy, na určené místo na základě jejich cílové adresy. Internetová vrstva se snaží doručit data nejvhodnější cestou, tzv. doručení s největším úsilím (*Best-effort delivery*). Pokud dojde při přenosu ke ztrátě dat, je o tom odesilatel informován a musí sám zajistit opětovné přenesení dat. ***Link Layer*** popisuje standardy pro fyzické médium a elektrické signály. Tato vrstva definuje funkce pro přístup k fyzickému médiu a zajišťuje zabalování datagramů do tzv. rámců.

Tab. 2. 1: TCP/IP a OSI modely.

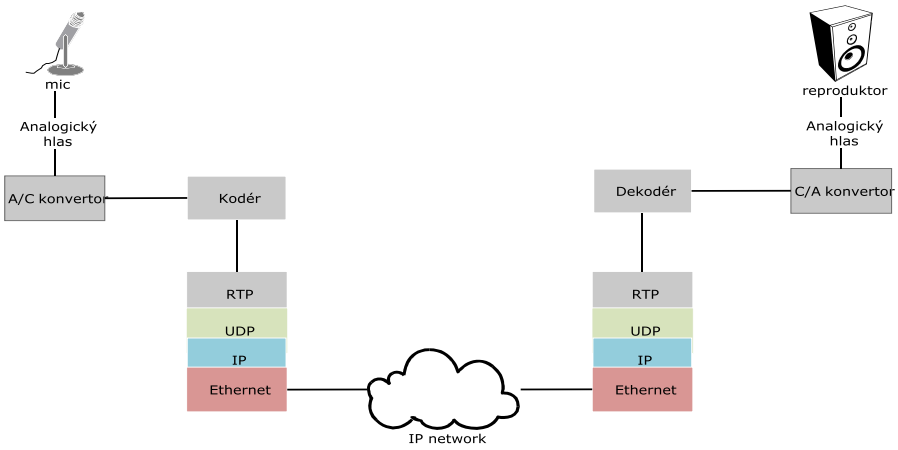
TCP/IP	OSI
4. Application Layer HTTP, IMAP, RTP, RTSP, SIP, SSH, TLS/SSL, atd.	7. Application Layer HTTP, FTP, SIP, SMTP, Telnet, atd.
3. Transport Layer TCP, UDP, DCCP, SCTP, RSVP, atd.	6. Presentation Layer MIME, XDR
	5. Session Layer Named Pipes, NetBIOS, SAP
2. Internet Layer IP (IPv4, IPv6), ICMP, ICMPv6, IGMP, IPsec, atd.	4. Transport Layer TCP, UDP, PPTP, SCTP, SSL, TLS
	3. Network Layer IP, ICMP, IPsec, IGMP
1. Link Layer ARP/InARP, NDP, OSPF, Tunnels (L2TP), PPP, Media Access Control (Ethernet, DSL, ISDN, FDDI) atd.	2. Data Link Layer ARP, CSLIP, SLIP, Frame relay, PPP
	1. Physical Layer RS-232, Ethernet, POTS, DSL

Tradiční způsob vypracování a transportování analogických informací přes sítě, která nevyužívá IP paketů ke vzájemné komunikaci mezi prezenčními uživateli sítě, zobrazuje obr. 1.1.



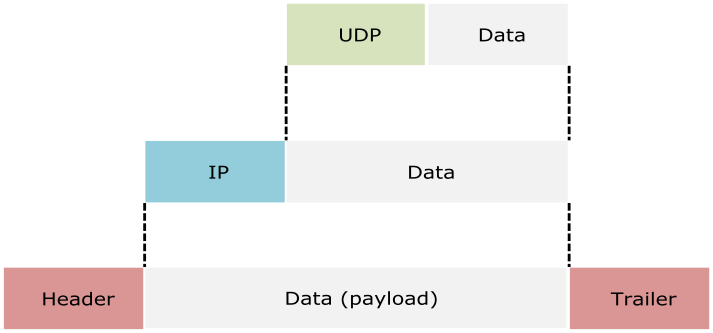
Obr. 2. 1: Tradiční postupy transformace hlasu.

Každopádně dominantní model v současné době je TCP/IP. Zpracování dat je na skoro všechny počítače založeno na tomto modelu, příklad viz obr. 1.2. Užitečná informace (v rámci této zprávy je hlas) se nejdříve od stránky odesílatele (mluvič) konvertuje do číslicové podoby pomocí příslušného konvertoru a pak kóduje prostřednictvím dostupných kodeků (PCM, ADPCM, G.711, G.719, G.722, atd.) v kodéru. Kodek je algoritmus nebo počítačový program, většinou instalovány jako software na serveru anebo přímo v hardware (ATA, IP telefon, atd.), schopný kódovat nebo dekódovat signály či tok digitálních dat. Slovo kodek je kombinací slov „kompresor–dekompresor“ nebo, častěji, „kodér–dekodér“. Tyto kodeky bývají podpořené příslušnými softwarovými telefony (v tomto experimentu je použit software *x-lite*). Kodeky konvertují analogové audio na digitální vysílání a pak počítač digitálního zvuku zpět na audio). Kodeky běžně splňují tří úkolů (velmi málo vykonat jen poslední): Kódování – dekódování; Komprese – dekomprese a Šifrování – dešifrování.



Obr. 2. 2: Transformace hlasu na IP síť.

Potom v Application Layer kódované informace budou příslušnými protokoly (RTP – Real Time Protokol) zabaleny do tzv. Paketů. Nižší vrstva Transport Layer poskytuje potřebné transformační protokoly. To se uskuteční pomocí procesu, který je známý pod termínu encapsulation nebo zapouzdření viz obr. 1.3. V počítačovém sítí zapouzdření v rámci telekomunikačních modelů, znamená, že je data od nižší resp. vyšší vrstvy modelu skryté v takovém novém objektu, k němuž je přístup omezen pouze na členy této vrstvy. Čili úkol každé vrstvy spočívá jenom v tom, že obdržená data zabaluje do své vlastní „balíčku“, s jehož obsahem není schopna ani vyšší ani nižší vrstva manipulovat. V obr. 1.3 Transport Layer do faktu neví, že je data od Application Layer v podstatě digitálně konvertovaný hlas plus UDP protokolu.

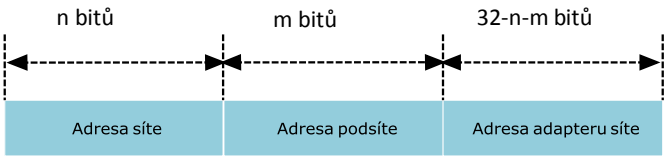


Obr. 2.3: Zapouzdření data skrze transportní, internetové a linkové vrstvy.

Na linkové vrstvě je základní jednotkou pro přenos dat „datový rámec“, jenž se skládá ze záhlaví (*Header*), přenášených dat (*Payload*) a zápatí (*Trailer*), a je jediný v sérii, který má záhlaví a zápatí. Ostatní datové rámce od vyšších vrstev mají pouze záhlaví. Datový rámec nese v záhlaví linkovou adresu příjemce, linkovou adresu odesílatele a další řídící informace. V zápatí nese mj. obvykle kontrolní součet z přenášených dat. Pomocí něho lze zjistit, zdali nedošlo při přenosu k porušení dat. V přenášených datech je pak zpravidla nesen paket síťové vrstvy. V přijímací straně inverzními postupy se získá původní analogový hlas.

2.1 Internet Protokol – IP

IP je primární protokol na internet Layer (Internetová vrstva) TCP/IP modelu. Má za úkol doručovat pakety ze zdrojového počítače do cílového pouze na základě jejich adres. IP adresa slouží k jednoznačné identifikaci zařízení (síťová karta počítače, směrovače, přepínače, atd.) v globální nebo lokální sítě. Každý datagram obsahuje adresy zdrojového a cílového koncového uzlu a internetová vrstva se snaží doručit datagram od zdroje k cíli. Pakety jsou předány z jednoho směrovače do druhého. Pro tento účel internetový protokol definuje způsob adresování a vhodné struktury pro datagramové zapouzdření. Každý směrovač na cestě paketů řeší samostatně způsob předání k dalšímu směrovači, až se pakety dorazí do cíle. IPv4 adresy jsou 32–bitová čísla, jež se zapisují v dekadickém formátu s tečkovou notací po osmi bitech. Tedy každá adresa se skládá ze čtyř oktetů. Hodnota každého oktetu se může pohybovat od 0 do 255. Když je oktet přeměněn na binární, bude skládat z osmi bitů, z nichž každý má určitou hodnotu. Z hlediska struktury se dělí IPv4 adresa na tři základní části, jak je zobrazeno na obr. 2.4. Každá adres v síti musí nutně být jedinečná a identifikuje pouze jedno zařízení.



Obr. 2. 4: Struktura IPv4 adresy.

Dříve byla IPv4 adresa tvořena pouze ze dvou částí: ID sítě a ID hostitele. ID sítě slouží k identifikaci konkrétní sítě nebo podsítě. Vzhledem k tomu, ID hostitele identifikuje hostitele na určité síti nebo podsíti. Například, s IPv4 adresou 132.10.26.2 a výchozím maskem podsítě 255.255.0.0; ID sítě je 132.10 a hostitelské ID je 26.2. Toto členění ovšem bylo příliš hrubé a docházelo tak k zbytečnému plýtvání adres, jelikož adresa sítě byla tvořena vždy pouze prvními osmi bity a zbylé adresy rozhraní, kterých bylo 16 miliónů pro každou síť, byly využity jen minimálně. Proto se později IPv4 adresy rozdělily do tříd, které se odlišovaly velikostí části, jež byla vyhrazena pro adresu sítě, tak se vytvořilo podstatně více sítí pro méně rozhraní. Nakonec se i toto rozdělení ukázalo jako nevhodné a adresa rozhraní se rozdělila na část adresy podsítě a rozhraní, což umožnilo ještě jemnější rozdělování rozhraní do sítí. Adresu sítě pro danou koncovou síť přiděluje vždy poskytovatel připojení (přesněji lokální registrátor). Jak bude rozdělena lokální část adresy, tedy jaká část bude vyhrazena pro adresy podsítí a jaká část pro adresy rozhraní, určuje již správce dotyčné koncové sítě.

Pro určení hranice mezi adresami podsítě a rozhraní se využívá tzv. masky podsítě (subnet mask). Stejně jako v případě IPv4 adresy, i maska podsítě je 32bitové číslo zapsané ve stejném formátu jako IPv4 adresa. V binárním tvaru obsahuje jedničky tam, kde se v IPv4 adrese nachází adresa sítě a podsítě a nuly tam, kde je adresa rozhraní. Jelikož část obsahující adresu podsítě může být různě velká, musí být součástí konfigurace síťového rozhraní vždy i maska podsítě.

Například IP adresa 192.168.0.1 má výchozí mask podsítě 255.255.255.0. První tři oktety masku podsítě jsou digitálně nastaveny na 1. To znamená, že všechny první tři oktety IP adresy identifikují ID sítě. Od posledního oktetu maska podsítě je nastaven jen na 0, což poslední oktet adresy IP se používá k identifikaci konkrétního hostitele v síti. Zkrátka maska podsítě se používá k určení, zda je cílový počítač na lokální nebo globální síti.

Tab. 2.2 zachycuje rozdělení IPv4 adres do jednotlivých tříd s informacemi o tom, jak velká část IPv4 adresy je vyhrazena pro identifikaci sítě a jak velká část pro identifikaci rozhraní. Z části vyhrazené pro adresu rozhraní lze ještě, v případě potřeby, ubrat pár bitů pro identifikaci podsítě, jak bylo zmíněno dříve. Dnes se již rozdělení do tříd nevyužívá, jelikož bylo nahrazeno rozdělením podle CIDR, které je flexibilnější.

Tab. 2. 2: Třídy IPv4 adres.

Třída	**Prefix sítě**	**1. bajt**	**Maska**	**Bitů sítě**	**Bitů počítačů**	**Počet sítí**
A	0	0 – 127	255.0.0.0	7	24	126
B	10	128 – 191	255.255.0.0	14	16	16 384
C	110	192 – 223	255.255.255.0	21	8	2 097 152
D	1110	224 – 239	Skupinové vysílání (multicast)			
E	1111	240 – 255	Rezervováno pro pozdější využití			

Nedílnou povahou, kterou si zaslouží zmínku, je směrování sloužící k dopravě datagramů ze zdrojového koncového uzlu do cílového koncového uzlu (tedy nejčastěji k přenosu dat mezi dvěma počítači). Směrování se provádí na základě směrovacích tabulek, jenž mohou být nastaveny staticky uživatelem nebo dynamicky pomocí směrovacích protokolů jako RIP nebo OSPF. Směrovací tabulky obsahují informace o tom, kterými porty směrovače nebo skrz které síťové rozhraní počítače se dá dostat do sítě, ve které leží koncový uzel s cílovou adresou. V dnešní době se pro směrování používá hlavně tzv. beztřídní mezidoménové směrování (CIDR, Classless Inter-Domain Routing), jenž umožňuje explicitně specifikovat předěl mezi částí s adresou sítě a částí s adresou počítače. Adresy se v tomto případě zapisují ve formátu IPv4/Y, kde Y je počet bitů adresy sítě. Pokud směrovači přijde datagram, podívá se do směrovací tabulky a zjistí, skrz které porty se dá dostat do sítě, do které náleží cílová IPv4 adresa v datagramu. Pokud je jich více, některý vybere na základě dalších informací (např. podle nastavené metriky, podle zahlcení dané cesty apod.). V případě, že datagram přišel na port, jenž vede do sítě, kam tento datagram směřuje, dojde k jeho zahození. Druhá situace, kdy může dojít k cílenému zahození datagramu, je v případě, že směrovač odděluje interní síť od sítě internet a cílová adresa v datagramu náleží do privátní sítě. Takovéto datagramy jsou nesměrovatelné v internetu! Seznam privátních sítí lze nalézt v tab. 2.3.

Tab. 2. 3: Seznam speciálních rozsahů IPv4 adres

CIDR adresový blok	Popis
0.0.0.0/8	Aktuální síť (pouze pro zdrojové adresy).
10.0.0.0/8	Privátní síť.
127.0.0.0/8	Loopback (vlastní adresa počítače).
169.254.0.0/16	Privátní síť (APIPA).
172.16.0.0/12	Privátní síť.
192.88.99.0/24	IPv6 to IPv4 překlad.
192.168.0.0/16	Privátní síť.
224.0.0.0/4	Multicast (skupinové vysílání, předchozí třída D).
240.0.0.0/4	Rezervováno (předchozí třída E).
255.255.255.255	Broadcast (všesměrové vysílání).

Posledním případem je situace, kdy cílová adresa je adresa pro všesměrové vysílání (broadcast) a ostatní porty směřují do jiných podsítí, takového datagramy nikdy nesmí překročit hranice podsítě.

V současnosti je internetový protokol verze 4 (IPv4) dominantní protokol Internetu, přestože nástupce, internetový protokol verze 6 (IPv6) aktivně rozmístěn po celém světě. IPv6 adresy jsou značně odlišné od IPv4 adres. Jsou to 128bitová čísla, která se standardně zapisují v hexadecimálním formátu s dvojtečkovou notací po skupinách 16 bitů. Tedy každá adresa je ve šest oktetů pohybující se od 0000 do FFFF. Často se ovšem zapisují ve zkrácených formátech, kdy se vynechávají úvodní nuly jednotlivých skupin nebo se slučují nulové skupiny a místo nich se píše pouze:: (toto nahrazení se může ovšem použít jen jednou v rámci jedné adresy). Na rozdíl od IPv4 zde chybí typ adres pro všesměrové vysílání (broadcast), tyto adresy jsou u IPv6 nahrazeny speciálním typem skupinových adres. Různé formy zápisu IPv6 adres shrnuje tab. 2.4.

Tab. 2. 4: Různé formáty IPv6 adres.

IPv6 adresa	Popis
fec0:0000:0000:000a:f563:5add:6fc4:152e	Standardní formát.
fec0:0000:0000:000a:f563:5add	S vynecháním úvodních nul každé skupiny.
fec0:0:0:a:f563:5add:6fc4:15	S vynecháním úvodních nul každé skupiny a sloučením po sobě jdoucích nulových skupin.

2.1.1 IP datagram

Obecně platí, že pojem paket se vztahuje ke každé zprávě ve formátu paket, kdežto termín datagram je obecně vyhrazeno pouze pro pakety z "nespolehlivých" služeb. Spolehlivé služby jsou ty, které upozorní uživatele v případě selhání doručení, zatímco nespolehlivé služby na tuto skutečnost uživateli neupozorní. TCP a IP poskytují spolehlivé služby, zatímco UDP a IP poskytují nespolehlivé služby. Všechny tyto protokoly používají pakety, ale UDP pakety jsou obecně nazývány datagramů.

Záhlaví IPv4 se skládá z:

i. *4 bity, které obsahují verze, která určuje, zda je to paket IPv4 nebo IPv6,*
ii. *4 bity, které obsahují Internet Header Length, která je délka záhlaví násobená 4 byty (např. 5 znamená 20 bajtů),*
iii. *8 bitů obsahují, Type of Service občas také odkazuje na Quality of Service (QoS),*
iv. *16 bitů, které obsahují délku paketu v bajtech,*
v. *16 bitů, které obsahují identifikační značku, pomocí které lze obnovit paket z několika fragmentů,*
vi. *3 bity, které obsahují nulu, položku, která definuje, zda je paket může být fragmentovaný nebo ne (DF: Don't fragment) a položku k určení, zda další fragmenty paketu následují (MF: More fragment),*
vii. *13 bitů, které obsahují fragment offset, pole k určení pozici fragmentu v rámci původního paketu,*
viii. *8 bitů obsahující TTL viz dále,*
ix. *8 bitů, které obsahují protokol (TCP, UDP, ICMP, atd. ...),*
x. *16 bitů, které obsahují záhlaví Checksum,*
xi. *32 bitů obsahují zdrojové IP adresy,*
xii. *32 bitů obsahují cílovou adresu.*

Čtyři dílče tvoří protokol IPv4. Jsou následovně: Vlastní protokol IP, Služební protokol ICMP sloužící zejména k signalizaci mimořádných stavů, Služební protokol IGMP sloužící pro dopravu adresných oběžníků a Služební protokoly ARP a RARP, které jsou často vyčleňovány jako samostatné, na IP nezávislé protokoly, protože jejich rámce nejsou předcházeny IP–header (záhlaví).

IPv4 má v hlavičce informační pole ToS (Type of Service) [5]:

1000 minimalizuj zpoždění,
0100 maximalizuj propustnost,
0010 maximalizuj spolehlivost,
0001 minimalizuj finanční náklady,
0000 normální služba.

Je to položka, která v praxi nenašla svého naplnění. V normách RFC–791 a RFC–1349 lze nalézt konkrétní návrhy využití. Záměr spočíval v jistém nedostatku IP–protokolu jehož podstatou je skutečnost, že v Internetu není zaručena šíře přenosového pásma mezi účastníky. Jistého vylepšení se mělo dosáhnout právě touto položkou, pomocí které je možné označit některé IP–datagramy tak, aby byly dopravovány přednostně či aby byla zaručena rychlá odezva atp. V praxi se nelze setkat s případem, že by se komunikovalo přímo IP–protokolem. Vždy je použit protokol vyšší vrstvy (TCP nebo UDP) nebo jeden ze služebních protokolů *Internet Control Message Protocol* (ICMP) či *Internet Group Management Protocol* (IGMP). Protokol vyšší vrstvy obsahuje číselnou identifikaci protokolu vyšší vrstvy, který využívá IP–datagram ke svému transportu. Protokoly ICMP a IGMP jsou sice formálně součástí protokolu IP, avšak chovají se jako protokoly vyšší vrstvy, tj. v přenášeném paketu je záhlaví IP–protokolu následováno záhlavím protokolu ICMP (resp. IGMP) [6].

ICMP je nedílnou součástí IP protokol soupravy, jak je definován v RFC–792. Jedná se především o odesílání chybových hlášení, např. požadovaná služba není dostupná, nebo že hostitel nebo směrovač nebyl dosažen. ICMP zprávy jsou typicky vytvořené jako reakce na chyby při odesílání IP datagramů, na diagnostiky nebo účely směrování. ICMP zprávy se konstruují na ***Internet Layer***, obvykle z IP datagramu, který vygeneroval ICMP odezvu. IP vystihuje příslušné ICMP zprávy s novou IP hlavičkou (aby se ICMP zpráva dostala zpět k původnímu odesilateli) a vzniklý datagram odešle obvyklým způsobem. V té nové IP hlavičce mj. *Time To Live* (TTL) osmi bitová položka, která v počítačové síti omezuje jeho maximální dobu existence a chrání ji tak před zahlcením, které by mohly způsobit datagramy zacyklené v nekonečných smyčkách. Položka TTL je nastavena na výchozí hodnotu při vytvoření datagramu (obvykle 64) a automaticky snižována alespoň o 1 při průchodu jakýmkoliv směrovačem. Po dosažení nuly je datagram zahozen a odesílatel je o tom informován ICMP zprávou *Time Exceeded* (zpráva číslo 11). Položku TTL využívá ke své činnosti program *ping* nebo *traceroute*. Ačkoli ICMP zprávy jsou obsažené ve standardních IP datagramů, jsou ale obvykle zpracovávány jako zvláštní případ, který se odlišuje od běžných IP zpracování. Mnoho běžně používaných síťových diagnostických utilit je založeno na ICMP zprávách. Příkaz Traceroute se provádí pomocí přenosu UDP datagramy se speciálně stanovenou oblasti IP hlavičku TTL, související *ping* utilita je implementována pomocí ICMP "*Echo Request*" a "*Echo Reply*" zprávy.

Strukturu ICMP záhlaví zobrazuje tab. 2.5. V níž: **Type** – typ ICMP, jak je uvedeno v A příloze. **Code** – další specifikace typu ICMP, např.: ICMP o nedosažitelné destinaci mohl mít toto pole nastaveno na 1 až 15, každý má jiný význam. **Checksum** – Toto pole obsahuje kontrolu chyb hodnoty počítány ze záhlaví ICMP plus data, s hodnotou 0 pro tuto oblast. Algoritmus je stejný jako záhlaví kontrolní součet pro IPv4. **ID** – Toto pole obsahuje hodnotu ID, který by měl být vrácen v případě *ECHO REPLY*. **Sequence** – Toto pole obsahuje sekvence hodnotu, kterou by měla být vrácena v případě *ECHO REPLY*. Záhlaví ICMP začíná po IPv4 záhlaví.

Tab. 2. 5: Struktura ICMP paketu.

Bity	0-7	8-15	16-23	24-31
0	**Type**	**Code**	**Checksum**	
32	ID		**Sequence**	

Na druhé straně IGMP je používán pro řízení složení skupin Internet *multicast* (přeposílání IP datagramů z jednoho zdroje skupině více koncových stanic) protokolu. IP hostitelé a vedlejší *multicast* směrovače používají IGMP k etablování členství *multicastických* skupin. IGMP je nutný pouze pro IPv4 sítě, zatímco je *multicast* v IPv6 sítích nakládáno jiným způsobem. Je to doplňující protokol pro *unicast* komunikace. IGMP lze využít pro on–line *multicasting* (streamování) videa a her, a umožňuje efektivnější využití zdrojů při podpoře těchto typů aplikací. Všechny IGMP pakety mají v IP záhlaví nastavenu položku TTL=1. Existují tři verze IGMP, jak jsou definovány v RFC dokumenty IETF. IGMPv1 je definován v RFC–1112, je IGMPv2 definován v RFC–2236 a IGMPv3 je definován v RFC–3376. Pakety protokolu IGMPv2 používají volbu (rozšíření) IP záhlaví „Upozornění pro směrovač (IP *Router Alert Option*)". IGMPv3 zdokonaluje IGMPv2 hlavně tím, že přidá schopnost naslouchat multicast pocházející ze skupiny IP adres.

Strukturu ICMP záhlaví zobrazuje tab. 2.6. V ní: **Type** – Existují tři typy zpráv IGMP se týkají *host – router* interakce:

0x11 = ***Membership Query***; Existují dva subtypy zpráv, první je *General Query*, využívá se k poznání, které skupiny mají členy na připojené sítě, a druhý je tzv. *Group-Specific Query*, který se používá k poznání, pokud zvláštní skupina má nějaké členy na připojené síti. Tyto dvě zprávy jsou rozděleny podle adresy skupiny.

0x16 = Členství Report – verze 2.

0x17 = Opustit skupinu.

Max Resp Time – určuje lhůtu pro příslušnou zprávu. Toto pole má význam pouze v *Membership Query* (0x11), v jiných zpráv je nastaveno na nule a ignoruje přijímače. **Checksum** – 16 bitový pro součet celé IGMP zprávy (celé IP payload). Pro výpočet *checksum*, jeho pole se nastavuje na nulu. Při předávání paketů, *checksum* se musí nejprve spočítat a vložit do tohoto pole. Při přijímání paketů se *checksum* musí být ověřen před zpracováním paketu. **Group Address** – V *Membership Query* je pole ***Group Address*** nastaveno na nulu při odesílání *General Query*, a pustit skupinu je dotázán při odesílání *Group-Specific Query*.

Tab. 2. 6: Struktura IGMPv2 paketu.

Bity	0-7	8-15	16-23	24-31
0	**Type**	**Max Resp Time**	**Checksum**	
32	**Group Address**			

2.2 Transmission Control Protocol – TCP

Služba od protokolu TCP není právě omezena jenom na procesu dělení dat v takovém formátu, aby IP – protokol mohl je posílat. Z důvodu přetížení sítě, balancování dopravní zátěže, nebo jiné nepředvídatelné chování sítě, může být IP pakety ztracené nebo doručené v nesprávném pořadí. Pokud tomu tak bylo, TCP rozpoznává tyto problémy a žádá o přeposílání ztracených paketů, přeskupuje pakety posílané v nesprávném pořadí a dokonce pomáhá minimalizovat přetížení sítě s cílem snížit výskyt dalších problémů. Jakmile TCP protokol konečně připraví dokonalou kopii původně předaného údaje, předává „datagram" vyšší *Application Layer*. TCP je optimalizován pro správné doručení a nikoli včasné dodání. Takže hlavní úkol tohoto protokol je zajišťovat doručení paketů. Proto se TCP datagramy někdy vznikají poměrně dlouhým zpožděním (v řádu vteřin); při čekání na znovu uspořádání nebo přeposílání ztracených dat. TCP protokol proto není vhodný zejména pro *Real – time Aplikace*, jako je *Voice over* IP, online hry, atd. Pro takové aplikace jsou doporučené protokoly jako *Real – time Transport Protocol* (RTP) běžící na *User Datagram Protocol* (UDP), o něm bude řeč v další odstavci. TCP však použije značně mnoho z nejoblíbenějších internetových aplikací, včetně *World Wide Web* (WWW), *E-mail*, *File Transfer Protocol* (FTP), Secure Shell, peer-to-peer sdílení souborů, a některé *streaming* multimediálních aplikací.

Protokoly *Transport Layer*, nejvíce pozoruhodně TCP a UDP, použijí pro *host–to–host* komunikaci tzv. *porty*. V počítačovém sítí, port je specifická aplikace nebo specifický proces softwarového konstruktu, sloužící jako koncový bod ke komunikaci používaný *Transport Layer*. Specifický port je identifikován svým číslem, obyčejně známý jako číslo portu, se kterým je IP adresa spojena. *Internet Assigned Numbers Authority* (IANA) má za starost udržovat oficiální přiřazení čísel portů pro konkrétní použití. IANA je provozován od *Internet Corporation* pro přiřazená jména a čísla, lépe známý jako ICANN. Čísla portů jsou rozděleny do tří rozsahů: *well-known ports* (dobře známý porty), registrované porty, a dynamické nebo soukromých portů. *well-known* porty jsou od 0 do 1023 (v tab. 2.7 jsou uvedené některé tyto porty), registrované od 1024 do 49151 a dynamické nebo soukromé porty 49152–65535. Podle IANA, *well-known* porty přiděluje IANA a na většině systémů mohou být použity pouze systémovým (nebo root) procesem nebo programy provedený privilegovaným uživatelem.

Tab. 2. 7: Seznam některých nejpoužívanějších portů.

Port	TCP / UDP	Popis	registrovaný od IANA
23	TCP	Telnet	Ano
53	TCP / UDP	Domain Name Systém	Ano
80	TCP / UDP	World Wide Web– HTTP	Ano
443	TCP	HTTP over Secure Sockets Layer	Ano
465	TCP	SMTP over SSL	Ne

Konstrukce TCP paketu se skládá ze segmentu záhlaví a *data section*. Záhlaví TCP obsahuje 10 povinná pole a volitelné rozšířené pole (*Options*, oranžové pozadí v tabulce). D*ata section* navazuje záhlaví, jeho obsah je *Payload*. Délka *data section* není uvedena v záhlaví TCP paketu. Může se však vypočítat odečtením kombinované Délky záhlaví TCP a zapouzdřený IP záhlaví z celkové délky IP paketu (je uvedeno v záhlaví IP paketu). Konstrukci záhlaví TCP zobrazuje tab. 2.8, v níž značí: **Source port** (16 bitů) – identifikuje odesílající port, **Destination port** (16 bitů) – označuje přijímající port, **Sequence number** (32 bitů) – má dvojí roli:

- Pokud je položka SYN nastavena, pak je to původní s*equence number*. S*equence* number aktuálního prvního data bytu je pak toto s*equence number* plus 1.
- Pokud není položka SYN nastavena - prázdná, pak je to nahromaděné s*equence number* prvního data bajtu tohoto paketu pro aktuální sekci.

Dále **Acknowledgment number** (32 bitů) – Pokud je položka ACK nastavena, pak hodnota tohoto pole je další s*equence number*, které příjemce očekává. To potvrzuje příjem všech předchozích bytů (pokud existuje), **Data offset** (4 bity) – určuje velikost záhlaví TCP v 32bitových slov. Minimální velikost záhlaví je 5 slov a maximum je 15 slov a umožňuje tak minimální velikost 20 bytů a maximálně 60 bytů, který umožňuje až 40 bytů možností v záhlaví, **Reserved** (4 bity) – pro budoucí použití a mělo by být nastaveno na nulu, **Flags** (8 bitů) – obsahuje osm jednobitové podpoložky, **Windows size** (16 bitů) – určuje počet bajtů (za pořadové číslo v potvrzení poli), které je přijímač právě ochotni přijmout, **Checksum** (16 bitů) – používá se pro ověřování chyby záhlaví a data, **Urgent pointer** (16 bitů) – pokud je položka URG nastavena, pak je toto 16bitové pole vykompenzováno pořadovým číslem (sequence number) označující poslední *urgent data* byte, a poslední pole „ **Options**“ (0-320 bitů, dělitelné 32) – Délka tohoto pole je určena *data offset* pole. Options 0 a 1 jsou osmibitové nebo jednobajtové. Ostatní options naznačují celkovou délku *option* (vyjádřená v bajtech) v druhém bajtu.

Tab. 2. 8: Konstrukce záhlaví TCP paketu[1].

Bit offset	**0 - 15**			**16 - 31**
0	**Source port**			**Destination port**
32	**Sequence number**			
64	**Acknowledgment number**			
96	**Data offset**	**Reserved**	**Flags**	**Windows size**
128	**Checksum**			**Urgent pointer**
160 …	**Options (if data offset > 5)** **…**			

[1] Přehlednější struktura viz D příloha.

2.3 User Datagram Protocol – UDP

UDP používá jednoduchý přenosový model bez implicitní dialogy pro zajištění spolehlivosti, objednávky, či integritu dat. Čili, UDP poskytuje nespolehlivé služby. Díky tomu datagramy můžou dorazit do cílové stanice v jiném pořadí, objeví se duplicitní, nebo se ztratí bez předchozího upozornění. UDP předpokládá, že oprava chyb buď není nutná či již uskutečněna v žádosti, aby se zabránilo režii takového zpracování na úrovni síťového rozhraní. Aplikace citlivé na čase často používají UDP pakety, protože jim je zahazování paketů lepší než čekat na opožděné pakety, které nemohou být možnost v *Real – time* systému. Je – li zařízení opravy chyb jsou potřebná na úrovni síťového rozhraní, může aplikace používat *Transmission Control Protocol* (TCP) nebo *Stream Control Transmission Protocol* (SCTP), které jsou určené pro tento účel. UDP neposkytuje žádné záruky na horní vrstvu protokolu pro doručování zpráv. Protokol UDP si žádné státní zpráv UDP nezachovává, jakmile ji posílá. Z tohoto důvodu se UDP někdy označuje jako *Unreliable Datagram Protocol*. UDP poskytuje aplikace multiplexování (přes čísla portů) a ověření integrity (přes kontrolní součet) z hlavičky a těla. Hlasový a obrazový provoz bývá obvykle přenášen pomocí UDP. *Real-time video* a *audio streaming* protokoly jsou navrženy, aby manipulovali občasné ztracené pakety tak, že jen mírná degradace kvality může dojít, spíše než velké zpoždění v případě, že ztrácené pakety byly přenášeny. Pokud spolehlivý přenos byl žádán, to se musí provádět v uživatelské aplikaci.

záhlaví UDP se skládá ze 4 polí. Použití dvou z nich je volitelná v IPv4 (ornžové pozadí v tabulce). V IPv6 pouze zdrojový port je volitelný. V tab. 2.9 **Source Port Number** – Toto pole určuje odesílající port, kdy by měly být smysluplné, a předpokládá se, že port na odpověď, bude – li potřeba. Není – li použit, pak by měl být nula. **Destination Port Number** – Toto pole označuje cílový port a je požadována. **Length** – 16bitové pole, které udává délku v bajtech celý datagram: hlavičku a data. Minimální délka je 8 bytů, protože to je délka hlavičky. Velikost pole obsahuje teoretický limit 65535 bajtů (8 bajtů header + 65527 bajtů dat) pro UDP datagramu. Praktický limit pro délku dat, která je uložena v základní protokol IPv4, je 65507 bytů. **Checksum** – 16bitový kontrolní součet pole se používá pro kontrolu chyb hlavičky a dat. Algoritmus pro výpočet kontrolního součtu je různá pro dopravu přes IPv4 a IPv6. Je–li *checksum* vynechán v IPv4, pole používá hodnotu *all-zeros*. Toto pole není dostupné u IPv6.

Tab. 2. 9: Struktura UDP paketu.

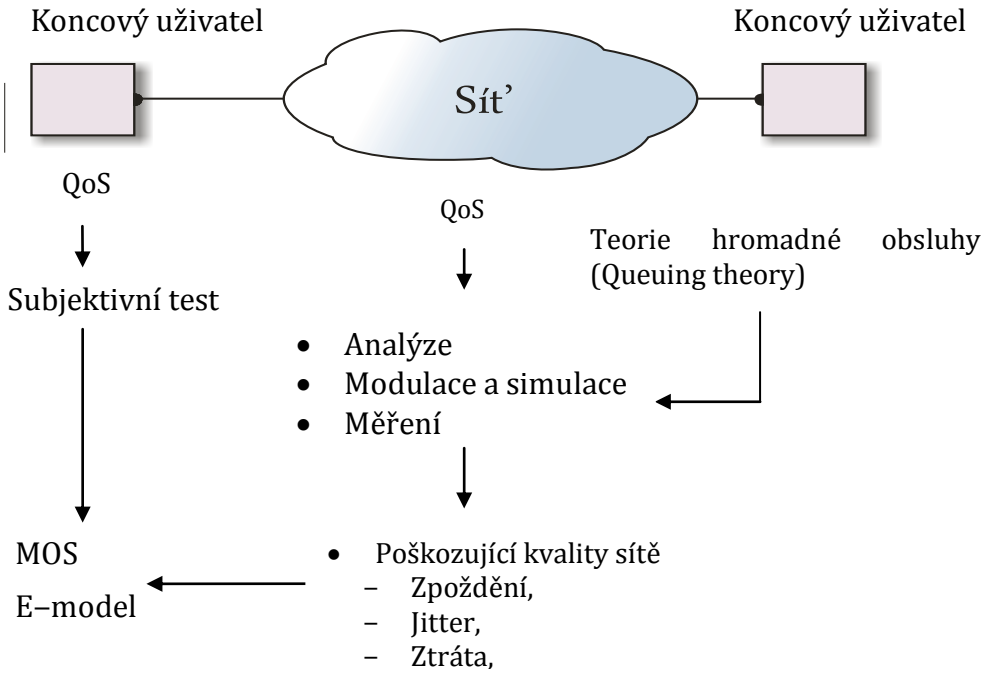
Bity	0-15	16-31
0	**Source Port Number**	**Destination Port Number**
32	**Length**	**Checksum**
64	**Data**	

3 QUALITY OF SERVICE

Quality of Service (dále jen QoS) může být chápána jako schopnost pakety cestující v síti s různými prioritami seřadit podle důležitosti. Jinak řečeno zajišťování určité úrovně výkonnosti, aby citlivé a důležité informace dorazili včas a s jistotou. Protokoly pro QoS pracují na zpomalení nedůležité pakety a v případě extrémního zatížení sítě je zcela vyhazuje. Zařízení, jenž vnímá priority jednotlivých paketů a na základě toho je zařadí do fronty, je známé pod jménem router. Nicméně se v této zprávě nebude zabývat o tom, co se v aktivních prvcích (router, switch, bridg, atd.) odehrává. Proto je tato situace reprezentována mrakem viz obr. 1.1.

3.1 Definice QoS

QoS může být definována jednak z hlediska QoS zkušenou koncovým uživatelem (např. IP telefonie, počítač nebo jiné komunikační (terminální) zařízení) a jednak z hlediska sítě. Jak je ilustrováno na obr. 3.1, který ukazuje tzv. end – to – end spojení, koncový uživatel může být zároveň i člověk, kdo toto terminální zařízení použije. Dále QoS, z pohledu koncového uživatele, je koncovým uživatelem vnímána jako kvalitu, kterou koncový uživatel dostává od sítě ISP pro danou službu nebo aplikaci, ke které se uživatel hlásil. Na druhé straně QoS z pohledu sítě odkazuje na schopnost sítě poskytovat určitou úroveň kvality. QoS je definována několika způsoby a neexistuje standardní definice. Vždy to záleží na kde, jak a proč jí užívat. Kombinací těchto všech definic je opravdu nejlepší definice. Technicky QoS je sadou technik k tomu, aby řídily Šířka pásma, zpoždění, jitter, a ztráta paketů v síti. Je možné manipulovat jeden nebo je všechny.



Obr. 3. 1: Definice QoS.

3.2 Významná role QoS

Na začátku telekomunikací byly v podstatě dvě samostatné sítě, jedna pro hlas a jedna pro data. Každá z nich byla s jednoduchým cílem zaměřena na přenosu určitého typu informací. Starší byla telefonická, která byla zavedena s vynálezem telefonu. Tato síť byla navržena, aby přenášela hlas. Přičemž se v telefonní sítě nacházejí jednoduchá zařízení (klasický telefon) připojena pomocí kabelu RJ12 a použijí starou technologii podporovanou v GSM sítích od prvních návrhů standardu – *Circuit Swithed Network* (Okruhově spínaná doména). Na druhé straně IP síť byla určena pro přenos data.

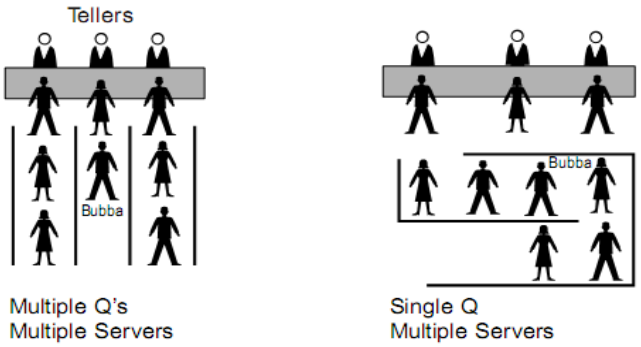
Telefonická komunikace, v průběhu celého období komunikace, je vyhrazená pouze jednomu hovoru. Jakmile je hovor ukončen, Obvody (*Circuit Swithed Network*) budou připraveny k vytvoření dalšího hovoru. Kvalitu hlasu závisí na kvalitu přenosu v sítě *end–to–end* během hovoru; jako je přenos ztráty, obvod šumu, echo, atd. Proto byla původní telefonická síť navržena tak, aby „poškozující parametry sítě“ byly přijatelné. Hlas byl (a stále je) komunikační služba v reálném čase (real time service). V telefonní síti nebyly žádné fronty pro ukládání zvukové signály pro pozdější doručení.

IP síť byla úplně jiný typ než síť telefonická. Za prvé, byla určená pro přenos data, která byla (a stále je většinou) na rozdíl od hlasu služba v nereálném čase (non–real time service). Navíc data mohou být uložena v síti pro pozdější doručení. Byly-li informace doručené s chybou, může být znovu odeslány. Datové služby byly někdy označované jako *"store–and–forward"* služby. Protože dřívější IP síť přenášela v podstatě jenom jeden druh informace, může být síť určena k provozu v módu tzv. *Best effort*, umožňující zpracovávat všechny pakety vyrovnaně. Vývojáři proto začali vymyslet novou síť, která by mohla integrovat oba typy služby (real and non–real time services).

V polovině devatenácté století tyto dvě samostatné sítě začali sjednotit pod termínu voice and data convergence [1]. V současné době takovouto síť představují mimo jiné systémy WiMAX, které musí splňovat požadavky na přenos dat, hlasu VoIP, videokonference, apod., s podporou multifunkčních služeb a řízením QoS. WiMAX podporuje protokoly IPv4, IPv6, ATM, Ethernet, apod. V konvergované síti, nejlepší úsilí (Best effort) není ta správná volba pro splnění různých požadavků jejích aplikací nebo služeb i když to může být vhodné pro tradiční internetových aplikací např. přenosy souborů, prohlížení webových stránek, FTP, e–mail. QoS je prostě technologie, která poskytuje řešení tohoto technického problému.

Pro zdůraznění důležitosti QoS, je využít obrázek 3.2 ukazující dva případy třeba v bance – viz [9]. Každý bankovní úředník má svoji vlastní linku z lidí čekající na to, aby mluvili s úředníkem a sjednali své obchody. Smůlu však má ten, který přijde za někým ve frontě, kdo je opravdu pomalý (tomu říká „Bubba“). Takže pokud by někdo přišel později a čekal na jiné frontě, by asi možná své jednání skončil dřív než ten stojící za „Bubbou“. Ve velkoměstech, kde jsou lidi ve větším spěchu, takový systém by asi moc nefungoval. Jak ukazovala teorie hromadné obsluhy; průměrný čas je pro jednu frontu a více úředníků menší než jedna fronta

pro každého úředníka. Z toho vyplývá, že nemá tato pomalá osoba (Bubba) žádný vliv na rychlosti fronty. Z prvního pohledu je efektivní využít jednu frontu pro více úředníky, má to ale jeden háček. Při vyžívání více front pro více úředníky, je možné, aby někdo přišel poslední a vyšel první (užitečné i když zdá nespravedlivé), což se při používání jednu frontu pro více úředníky stát nelze.



Obr. 3. 2: Porovnání rychlosti fronty při prezenci „Bubba“ [9].

V networking není vždycky vhodné použít metodu „jednu frontu pro více úředníky“. Naopak je důležité pro některé aplikace používající internetovou linku pro komunikaci mezi dvěma koncovými uživateli, aby jejich přeneseným paketům přístup umožňoval před pakety jiných aplikací. Požadovaná kvalita přenosu je uvažována ve velkých polí konceptů a v nástrojích, které mohou být použité k ovlivněním přístupu k paketu nějaké aplikace z hlediska služby. Myšlenka může být přeskupením výstupní fronty tak, že jeden paket dostává lepší službu než druhý. Požadovaná kvalita může být také ovlivněna kompresí přenášeného data, shaping nebo policing. Ať jde o kterýkoli mechanismus (způsob), snaha spočívá právě v tom, které pakety dostávají přednost a které ne. Právě jako v bance, provádění požadované kvality muže být pomocí "zařízené spravedlivosti" a zároveň i "nespravedlivosti" – upřednostňovat jeden paket před druhým. Přidáním větší Bandwdith k existujícímu spojení může být řešení, je ale finančně nevhodné.

Čtyři parametry v síťové dopravě můžou ovlivnit požadovanou kvalitu: Šířka pásma, zpoždění, Jitter a ztráta paketů. Tab. 3.1 popisuje některá chování síťového provozu bez zajištění QoS.

Tab. 3. 1: Síťový provoz bez realizace QoS[9].

Typ síťová provoz	**Chování sítě**
Zvuková	Hlas je těžko rozumět.
	Hlas rozbíjí, zní zčeřené.
	Díky zpoždění; interference hlasu volajících.
	Volání jsou odpojená.
Datová	Data přijíždí poté, co není již déle užitečné.
	Kolísavá časová odezva zklame uživatele, jenž se buď vzdá, nebo zkusí jindy.

3.3 QoS pro Real Time Services

Uživatelské hodnocení QoS v *real time services* (reálném čase služeb) se určuje subjektivními testy. Parametr, který může ovlivňovat kvalitu v real time services , může být jeden nebo kombinace z: Kvantizační šum, Poměr bitové chyby, **Zpoždění**, **Jitter**, **Ztrátovost**, **Volba kodeku**, Echo ovládání, Design sítě a další. Nicméně v této zprávě pozornost bude věnována především tučně označeným parametrům, které jsou vysoce navzájem závislé. Tato závislost bohužel vede k tomu, při zlepšení jednoho z nich může být důsledkem ke zhoršení druhého.

3.3.1 Zpoždění

Zpoždění postihuje především služby v reálném čase, jako jsou hlasové nebo obrazové. Jednosměrné zpoždění je množství času měřené od okamžiku, kdy mluvčí odnáší zvuk, do okamžiku slyšení tohoto zvuku od přijímací strany. Obousměrné zpoždění je pochopitelně součet těchto dvou jednosměrných zpoždění. K hlavním důvodům zpoždění patří: *Zdrojové kódování* – Zpoždění kvůli A/D a D/A převodu nebo Frame zpoždění, *Packetization zpoždění*, *Kanálové kódování* – Detekce a oprava chyb nebo Prokládání, *Jitter buffer zpoždění Propagation* zpoždění a další. Ke zpoždění způsobená zdrojovým kódováním, packetization, kanálovém kódováním nebo jitter buffer přispěje použitý kodek. Síť přispěje ke zpoždění způsobená *Propagation* zpoždění, které definuje dobu, kterou jeden bit potřebuje, aby se dostal z jednoho konce přes medium na druhém konci. Když je elektrický nebo optický signál umístěn na kabelu, energie nemůže propagovat do druhého konce kabelu okamžitě – K určitým zpožděním dojde. Rychlost energie na elektrických a optických rozhraní se blíží k rychlosti světla. Pro takové zpoždění bohužel nedá nic dělat, protože jsem omezený zákony fyziky! Však jediná proměnna, která ovlivňuje *Propagation* zpoždění (šíření zpoždění) je délka kabelu.

3.3.2 Jitter

V počítačových sítích a převážně v sítích založených na protokolu IP jako je Internet, jitter znamená kolísání velikosti zpoždění paketů při průchodu sítí (vzniká např. na směrovačích jako důsledek změn rotování, chování interních front routeru atd.). Lze tak všeobecně tvrdit, že jitter postihuje každé po sobě jdoucích paketů mající různé množství zpoždění. V jakékoli sítě použijící pakety s komponenty proměnného zpoždění je jitter vždycky prezenční. Otázkou je, zda uplatňovaný jitter natlačí dost, aby degradoval dané služby. Nicméně, některé provozy, jako jsou digitalizovaný hlas, vyžaduje, aby se pakety předávají v určitém jednotném „rytmu“ (např. každých 20 ms), tj. pakety by měly dorazit na místo doručení se stejnou časovou vzdáleností mezi nimi – izochronní provoz.

Nejlepší řešení pro problémy jitteru je větší šířka pásma! Jelikož to pomůže ke snížení zpoždění, a protože jitter je kolísání zpoždění, jitter potom bude menší. Například, jestliže zpoždění bylo v průměru mezi 100 ms a 200 ms, jitter by typicky byl až 100 ms. Pokud je zpoždění sníženo na hodnotu mezi 50 ms a 100 ms přidáváním větší šířku pásma, může být typické jitter snižován až na 50 ms.

3.3.3 Ztrátovost

Pro signály v reálném čase, např. hlas a video, ztrátovost (nebo také ztráta paketů) se projevuje jako šum v dekódovaném signálu, což vede k hlasovému výstřižku nebo poskoku, snižuje srozumitelnost řeči a kvalitu videa. Uvedeny jsou tady některé z hlavních zdrojů ztrátovosti: Bitová chybovost kvůli přenosu horšících parametrů vedení, např. "slábnutí", obvodový šum, Kolize paktu, Chyby zpracování, Buffer overflows, náhodný paket odkládání a další.

3.3.4 Subjektivní test

Subjektivní test se provádí dvěma způsoby:

1. **Mean Opinion Score (MOS):**

   V multimédia, zvláště když jsou kodeky použity ke kompresi požadavky na šířka pásma, MOS stanoví číselné označení vnímanou kvalitu obdržených médií po kompresi či přenosu. MOS se vyjadřuje jako jedno číslo v rozmezí 1 až 5, kde je 1 nejnižší vnímanou kvalitou zvuku a 5 je nejvyšší vnímanou kvalitou zvuku. MOS testy pro přenos hlasu jsou specifikované od ITU-T v doporučení P.800. MOS je generován průměrování výsledků, kde několika posluchačům pustí zkušební hlas a oni hodnotí kvalitu slyšeného hlasu standardními větami určené pomocí následující tabulky:

   Tab. 3. 2: Stanovená tabulka pro MOS testy.

MOS	Kvalita	Zhoršení
5	Výborná	Nepatrný,
4	Dobrá	Patrný, ale ne rušící,
3	Středná	Mírně rušící,
2	Slabá	Rušící,
1	Špatná	Velmi rušící.

$$MOS = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{Š} \times 1)}{N_c} \quad (3.1)$$

$$N_c = N_V + N_D + N_{St} + N_{Sl} + N_{Š} \quad (3.2)$$

Přičemž N je počet lidí, jeho index je první písmeno vyhodnocované kvality a N_c je celkový součet počtu lidí.

2. **E–modelem:**

   E–model je definován od ITU-T v doporučení G. 107. Jedná se o výpočetní model určený k produkci MOS, aniž by provedl subjektivní testování. Subjektivní testování je nákladné a časově náročné. E–model je výpočetní model, kterým lze nahradit subjektivní testování. Při použití E–modelu, vliv zpoždění, jitter, ztráta paketů, a další postižení parametry jsou sloučeny do jediného objektivního parametru tzv. R–faktor, který pohybuje od 0 do 100.

3.4 Mechanizmy pro QoS

Vzhledem k zadání této bakalářské zprávy, mechanizmy pro QoS se v dané laboratoři nezahrnují a budou v dalších podkapitolách velmi stručně proprány.

3.4.1 Služba nejlepšího úsilí (Best effort)

V tomto modelu posílají aplikace data, když se jim zachce, kolik chtějí a bez vyžádání si jakéhokoliv povolení. Síťové komponenty se pokouší přenést data co nejlépe, bez omezení na zpoždění, zpoždění reakce (latency), nebo rozptyl zpoždění. Dělají to i tehdy, když nemohou data doručit, bez informování odesílatele nebo příjemce. Příkladem takovéto služby je doručování v IP sítích [3],[4].

3.4.2 Integrované služby (Integrated Services – IntServ)

V případě integrovaných služeb aplikace oznámí počítačové síti své požadavky na přenos dat ve formě požadovaných QoS. Počítačová síť ověří, zda jsou k dispozici požadované prostředky, a rozhodne, zda požadavkům vyhoví. Tato funkce je označována jako admission control (řízení přístupu). V případě, že síť nemůže požadavku vyhovět, není spojení povoleno a aplikace se může rozhodnout, zda požádá o méně náročné QoS. Pokud je požadavek přijat, musí počítačová síť informovat všechny komponenty, přes které bude probíhat přenos, aby pro dané spojení rezervovaly odpovídající objem prostředků, např. šířku pásma mezi dvěma směrovači, kapacitu fronty paketů, atd. K tomuto účelu slouží rezervační protokoly. Nejrozšířenějším rezervačním protokolem je RSVP (*Resource reSerVation Protocol*), který je však poměrně složitý a představuje významnou režii při řízení chodu sítě. Proto se v poslední době objevují návrhy jednodušších protokolů pro rezervaci, např. YESSIR.

Možnost specifikovat přesně QoS není u všech aplikací nutná. Řada aplikací vystačí s tím, že požadované parametry se podstatně nezhorší při změně zatížení počítačové sítě. Navíc se zvyšujícím se objemu přenášené informace je třeba snížit režii při přepínání a minimalizovat objem stavové informace ve směrovačích. Proto se v poslední době objevuje jiný způsob implementace QoS – rozlišované služby (*Differentiated services* – diffserv).

IntServ vychází z modelu, kdy je před přenosem zajištěna potřebná kvalita přenosového kanálu. K tomu slouží RSVP (*Ressource reSerVation Protocol*). Pro řízení sítě se používají následující strategie: Udržování stavu propojení, Hlídání a úprava přenosu, Předcházení zahlcení, Management předcházení nebo odstranění zahlcení a Mechanizmus sledování výkonnosti linky. Integrované služby rozlišují mezi následujícími kategoriemi aplikací:

Elastické aplikace – bez požadavku na doručování. Do této kategorie zapadají aplikace nad TCP. Nejsou kladeny požadavky na omezení zpoždění nebo kapacitu spojení. Příkladem je el. Pošta, http protokol, atd.

Real–Time Tolerant (RTT) aplikace – požadují omezení na maximální zpoždění v síti. Občasná ztráta paketů je přijatelná. Příkladem jsou video aplikace využívající

bufferování, které před aplikací skryjí ztrátu paketů.

Real–Time Intolerant (RTI) aplikace – tato třída požaduje minimální odezvu (latency) a rozptyl zpoždění (*jitter*). Příkladem jsou videokonference.

K zajištění obsluhy těchto aplikací má RSVP k dispozici následující třídy služeb *Class of Service* (CoS): *Garanteed Service* – služba je určena pro RTI aplikace a zaručuje: Šířku pásma pro přenos v rámci aplikace a Deterministickou horní hranici zpoždění. To je důležité pro interaktivní aplikace nebo aplikace v reálném čase. Aplikace mohou snížit zpoždění zvýšením požadavků na šířku pásma. *Controlled Load Service* – je určena pro RTT aplikace. Zaručuje průměrné zpoždění, ale zpoždění přenosu jednoho paketu mezi koncovými uzly není deterministické.

K zajištění IntServ se mohou použít různé protokoly. V současné době jsou rozpracovány *ReSerVation Protocol* (RSVP) od IETF a *Common Open Policy Service* (COPS), který navrhlo CISCO. RSVP slouží k přenosu rezervačních požadavků a vytváří virtuální okruhy s danými přenosovými parametry. COPS slouží k řízení možností rezervace, které mají jednotlivé směrovače. Čili říká směrovačům, které požadavky přijmout a které ne.

3.4.3 Rozlišované služby (*Differentiated services* – diffserv)

Rozlišované služby se od integrovaných služeb liší zejména tím, že aplikace neoznamuje předem počítačové síti své požadavky na QoS. Použití rezervačních protokolů není nutné. Jednotlivé směrovače neudržují žádnou stavovou informaci o jednotlivých spojeních. Implementace QoS je řešena tak, že každý paket vstupující do počítačové sítě je označen značkou, která určuje třídu přenosu, poskytovanou paketu. Označování paketů probíhá pouze na vstupu do počítačové sítě, během přenosu pouze směrovače čtou značku a podle této značky řídí způsob zpracování paketu. Počet značek je poměrně malý, maximálně desítky.

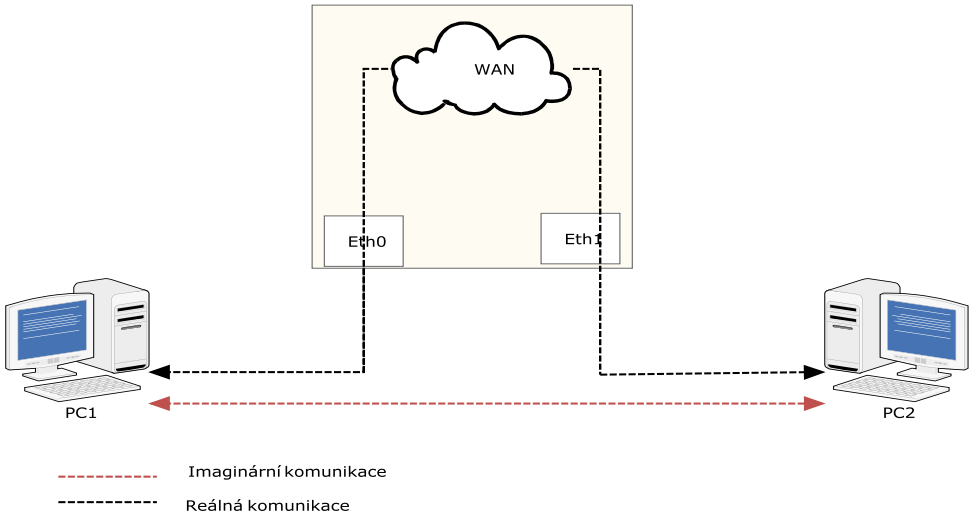
U integrovaných služeb udržuje každý směrovač stavovou informaci vztaženou ke každému spojení, u rozlišovaných služeb směrovače pouze přidělí určené prostředky každé třídě přenosu a zajišťují určitý vztah mezi třídami.

4 LABORATOŘ KVALITY SLUŽBY

Cílem laboratoře kvality služby je sestrojit vhodnou konstrukci (virtuálně postavena), pomocí níž se dá řídit ztrátovost, jitter a zpoždění. Dále aby byla sestavena ze dvou IP telefonů, softwaru schopného reprodukovat audio nahrávku jako VoIP hovor a SIP telefonní ústřednu. Po zjištění správných konfigurací uvedených dílčích, prakticky realizovat laboratoř. Navržené řešení by mělo umožnit subjektivní testy (MOS) kvality přenášeného zvuku, tedy nastavovat ztrátovost a explicitně volit kodeky a jejich parametry.

4.1 Konstrukce laboratoře

K tomuto účelu budou používány čtyři dílče; Dva virtuální IP telefony (*x-lite*), jednotka (WANem), přes kterou naladíme parametry QoS – ztrátovost, jitter a zpoždění. Poslední dílče bude SIP ústředna. Pro tento účel byla vybrána 3CX Phone System for Windows od 3CX Ltd. Princip uskutečňování hovoru je zobrazen na obr. 5.1. Oba koncové uživatele budou připojeni přes WANem. V tomto případě počítač, na kterém je instalován WANem, bude mít dvě aktivované síťové karty. To znamená konec každého hostitele je připojen k jedné kartě rozhraní WANem. Jinými slovy oba jsou připojené k zapnutí LAN. Čili PC1 a WANem jsou připojeny k LAN1 dále WANem a PC2 jsou připojeny k LAN2.



Obr. 4. 1: Představa probíhající komunikace.

Řídicí jednotka WANem (*Wide Area Network emulator*) vydaný od PERC (*Performance Engineering Research Center*) je v podstatě doplňující program běžící na *Linux–Debian* platformě, která může napodobovat skutečné sítě v rozvoji nebo testovacím prostředí. Hostitelský program, na kterém běží všechny tyto dílče bude *VMLite Workstation*. VMLite vytváří virtuální prostředí na HOST (počítač, na kterém je instalován). V tomto prostředí pak lze vybudovat další užitečná zařízení nebo počítače. Tímto způsobem lze pak celou laboratoř prostě vybudovat na jednom počítači.

Po instalaci WANem do samostatného zařízení a SIP ústředny do jednoho z uvedených počítačů, provede se nastavení jednotlivých zařízení, programů nebo jednotek následujícími postupy:

1. Na oba testovací zařízení (počítače) se nainstaluje operační systém (v tomto experimentu XP), softwarový telefon (*x-lite*) a SIP ústřednu (stačí pouze na jednom počítači). Dále se zvolí jeden z těchto počítačů a na něj se instaluje program *Wireshark*, který umožňuje zaznamenat (*capture*) odesílací pakety. To umožní uživateli zpoždění a ztrátovost paketů měřit nebo graficky zobrazit jitter.

   Zařízení budou připojeny k jednotce WANem tak, že první zařízení s WANem-eth1 vytvoří LAN1 a druhé zařízení s WANem-eth2 vytvoří LAN2. Tímto způsobem testovací pakety z prvního konce dorazí do druhého přes WANem. Existují však i jiné varianty. Zvolené konfigurace síťových karet jednotlivých zařízení byla zapisované do tab. 5.1.

Tab. 4. 1: Konfigurace síťových karet testovacích zařízení.

Zařízení	IP adresa	Netmask	Gateway
PC1	168.254.103.165	255.255.0.0	—
PC2	168.254.2.64	255.255.0.0	—

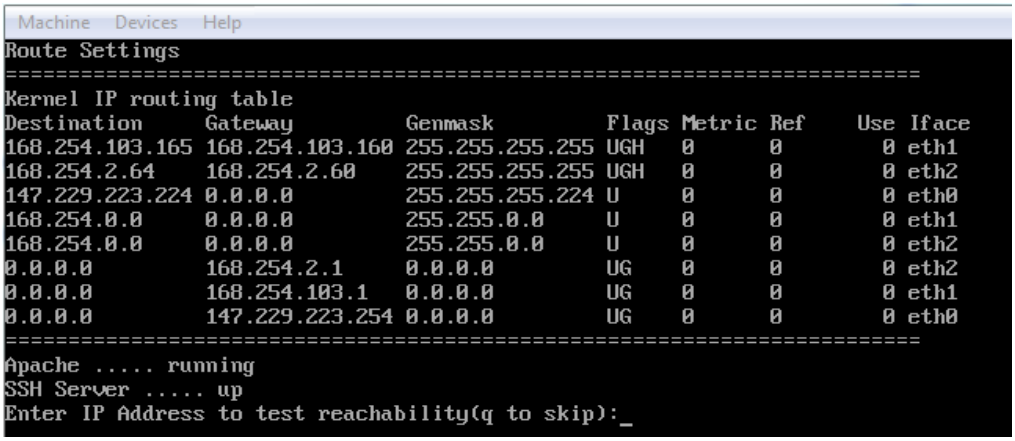
2. Přidělí se jednotce WANem dvě síťové karty (eth1 a th2), každá z nich bude logicky umístěna na samostatné lokální síti (*Intranet*). Potom se bude přidělena jednu externí kartu (eth0), přes kterou se naladí QoS parametry, tj. ztrátovost, jitter, atd. Konfigurace této karty provede program automaticky pomocí DHCP (*Dynamic Host* Configuration *Protocol*). Další konfigurace jsou vylistovány do tab. 5.2.

Tab. 4. 2: Konfigurace sítě.

Síťová karta	IP adresa	Netmask	Gateway
eth1	168.254.103.160	255.255.0.0	168.254.103.1
eth2	168.254.2.60	255.255.0.0	168.254.2.1

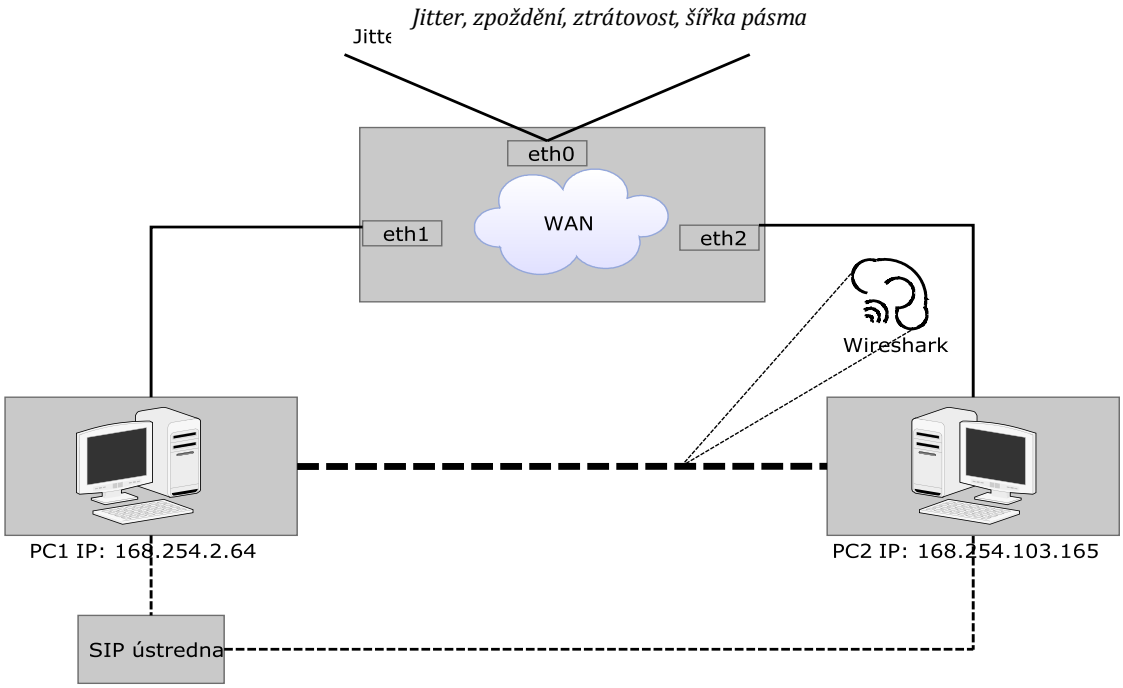
Po správném nastavení začíná program uvedené karty zprovozňovat. Potom bude zapotřebí si připisovat (pomocí příkazem assign) testovací zařízení do příslušné karty. Byla zvolena eth1 počítači−1 (běžící na VMLite XPMode) a eth2 počítači−2 (běžící na VMLite XPMode1). Na základě toho se potom sestrojí routovací tabulka, pomocí které se nastaví síťový provoz (*network traffic*) viz obr. 5.1. Nejdříve na počítači PC1 se orientují pakety z tohoto počítače do WANem pomocí příkazu *route add*. Udělá se taktéž i na

druhém počítači. Kontrola správného připojení se provádí promocí příkazu *Ping* nebo i *tracert*.



Obr. 4. 2: Routovací tabulka.

Obr. 5.2 zobrazuje přehlednější představu konstrukce konkrétní laboratoře. Přičemž diskrétní čára reprezentuje logické spojení mezi jednotlivými zařízeními. PC2 se SIP ústřednou komunikuje přes PC1 síťovou kartou. Jitter, zpoždění a další QoS parametry se nastaví dálkovým počítačem přes eth0. WANem aplikuje tyto nastavení na odesílací pakety pomocí tzv. *Graphical User Interface* (USI).



Obr. 4. 3: Zvolená konstrukce laboratoře kvality služby.

Na dálkovém počítači se otevře *Internet Explorer*, *Firefox* nebo jiný funkčně obdobný program a se zadá adresu URL http://wanemip/WANem. V tomto případě *wanemip* bude IP eth0. Ukáže se zmíněné GUI s pěti možnostmi (options) viz E příloha. Dále po zahájení hovoru se změří ztrátovost, maximální a minimální jitter a typ payloadu tedy použití kodeků. Tyto kodeky jsou zprostředkovány

programem *x-lite*. Čtyři kodeky byly v tomto experimentu vyšetřované (G. 729, G. 711*a*, G. 711*u* a GSM). Spojení bylo navázáno na maximální dobu cca 1 min.. Měření se provádělo skrze tři scénáře. Nicméně se konstrukce laboratoře ve všech scénářích byla zachována. Dva parametry však byly během měření proměnné; šířka pásma a zpoždění. O vyhodnocované kvalitě přenášeného hlasu na základě subjektivního testu — MOS bude řeč až v závěru.

4.2 První scénář

V tomto scénáři byla položka šířky pásma v UGI nastavena na hodnotu 1.544 Mbps. Pro položku zpoždění byly zvolené tří hodnoty následovně: 10ms, 50ms a 150ms.

1. **Zpoždění o cca 10ms.**

Payload	Počet paketů	Ztrátovost	Max. jitter	Min. jitter
G. 711 alaw	3506	0	21,35	15,27
G. 711 ulaw	3400	0	36,2	15,46
G. 729	3602	0	24,37	14,26
GSM	3410	0	34,15	15,9

2. **Zpoždění o cca 50ms.**

Payload	Počet paketů	Ztrátovost	Max. jitter	Min. jitter
G. 711 alaw	4494	0	47,28	29,63
G. 711 ulaw	4528	0	36,42	28,6
G. 729	3401	0	29,15	23,32
GSM	3508	0	39,59	30

3. **Zpoždění o cca 150ms.**

Payload	Počet paketů	Ztrátovost	Max. jitter	Min. jitter
G. 711 alaw	5188	0	105,68	79,54
G. 711 ulaw	4410	0	110	69,40
G. 729	4360	0	100,55	68,38
GSM	4730	0	106,19	88,39

4. **Subjektivní test**, šířka pásma 1,544 Mbps a zpoždění cca 150 *ms:*

$$MOS_{G.711a} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(4\times3)+(4\times2)+(1\times1)}{9} = \boxed{2{,}333}$$

$$MOS_{G.711u} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(2\times3)+(6\times2)+(1\times1)}{9} = \boxed{2{,}111}$$

$$MOS_{G.729} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(1\times3)+(6\times2)+(3\times1)}{9} = \boxed{2}$$

$$MOS_{GSM} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(1\times3)+(4\times2)+(5\times1)}{9} = \boxed{1{,}778}$$

4.3 Druhý scénář

Položka zpoždění byla v tomto scénáři nastavena na hodnotu 10ms. Pro položku šířky pásma byly zvolené tří hodnoty následovně: 10 Mbps a 100 Mbps.

1. **Šířka pásma 10 Mbps.**

Payload	Počet paketů	Ztrátovost	Max. jitter	Min. jitter
G. 711 alaw	4556	0	18,26	12,37
G. 711 ulaw	4602	0	18,35	12,21
G. 729	4556	0	35,82	12,31
GSM	4478	0	42,34	12,31

2. **Šířka pásma 100 Mbps.**

Payload	Počet paketů	Ztrátovost	Max. jitter	Min. jitter
G. 711 alaw	4498	0	22,9	9,29
G. 711 ulaw	4558	0	30,13	10,47
G. 729	4532	0	24,76	11,96
GSM	4566	0	32,16	11,91

3. **Subjektivní test,** šířka pásma 100 Mbps a zpoždění cca 10 ms:

$$MOS_{G.711a} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(7\times5)+(2\times4)}{9} = \boxed{4{,}778}$$

$$MOS_{G.711u} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(7\times5)+(2\times4)}{9} = \boxed{4{,}778}$$

$$MOS_{G.729} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(6\times5)+(3\times4)}{9} = \boxed{4{,}667}$$

$$MOS_{GSM} = \frac{(N_V \times 5)+(N_D \times 4)+(N_{St} \times 3)+(N_{Sl} \times 2)+(N_{\check{S}} \times 1)}{N_c} = \frac{(6\times5)+(2\times4)+(1\times3)}{9} = \boxed{4{,}556}$$

5 ZÁVĚR

Záměr při zpracování této bakalářské práce je zde soustředěn na to, co se děje v datové síti, konkrétně IP síť, při využívání této sítě k přenosu hlasu. A to z hlediska kvality tohoto hlasu a spokojenosti uživatele provedením subjektivní test nebo tzv. MOS. Příčiny snížení kvality hlasu jsou celá řada. Některé však jsou uvedené v podkapitole 3.3. Z těchto příčin byly zkoumány v laboratoři této bakalářské práce hlavně ztrátovost, použití kodeků, zpoždění, atd. Snížení kvality digitalizovaného signálu (hlas) v IP síti se vyskytuje často ve dvou směrech. Buď v algoritmu kodeku, při převodu z A/D nebo zpět, anebo se nastanou, kvůli ztraceným datagramům v rámci IP sítě. Kodeky s nízkou kódovací rychlosti zhorší kvalitu akustického signálu mnohem více, než vysokorychlostní kodeky, protože provádějí kompresi signálu se ztrátovou kompresí.

Podle praktického měření provedení uvedenými softwary, zkušební síť by se mohla udržovat v provozu a spojení bylo víceméně navázáno bez kritických ztrátovostí užitečných paketů, při poskytování šířky pásma cca 1 Mbps se zpožděním nepřesahujícím cca 180 ms. Však takové hodnoty by vůbec nevyhovovali službám v reálném čase, tedy VoIP. V přílohách A, B a C byly provedeny testy sítě s různými kombinacemi šířky pásma a zpoždění. Působení zpoždění na QoS je horší než působení šířky pásma z hodně důvodů. Totiž je vícestranní parametr a vysoce závislé na jiné parametry, které svým výskytem se zhorší kvalita přenosu. Jako třeba jitter.

Dle subjektivního testu viz podkapitole 5.3 a 5.4, přenášený hlas přes sítě (alespoň tu v uvedeném experimentu) s šířkou pásma 1,544 Mbps a zpoždění cca 150 *ms* by byl docela rušící a při zatížení sítě vůbec nepoznaný. Vyhovovala by požadavkům síť s šířkou pásma cca 10 Mbps a zpoždění do 30 *ms*. Tyto změřené hodnoty jsou však hrubé a je třeba uvažovat nedokonalost použité stroje a omezená schopnost počítače při zpracování velké množství dat, což vede k dalšímu typu zpoždění.

LITERATURA

[1] Referenční model ISO/OSI, *www.wikipedia.org*.

[2] Kun, I., Park, *QoS in Packet Networks*,The MITRE corporation USA, 2004. LESTARI, A.A., YAROVOY, A.G., LIGTHART, L.P. Numerical and experimental analysis of circular-end wire bow-tie antennas over a lossy ground. *IEEE Transactions on Antennas and Propagation*. 2003, vol. 52, no. 1, p. 26–35.

[3] Gen2 Ventures, *Clearing the Way for VoIP*, White Paper, www.gen2ventures.com

[4] Arindam Paul: QoS in Data Network: Protocols and Standards.

[5] Jan Kubr, Y36SPS QoS, 2008.

[6] DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.

[7] A Handbook for Successful VoIP Deployment: Network Testing, QoS, and More by John Q. Walker, *NetIQ Corporation*.

[8] Sven Ubik: QoS a diffserv – Úvod do problematiky, Technická zpráva TEN-155 CZ číslo6/2000.

[9] Wendell Odom, Michael J. Cavanaugh. Cisco QOS Exam Certification Guide, Second Edition Published by: Cisco Press ISBN1-58720-124-0.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADPCM – Adaptive differential pulse-code modulation

ARP – Address Resolution Protocol

ATM – Asynchronous Transfer Mode

CIDR – Classless Inter-Domain Routing

COPS – Common Open Policy Service

CoS – Class of Service

CSLIP – Compressed Serial Line Internet Protocol

DARPA– Defense Advanced Research Projects Agency

DCCP – Datagram Congestion Control Protocol

DF – Don't fragment

Diffserv – Differentiated services

DNS – Domain Name System

DSL – Digital Subscriber Line

FDDI – Fiber Distributed Data Interface

FTP – File Transfer Protocol

GSM – Global System for Mobile Communications

HTTP – Hypertext Transfer Protocol

IANA – Internet Assigned Numbers Authority

ICANN – Internet Corporation Assigned Numbers Authority

ICMP – Internet Control Message Protocol

IETF – Internet Engineering Task Force

IGMP – Internet Group Management Protocol

IMAP – Internet Message Access Protocol

IntServ – Integrated Services

IP – Internet Protocol

ISDN – Integrated Services Digital Network

ISO – standardizoval mezinárodní standardizační úřad

MF – More fragment

MIME – Multipurpose Internet Mail Extensions

MOS – Mean Opinion Score

NDP – Neighbor Discovery Protocol

OSI – Open System Interconnection

OSPF – Open Shortest Path First

PCM – Pulse-code modulation

PPP – Point-to-Point Protocol

PPTP – Point-to-Point Tunneling Protocol

QoS – Quality of Service

RFC – Request For Comments

RSVP – ReSerVation Protocol

RSVP – Resource reSerVation Protocol

RSVP –Ressource reSerVation Protocol

RTI – Real–Time Intolerant

RTP – Real Time Protocol

RTP – Real Time Protokol

RTS – Real Time Services

RTSP – Real Time Streaming Protocol

RTT – Real–Time Tolerant

SCTP – Stream Control Transmission Protocol

SCTP – Stream Control Transmission Protocol

SIP – Session Initiation Protocol

SLIP – Serial Line Internet Protocol

SSH – Secure Shell

SSL – Secure Socket Layer

TCP – Transmission Control Protocol

TLS – Transport Layer Security

ToS – Type of Service pole

TTL – Time To Live

UDP – User Datagram Protocol

VoIP– Voice over IP

WiMAX – Worldwide Interoperability for Microwave Access

WWW – World Wide Web

RTSP – Real – Time Streaming Protocol

WANem - Wide Area Network emulator

DHCP – Dynamic Host Configuration Protocol

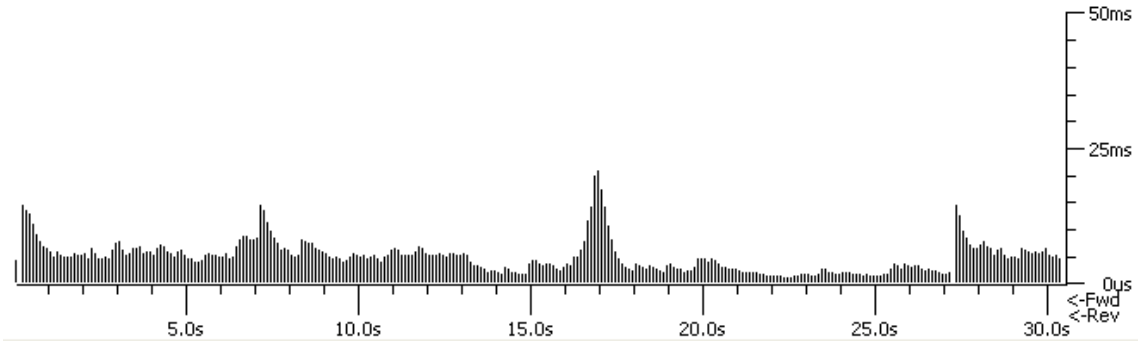
USI – Graphical User Interface

A PŘÍLOHA

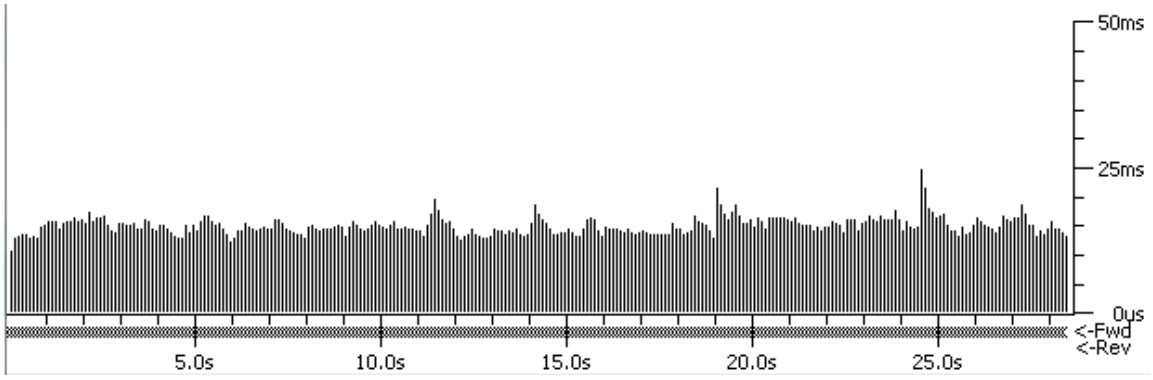
Nominální hodnoty pro danou síť, tj. při maximální přenosové rychlosti 100 Mbps se zpožděním menší než 1ms.

Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb?
ITU-T G.729	2044	0 (0.0%)	199.37	20.46	4.74	X
ITU-T G.729	2032	0 (0.0%)	58.55	10.86	3.64	X
BV32	2199	0 (0.0%)	0.00	0.00	0.00	X
BV32	2183	0 (0.0%)	0.00	0.00	0.00	X
ITU-T G.711 PCMU	2075	0 (0.0%)	83.76	8.65	2.17	X
ITU-T G.711 PCMU	2053	0 (0.0%)	51.93	6.71	2.85	X
ITU-T G.711 PCMA	2106	0 (0.0%)	122.60	15.48	2.94	X
ITU-T G.711 PCMA	2093	0 (0.0%)	62.10	10.26	3.26	X
ITU-T G.729	2083	0 (0.0%)	459.61	56.76	8.03	
ITU-T G.729	2106	0 (0.0%)	108.33	14.01	4.11	X
GSM 06.10	2713	0 (0.0%)	120.03	15.80	3.93	X
GSM 06.10	2694	0 (0.0%)	116.95	12.25	3.96	X

Jitter při použití G. 711 *a*law.(100 Mbps, <1ms).

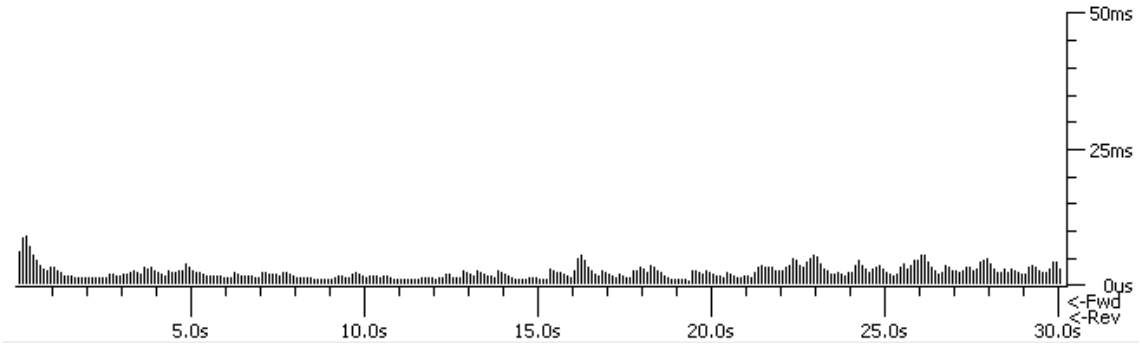


Jitter při použití G. 711 *a*law.(1,544 Mbps, cca 10ms).

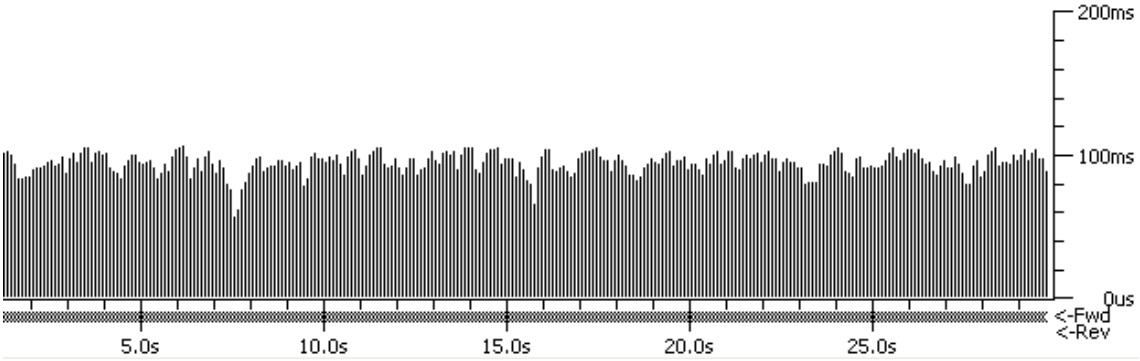


B PŘÍLOHA

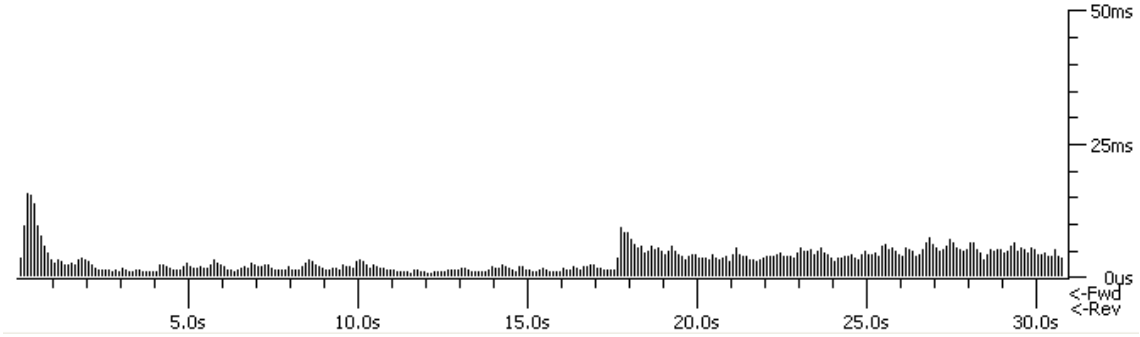
Jitter při použití G. 711 *u*law.(100 Mbps, <1ms).



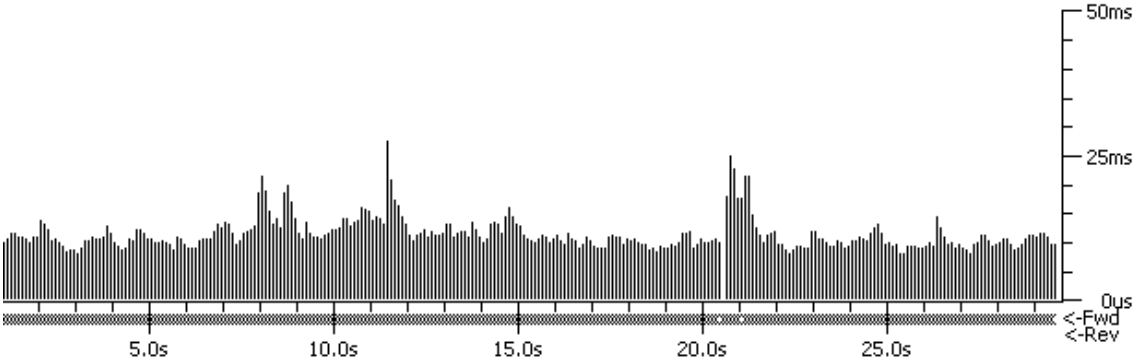
Jitter při použití G. 711 *u*law.(1,544 Mbps, cca 130ms).



Jitter při použití G. 729.(100 Mbps, <1ms).

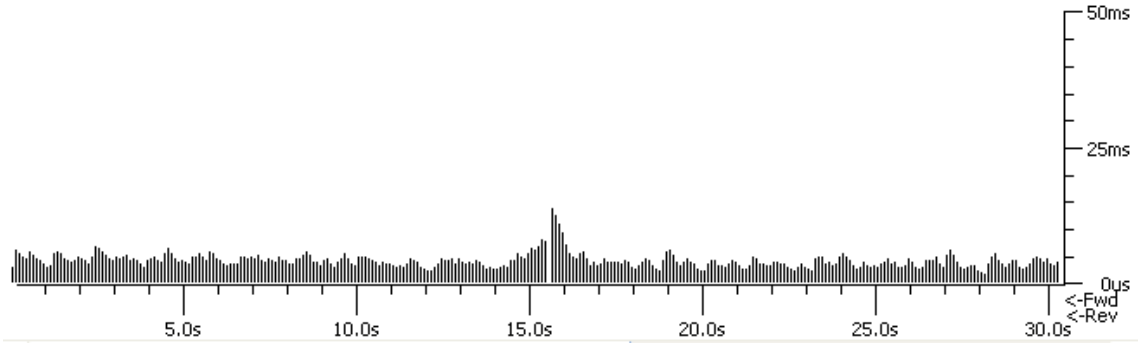


Jitter při použití G. 729.(10 Mbps, cca 10ms).

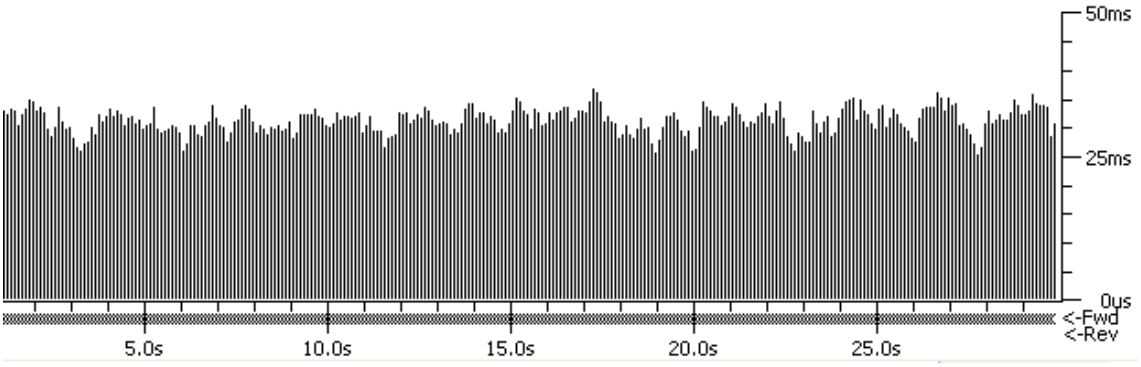


C PŘÍLOHA

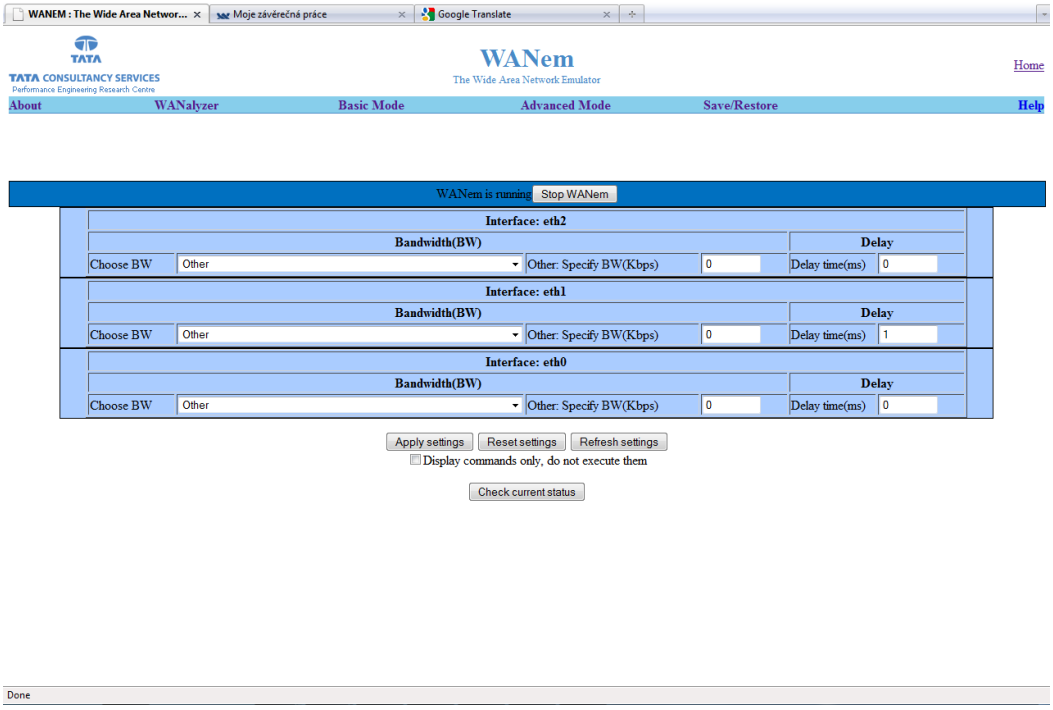
Jitter při použití GSM.(100 Mbps, <1ms).



Jitter při použití GSM. (1, 445 Mbps, 30ms).



Ukázka USI programu WANem.



D PŘÍLOHA

Přehlednější struktura záhlaví TCP paketu.

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Source port																Destination port															
32	Sequence number																															
64	Acknowledgment number																															
96	Data offset				Reserved				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
128	Checksum																Urgent pointer															
160 …	Options (if Data Offset > 5) …																															