Zajištění kvality služby (QoS) v operačním systému Windows

Teoretické možnosti aplikace mechanismů zabezpečení kvality služby (QoS) v nových verzích MS Windows a praktické ověření funkčnosti.

Richard Biječek, BIJ006
Martin Vrba, VRB050

27.6.2006

Úvod

V počítačových sítích s provozem aplikací náročných na datové toky a dobu odezvy, jako jsou IP telefonie, video konference apod. je potřeba zajistit dostatečnou průchodnost a zároveň spolehlivost. Za tímto účelem disponují operační systémy Windows (v námi testovaných verzích XP a 2003) nástroji pro rozlišování a upřednostňování datového toku v IP síti. Tyto verze (na rozdíl od Windows 2000) již nepodporují rezervaci síťového pásma pomocí protokolu RSVP.

Nástroje pro konfiguraci QoS zajišťují potřebné parametry sítě, jakými jsou šířka pásma, zpoždění, rozptyl zpoždění a ztrátovost paketů. Tyto parametry jsou zajišťovány jednak značením rámců a paketů na 2. a 3. vrstvě a také tvarováním provozu. Pro provoz QoS je nutné, aby použité standardy podporovala všechna síťová zařízení po cestě (koncové stanice, přepínače, směrovače).

Cílem práce je popsat možné postupy konfigurace a jejich výsledky.

Typy služeb

Datové toky jsou rozdělovány do standardizovaných kategorií, podle „typu služby“. Zde je jejich stručná charakterizace:

- Best effort – Standardní model přenosu dat v IP síti, bez zaručené spolehlivosti, zpoždění či jiných parametrů
- Controlled load – Data jsou přenášena podobně jako v kategorii Best effort při nezatížené síti. Pokud je síť přetížená, pak se parametry přenosu blíží stavu při nezatížené síti. To znamená že převážná část paketů je úspěšně doručena na cílovou stanici a zároveň převážná část paketů dorazí do cíle s minimální dobou odezvy. Tedy jedná se o přednostní přenos bez garance doby odezvy. (Přesná specifikace viz. RFC 2211)
- Guaranteed service – Jedná se o přednostní přenos s garantovanou dobou odezvy. Dopad této kategorie na síť je znatelný, proto se běžně využívá pouze pro přenosy, kladoucí velké požadavky na parametry sítě. (viz. RFC 2212)
- Network control – Nejvyšší kategorie, navržená pro provoz řízení sítí.
- Qualitative service – Je navržen pro aplikace, které vyžadují přednostní přenosy, ale nemohou vyčíslit přenosové požadavky. Typicky jsou to aplikace s přerušovaným či shlukovým provozem a aktuální stav sítě určuje jak jsou data přenesena.

QoS protokoly

Windows XP a 2003 podporují jak 802.1p (na 2. vrstvě), tak DSCP (na 3. vrstvě).

802.1p

Toto značení využívají přepínače (a případně ostatní síťové prvky na 2. vrstvě) za účelem rozdělení příchozích rámců do oddělených tříd. V případě zahlcení sítě, které může způsobit ztrátu rámců, tato zařízení přednostně zpracují (přenesou) rámce s vyšší prioritní třídou.

Na sítích typu Ethernet je označení priority přenášeno spolu s VLAN ID (802.1q), kde jsou pro tento účel vyhrazeny 3 bity. Jejich kombinací může být nadefinováno 8 prioritních tříd (0-7). Standardní přiřazení je uvedeno v tabulce 1.

Typ služby	Označení priority
Neoznačené pakety	0
Best effort	0
Controlled load	4
Guaranteed service	5
Network control	7
Qualitative service	0

Tabulka 1

Toto nastavení může být změněno systémovou politikou (viz dále), která ovlivňuje tento klíč v registrech: HKLM\Software\Policies\Microsoft\Windows\Psched\UserPriorityMapping

DSCP

DSCP je protokol definující priorizaci na 3. vrstvě. Tyto údaje jsou po cestě využívány směrovači (pokud DSCP podporují) k určení priority přeposílání. Údaj se nachází v IP hlavičce a má 8 bitů.

Původně byly první 3 bity používány pro prioritu, která vyjadřovala důležitost paketu, nyní se používá 6 bitů pro určení třídy. Hodnoty tříd jsou nastaveny tak, aby byly zpětně kompatibilní (co do obsahu prvních 3 bitů) se staršími směrovači. Standardní přiřazení je opět uvedeno v tabulce 2.

Typ služby	Označení priority
Best effort	0 (0x00)
Controlled load	24 (0x18)
Guaranteed service	40 (0x28)
Network control	48 (0x30)
Qualitative service	0 (0x00)

Tabulka 2

Pakety, které neodpovídají žádné položce („Filterspec“) ve filtru se označují prioritou 0.

Toto nastavení může být změněno systémovou politikou (viz dále), která ovlivňuje klíče v registrech v závislosti na tom, zda pakety odpovídají specifikaci toku resp. nikoli:
HKLM\Software\Policies\Microsoft\Windows\Psched\DiffservByteMappingConforming
HKLM\Software\Policies\Microsoft\Windows\Psched\DiffservByteMappingNonConforming

Architektura QoS ve Windows XP a 2003

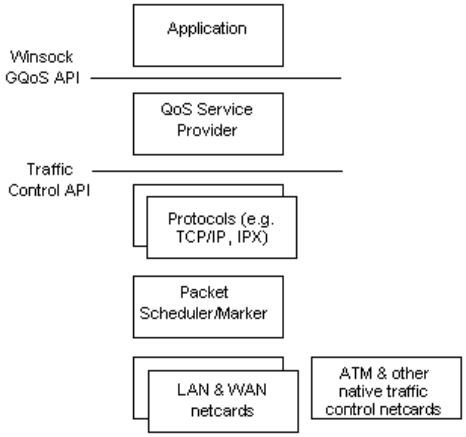
Kontrolu síťového provozu zajišťují 3 systémové komponenty: Traffic Control API (TC API), Generic Packet Classifier (GPC) a QoS Packet Scheduler (česky „Plánovač paketů technologie QoS“).

- Traffic control API (*traffic.dll*) rozděluje pakety na jednotlivé toky a přiřazuje jim příslušné parametry. Vývojáři aplikací nebo správci sítí používají TC API ke specifikaci přednostního provozu. S tímto jsou spojeny následující pojmy:
  - Flowspec – seznam definující QoS parametry (např. typ služby, zpoždění atd.), které se přiřazují datovým tokům.
  - Flow – tok (tvořený sekvencí paketů) podléhající konkrétnímu seznamu parametrů „Flowspec“. Všechny pakety v toku jsou zpracovány stejně
  - Filterspec – položka filtru, sestávající se ze zdrojových a cílových adres a portů, třídicího síťový provoz do toků.

- Generic Packet Classifier (*msgpc.sys*) – Když aplikace (ať už sama nebo v závislosti na filtrech v konfiguraci operačního systému) vyžaduje QoS, GPC zdrojové stanice zjistí typ služby (definovaný ve „Flowspec“), do které daný paket patří a předá tento paket do toku („Flow“) podle typu služby. Dále frontu zpracuje „Qos Packet Scheduler“ podle parametrů definovaných ve „Flowspec“.
- QoS Packet Scheduler (*psched.sys*) je komponenta jádra, která značí pakety a plánuje přenos paketů ve frontách. Také uplatňuje QoS parametry k docílení tvarování provozu. Provádí odesílání paketů do sítě na základě jejich priority, ale s omezenou rychlostí podle nastavených parametrů. Výrazně tedy ovlivňuje filozofii IP sítí „poslat vše co nejdříve“ tím, že fronty s různou prioritou paketů spolu na základě parametrů „soutěží“ o rozvrh odeslání. Aby byl zajištěn přednostní tok skrz síťové prvky, provádí se zmíněné značení rámců a paketů.

Rozhraní QoS

Windows XP a 2003 poskytují dvě úrovně aplikačních rozhraní (API) pro přiřazování QoS parametrů. První je Generic QoS API (GQoS), které je určeno pro vývojáře aplikací, tedy nastavování parametrů na úrovni programu. Druhé je Traffic Control API (TC API), určené pro správce sítí, kterým umožňuje skrze obslužný software nastavit chování síťového provozu pomocí filtrů vždy pro konkrétní stanici. Vrstvení systémových komponent a jejich rozhraní je naznačeno na obrázku 1.



Obrázek 1

GQoS API

MSDN definuje pro práci s GQoS API několik funkcí a struktur, zde uvádíme pouze jejich zběžný přehled. Kompletní dokumentace je k dispozici na:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/qos/qos/qos_reference.asp

Funkce:

```
INT WPUGetQOSTemplate( const LPGUID lpProviderId, LPWSABUF lpQOSName, LPQOS
lpQOS);
BOOL WSAAPI WSAGetQOSByName( SOCKET s, LPWSABUF lpQOSName, LPQOS lpQOS);
BOOL WSCInstallQOSTemplate( const LPGUID lpProviderId, LPWSABUF lpQOSName,
LPQOS lpQOS);
BOOL WSCRemoveQOSTemplate( const LPGUID lpProviderId, LPWSABUF lpQOSName);
BOOL WSAAPI WSPGetQOSByName( SOCKET s, LPWSABUF lpQOSName, LPQOS lpQOS,
LPINT lpErrno);
```

Struktury:

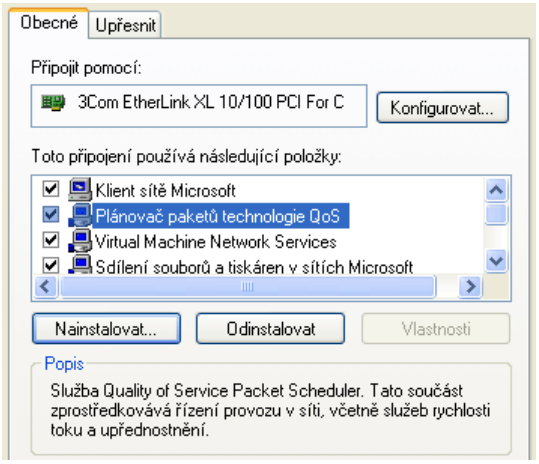
```
typedef struct _flowspec {
  ULONG TokenRate;
  ULONG TokenBucketSize;
  ULONG PeakBandwidth;
  ULONG Latency;
  ULONG DelayVariation;
  SERVICETYPE ServiceType;
  ULONG MaxSduSize;
  ULONG MinimumPolicedSize;
} FLOWSPEC,
 *PFLOWSPEC,
 *LPFLOWSPEC;

typedef struct _QualityOfService {
  FLOWSPEC SendingFlowspec;
  FLOWSPEC ReceivingFlowspec;
  WSABUF ProviderSpecific;
} QOS,
 *LPQOS;
```

Postup konfigurace systému pro QoS

(Názvy a obrázky jsou použity z české verze Windows XP SP2)

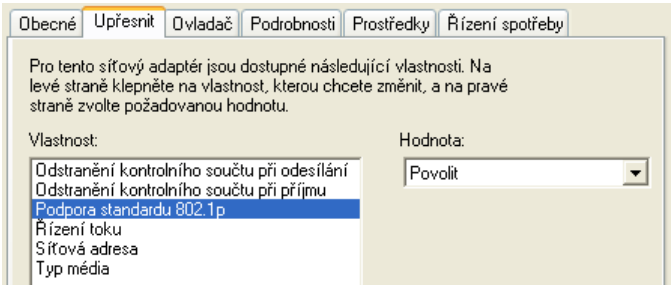
Pro správnou funkci QoS je potřeba mít na zvoleném síťovém adaptéru nainstalován „Plánovač paketů technologie QoS“ a mít jej povolen. (ve Windows XP je standardně, zatímco ve Windows 2003 je nutno jej nainstalovat). Postup instalace:

- Pokud máme zobrazeny vlastnosti konkrétního síťového připojení, klepneme na tlačítko „Nainstalovat“
- Poklepeme na kategorii „Služba“ a vybereme „Plánovač paketů technologie QoS“.
- Zkontrolujeme zda-li je povolen:



Obrázek 2

Pokud budeme vyžadovat také značení rámců (802.1p) a síťové zařízení toto podporuje, pak ve vlastnostech tohoto zařízení zkontrolujeme nastavení (názvy položek jsou závislé na HW a jeho ovladači):



Obrázek 3

Tímto jsou požadavky z hlediska nastavení systému kompletní.

Vynucení QoS parametrů pomocí TC API

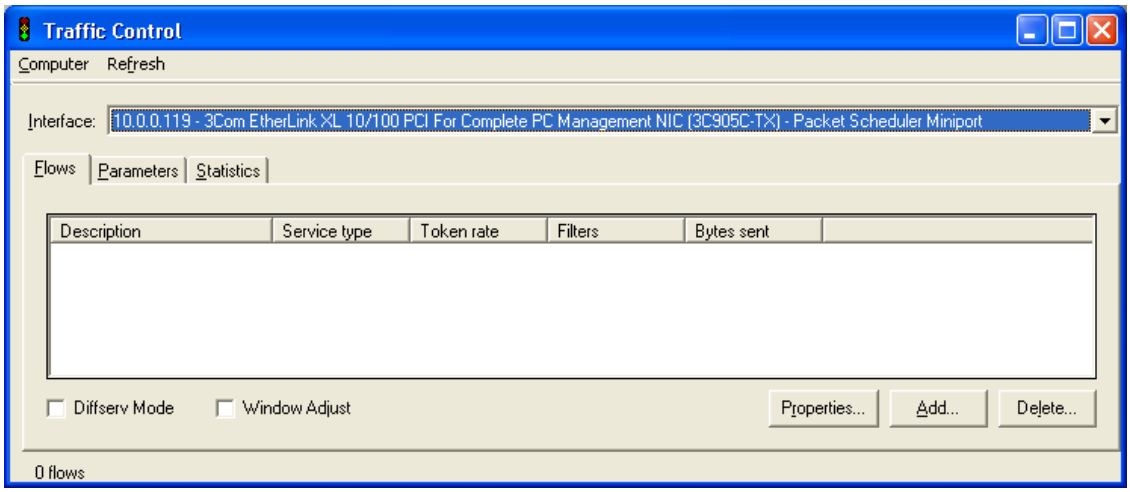
V této kapitole se budeme zabývat programem Traffic Control Monitor (*tcmon.exe*), který slouží k nastavení parametrů QoS pro aplikace nepodporující GQoS API. Nastavení se provádí vždy pro konkrétní stanici na základě zdrojových a cílových adres a portů, kterým se pomocí filtrů přidělují kategorie typů služeb.

Nástroj Traffic Control Monitor není součástí Windows, ale je volně k dispozici ve „Windows Ressource Kit Tools pro Windows Server 2003“. Zde je aktuální odkaz ke stažení: http://www.microsoft.com/downloads/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en. (Tento nástroj je možné provozovat na verzích 2000, XP i 2003.)

Po stažení souboru *rktools.exe* je třeba jej spustit (čímž se nainstalují nástroje resource kitu).

Aplikaci *tcmon.exe* nalezneme v instalační složce, ale vzhledem k tomu, že pro svůj provoz potřebuje zaregistrovat knihovny, nelze spustit, dokud tuto registraci neprovedeme – nejlépe pomocí souboru *tcmon.bat*.

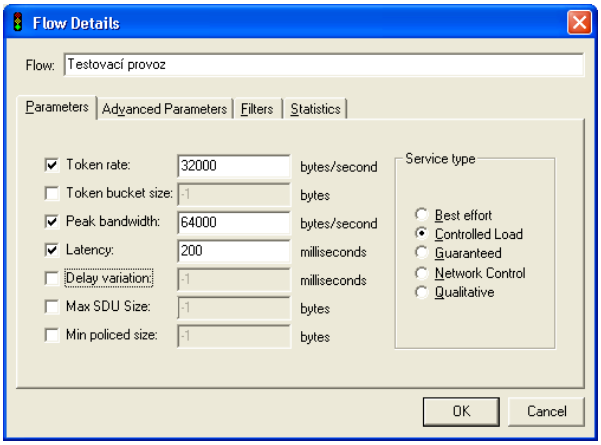
Nyní můžeme spustit aplikaci a provést potřebnou konfiguraci (toto vyžaduje administrátorská práva).



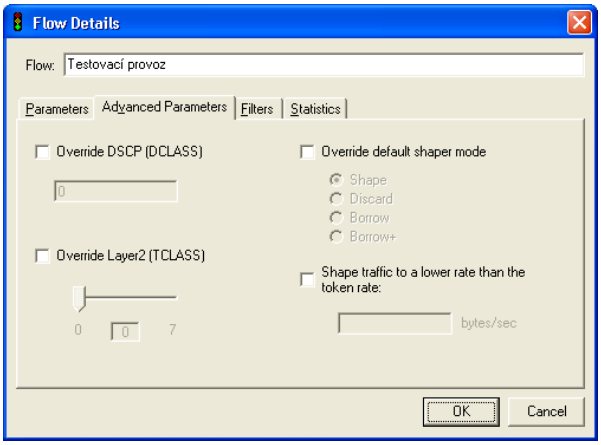
Obrázek 4

V hlavním okně aplikace (obrázek 4) musíme zvolit síťový adaptér, kterého se konfigurace bude týkat (položka „Interface“). Dále jsou v programu důležité 3 karty, přičemž konfigurace se týká zde zobrazená karta „Flows“, ve které definujeme toky. Jejich definice se nám zobrazí po klepnutí na tlačítko „Add…“ v okně se čtyřmi záložkami (obrázek 5). První záložka „Parameters“ definuje „Flowspec“ sestávající se zejména z šířky pásma, zpoždění, variace

zpoždění a dalších dle obrázku 5. U každého z těchto parametru také zaškrtávacím políčkem volíme, zda-li vůbec bude v daném „Flowspec“ zahrnut. Tímto nastavením ovšem není nijak ovlivněna nejdůležitější věc, kterou je typ služby („Service type“ v pravé části okna). Zde zvolíme jednu ze standardizovaných variant, jejichž vliv na komunikaci je zmíněn v úvodní části projektu.
Na obrázku 6 je zobrazena karta upřesňujících nastavení, které nám umožňuje například upravit hodnotu, kterou se označují rámce a/nebo pakety, či změnit algoritmus tvarování provozu.

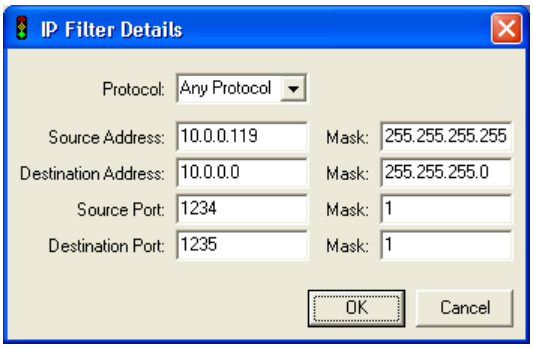


Obrázek 5

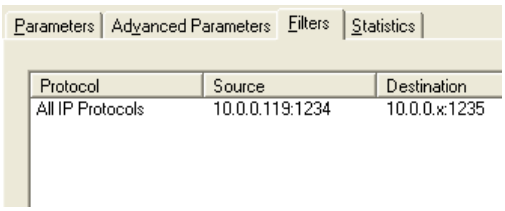


Obrázek 6

V tomto okamžiku konfigurace zbývá nadefinovat položky filtru („Filterspec“), pomocí kterých se v provozu třídí provoz. Položku filtru definujeme dialogem z obrázku 7, kde můžeme omezit protokol na TCP nebo UDP, či ponechat libovolný („Any protocol“), dále nadefinovat zdrojovou adresu (která musí být pro správný provoz shodná s alespoň jednou IP adresou konfigurovaného rozhraní na stanici), cílovou adresu, zdrojový port a cílový port. Masky definují jak u adres, tak u portů rozsah. Takovýchto položek filtrů můžeme nadefinovat libovolné množství, jejich seznam je pak zobrazen na kartě „Filters“.



Obrázek 7

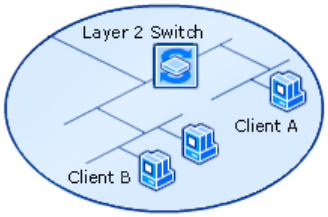


Obrázek 8

Nastavení se projeví ihned po potvrzení, není potřeba provádět restart žádných služeb. Ukládání probíhá do registrů. Pro monitorování jsme použili známý program *regmon.exe* z http://sysinternals.com. Při uložení proběhne přibližně 8000 přístupů ke klíčům registru, z velké většiny závislých na SID identifikátorech, což prakticky zabraňuje konfiguraci provádět (přenášet) jiným způsobem něž přes tyto API, nebo přímo používat generické QoS aplikace, které si QoS řídí samy.

Výsledky testování a závěr

Testování nastavení jsme prováděli jak na systémech Windows XP, tak Windows Server 2003 zapojených na lokální síti FastEthernet pomocí přepínače, viz obrázek 9.



Obrázek 9

Reálná funkčnost nastavených parametrů byla ověřena jednak teoreticky programem Ethereal, čímž jsme ověřili, zda skutečně dochází k podepisování paketů (náhled je na obrázku 10) a pak také prakticky v několik konfiguracích.

```
⊟ Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.0.3 (10.0.0.3)
    Version: 4
    Header length: 20 bytes
  ⊟ Differentiated Services Field: 0xa0 (DSCP 0x28: Class Selector 5; ECN: 0x00)
      1010 00.. = Differentiated Services Codepoint: Class Selector 5 (0x28)
      .... ..0. = ECN-Capable Transport (ECT): 0
      .... ...0 = ECN-CE: 0
    Total Length: 60
    Identification: 0x2a89 (10889)
```

Obrázek 10

Při praktickém testování jsme přistoupili ke zpomalení rychlosti sítě na 10Mbit half duplex a dále jsme využili program *fping*, čímž bylo snadno dosaženo potřebné zatížení nebo úplné zahlcení sítě provozem nad ICMP protokolem. Pro tento provoz jsme v žádném testu nevytvářeli pravidla filtrů a tedy spadal do kategorie „Best effort“.
První provedený test spočíval v navázání komunikace síťové hry, pro kterou byly vytvořeny filtry na obou stanicích (server a klient) a provoz byl zařazen do kategorie „Controlled load“. Při dostatečné hodnotě šířky pásma provoz probíhal bez ztrát nebo výrazného zpoždění. Ovlivňování hodnot zpoždění a variace zpoždění nemělo podle očekávání žádný vliv. Problém nastal v situaci kdy přidělená šířka pásma pro tento tok byla menší než požadovaná. Poté docházelo k řádově sekundovým výpadkům spojení nebo se spojení vůbec nedařilo navázat. Hranice mezi bezproblémovým provozem a nepoužitelným stavem je velice úzká, hlavně proto, že síťové hry využívají prakticky konstantní šířku pásma.
V dalším testu jsme zavedli ke stávajícím dvěma tokům (fping a síťová hra) třetí. Jednalo se o kopírování velkých souborů pomocí vestavěných služeb MS Windows, což jsme záměrně zvolili jako službu, která je schopna pojmout celou šířku pásma a přitom se snadno zjišťuje aktuální rychlost. Účelem testu bylo nastavit kategorii „Guaranteed service“ a zkoumat zda a jak přesný vliv má hodnota šířky pásma a jak se budou chovat další přenosy v nižší kategorii zejména z hlediska zpoždění. Výsledky rychlosti byly v tomto případě opět odpovídající zvoleným parametrům, aktuální rychlost přenosu dat byla vždy podle nastavené šířky pásma. Při využití větší části fyzické šířky pásma tímto tokem bylo negativně ovlivněno zpoždění síťové hry, což je vzhledem ke kategoriím typu služeb a jejich chování správné.

Těmito testy (s drobnými obměnami v nastavení) jsme ukončili praktickou část se závěrem, že nastavení základních hodnot skutečně ovlivňuje chování síťového provozu. Otázka kvality služby v reálném nasazení ale může být oproti laboratorním podmínkám vždy složitější, neboť její nasazení vyžaduje citlivé nastavení často velkého množství parametrů pro rozličné síťové přenosy a jejich iterační ladění.