2 Formát datagramu

Základním kamenem IPv6 je RFC 2460: *Internet Protocol, Version 6 (IPv6) Specification*, které obsahuje především formát datagramu pro IPv6. Ostatním mechanismům a datovým formátům, které souvisejí s IPv6, jsou věnovány další RFC dokumenty.

2.1 Datagram

Datagram má v IPv6 obvyklý základní tvar: začíná hlavičkami, za kterými pak následují nesená data. V porovnání s IPv4 však došlo v hlavičkách ke koncepční změně. Dříve byla jejich délka proměnlivá a jednotliví účastníci komunikace mohli připojovat další nepovinné části podle potřeby. Hlavička obsahovala kontrolní součet, který bylo třeba znovu vypočítat na každém směrovači, kterým datagram prošel (protože se změnila přinejmenším položka TTL).

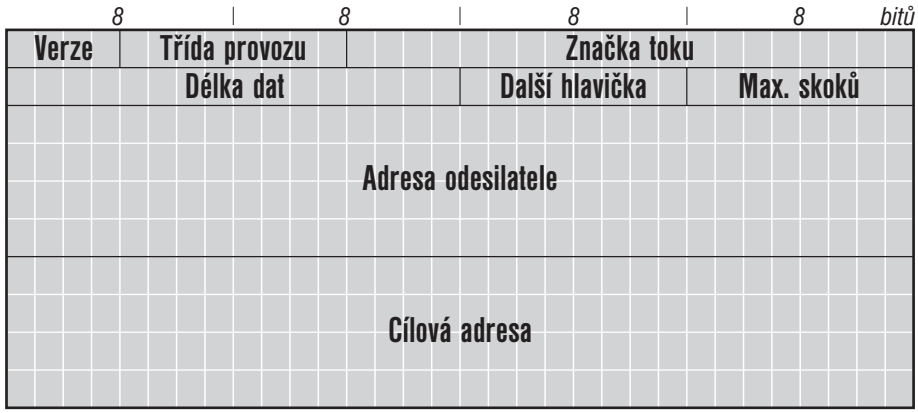
IPv6 naproti tomu standardní hlavičku minimalizovalo a omezilo její prvky jen na ty nejnutnější. Tato standardní hlavička má konstantní velikost. Veškeré doplňující, nepovinné či příležitostně užívané údaje byly přesunuty do rozšiřujících hlaviček, které v datagramu mohou a nemusí být přítomny. Jejich podobu a zpracování popíši v následující části.

Tvar základní hlavičky vidíte na obrázku 2.1. Přestože se adresy odesilatele a příjemce prodloužily čtyřikrát, celková délka základní hlavičky datagramu vzrostla ve srovnání s IPv4 jen dvojnásobně (z 20 B na 40 B, z toho 32 B zabírají adresy). Minimalismus je patrný na první pohled.

verze
Položka *Verze* je obvyklým zahájením IP datagramu, které identifikuje verzi protokolu. Zde obsahuje hodnotu 6.

třída provozu
Za ní následuje osmibitová *Třída provozu (traffic class)*, která vyjadřuje prioritu datagramu či jeho zařazení do určité přepravní třídy. Cílem je, aby tato položka umožnila IP poskytovat služby se zaručenou kvalitou. Její přesný význam však zatím nebyl definován. Očekává se, že vzejde z výzkumu a experimentů v oblasti diferencovaných služeb, zatím je požadováno, aby implicitní hodnotou byla 0.

značka toku
Dalších 20 bitů je věnováno *Značce toku (flow label)*. Koncepce toku je novinkou v IPv6 a stejně jako třída provozu zatím není přesně definována. V zásadě by jako tok měl být označován proud datagramů se společnými



Obrázek 2.1: Základní hlavička datagramu

vlastnostmi (odesilatel, adresát, požadavky na vlastnosti spojení). Prostřednictvím identifikátoru (značky) směrovač rychle rozpozná, že datagram je součástí určitého toku, což mu usnadní rozhodování o jeho dalším osudu (bude s ním naloženo stejně, jako s předchozími členy téhož toku). Jak již bylo řečeno, jedná se stále o experimentální půdu a nic moc konkrétního zatím nebylo stanoveno. K tématu se vrátím na konci kapitoly.

délka dat

Délka dat (payload length) nese údaj o délce datagramu. Přesně řečeno počet bajtů následujících za standardní hlavičkou. Z toho plyne, že základní hlavička se do této délky nepočítá, zatímco případné rozšiřující hlavičky ano. Jelikož je položka dvoubajtová, je maximální délkou 64 KB. Pokud je třeba vytvořit delší datagram, lze použít rozšiřující hlavičku *Jumbo Payload*.

další hlavička

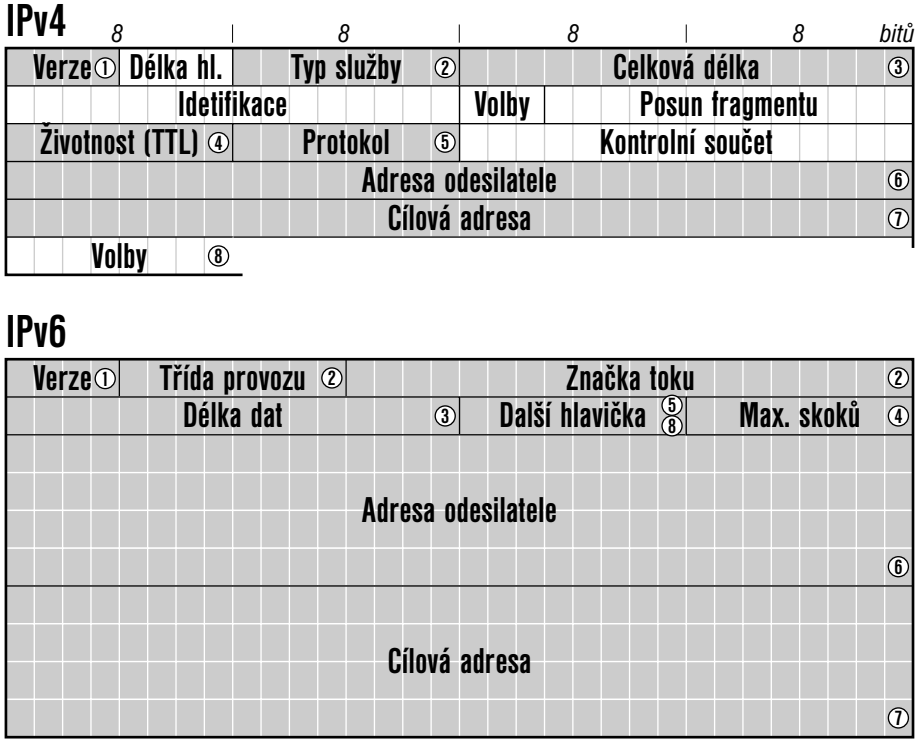
Další hlavička (next header) obsahuje identifikaci, jaká hlavička či jaký druh dat následuje za standardní hlavičkou. Podrobněji se jí budu věnovat zanedlouho v části 2.2.

dosah

Maximální počet skoků (hop limit) je náhradníkem dřívější životnosti datagramu (TTL). Průchod datagramu jedním směrovačem je považován za jeden skok. Odesilatel v této položce uvede, kolik takových skoků smí datagram maximálně absolvovat. Každý směrovač po cestě pak sníží hodnotu o jedničku. Dojde-li tím k vynulování položky, datagram bude zlikvidován a odesilateli se pošle ICMP zpráva o vypršení maximálního počtu skoků. Smyslem položky je ochrana proti cyklům při směrování (zacyklený datagram nebude v síti strašit do nekonečna).

adresy

Závěrečnými dvěma položkami je dvojice IPv6 adres: *Adresa odesilatele (source address)* a *Cílová adresa (destination address)*. Vzhledem k délce adresy v IPv6 zabírají tyto dvě položky 80 % rozsahu celé hlavičky. Podrobnosti o adresování se dočtete v kapitole 3 na straně 35.



Obrázek 2.2: Porovnání hlaviček IPv4 a IPv6

Při srovnání s IPv4 je nejnápadnější absence tří informací: rozšiřujících voleb, kontrolního součtu a fragmentace. Rozšiřující volby byly nahrazeny obecnějším mechanismem zřetězení rozšiřujících hlaviček. Obdobně údaje související s fragmentací byly přesunuty do těchto rozšiřujících hlaviček. Zdaleka ne každý paket je totiž fragmentován a lze očekávat, že v IPv6 bude fragmentace ještě vzácnější než v současnosti. IPv6 totiž požaduje, aby infrastruktura pro jeho přenos dovedla přenášet pakety minimálně o délce 1280 B (MTU). Vzhledem k tomu, že drtivá většina koncových zařízení je dnes připojena prostřednictvím různých variant Ethernetu s MTU 1500 B, lze očekávat, že tato maximální velikost paketů se usídlí ve valné většině infrastruktury a fragmentace prakticky zmizí ze světa.

Kontrolní součet zmizel bez náhrady. Tuto službu totiž typicky vykonává nižší vrstva síťové architektury (např. zmiňovaný Ethernet), takže na úrovni IP by se jen opakovalo její snažení. Vzhledem k tomu, že hlavička se mění v každém směrovači, znamenalo by to zbytečné zpomalování.

Porovnání hlaviček IPv4 a IPv6 názorně představuje obrázek 2.2. V IPv4 datagramu jsou šedě vyznačeny položky, které byly (zpravidla v poněkud

pozměněné podobě) převzaty do IPv6. Stejná čísla označují položky, které si navzájem odpovídají.

2.2 Zřetězení hlaviček

IP verze 6 používá odlišný způsob reprezentace rozšiřujících hlaviček než jeho předchůdce. Každá hlavička je nyní samostatným blokem a k jejich vzájemnému propojení slouží položka *Další hlavička (Next header)*. Kód v ní obsažený identifikuje, jakého typu je hlavička, která následuje za tou stávající. Každá rozšiřující hlavička začíná položkou *Další hlavička*. Prostřednictvím těchto hodnot lze za sebe zřetězit hlaviček, co hrdlo ráčí.

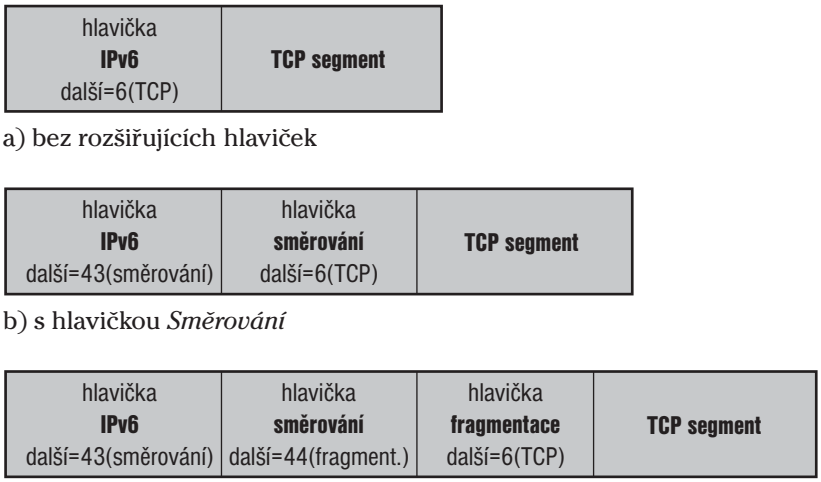
Poslední z nich obsahuje v položce *další hlavička* typ dat, která datagram nese. Hodnota položky *další hlavička* tak zároveň zastupuje dřívější položku *Protokol*. Nejvýznamnější hodnoty shrnuje tabulka 2.1. Aktuální a kompletní specifikaci hodnot pro typy přenášených dat najdete na adrese *http://www.iana.org/assignments/protocol-numbers*.

Rozšiřující hlavičky	
0	volby pro všechny (hop-by-hop options)
43	směrování (routing)
44	fragmentace (fragment)
50	šifrování obsahu (ESP)
51	autentizace (AH)
59	poslední hlavička (no next header)
60	volby pro cíl (destination options)
62	mobilita (návrh – viz strana 151)
Typ nesených dat	
6	TCP
8	EGP
9	IGP
17	UDP
46	RSVP
47	GRE
58	ICMP

Tabulka 2.1: Hodnoty položky *Další hlavička*

Pokud tedy datagram neobsahuje žádné rozšířené hlavičky, bude přímo jeho základní IPv6 hlavička obsahovat jako *Další hlavičku* identifikátor typu nesených dat. Tuto situaci ilustruje obrázek 2.3a. Na obrázcích 2.3b a 2.3c můžete sledovat, jak se změní obsah položek *Další hlavička*, když datagramu přidáme rozšiřující hlavičky *Směrování* a *Fragmentace*.

Hlavními devizami koncepce hlaviček v IPv6 je pružnost a úspornost. Součástí datagramu jsou jen ty průvodní informace, které skutečně potřebuje.



Obrázek 2.3: Zřetězení hlaviček datagramu

Rubem mince je, že zpracování kompletních hlaviček může představovat průchod relativně dlouhým řetězcem. Pokud by se mělo odehrávat v každém směrovači na cestě mezi odesilatelem a příjemcem, mohlo by to vést k nezanedbatelné degradaci výkonu.

pořadí hlaviček

Tento problém řeší IPv6 velmi jednoduše – rozšiřující hlavičky mají předepsáno následující pořadí:

1. základní hlavička IPv6
2. volby pro všechny (hop-by-hop opions)
3. volby pro cíl (destination options) – pro první cílovou adresu datagramu a případné další uvedené v hlavičce *Směrování*
4. směrování (routing)
5. fragmentace (fragment)
6. autentizace (authentication)
7. šifrování obsahu (encapsulating security payload)
8. volby pro cíl (destination options) – pro konečného příjemce datagramu

Jeho cílem je, aby se informace zajímavé pro uzly, kterými datagram prochází, ocitly vpředu a hlavičky určené až pro koncového příjemce následovaly teprve za nimi. Pro průchozí směrovač jsou potenciálně zajímavé jen *Volby pro všechny*, které se smí vyskytnout jen bezprostředně za základní

hlavičkou. Ničeho jiného si nemusí všímat. Jakmile vidí v *Další hlavičce* jiný kód než 0 (*Volby pro každého*), ví, že může s analýzou datagramu skončit.

Ostatní rozšiřující hlavičky jsou zajímavé jen pro adresáta datagramu – ať už průběžného (pocházejícího z hlavičky *Směrování*) či koncového. Průběžného adresáta zajímají jen první tři (volby pro všechny, volby pro cíl a směrování), zatímco koncového se týkají všechny. Podle RFC 2460 adresát musí být schopen se vyrovnat s libovolným pořadím hlaviček, nicméně důrazně se doporučuje dodržovat výše uvedené.

Každá z rozšiřujících hlaviček by se měla objevit nanejvýš jednou. Výjimkou jsou volby pro cíl, které se mohou vyskytnout dvakrát – jednou před *Směrováním* a podruhé před nesenými daty.

Speciální význam má, pokud položka *Další hlavička* obsahuje hodnotu 59 (no next header). Ta signalizuje, že se jedná o poslední hlavičku, za kterou již nenásleduje vůbec nic. Pokud datagram podle své délky obsahuje ještě nějaká data, musí být ignorována. Je-li datagram přeposílán dále, musí do něj předávající tato data zkopírovat beze změny.

Ve zbytku kapitoly podrobněji popíši tvar a význam jednotlivých rozšiřujících hlaviček.

2.3 Volby

Stávající IPv6 zavádí dvě hlavičky obsahující volby: *Volby pro všechny* (*hop-by-hop options*, *Další hlavička* před nimi má hodnotu 0) a *Volby pro cíl* (*destination options*, předcházející *Další hlavička* má hodnotu 60).

Obě hlavičky mají společný tvar, který najdete na obrázku 2.4. Význam položky *Další hlavička* jsem již vysvětlil. *Délka dat* obsahuje délku hlavičky v osmicích bajtů. Do délky se nepočítá prvních 8 bajtů, takže pokud má hodnotu 1, znamená to, že celá hlavička s volbami měří 16 B.

Obrázek 2.4: Rozšiřující hlavičky *volby pro všechny* a *volby pro cíl*

Položka *Volby* pak obsahuje vlastní volby. Ty mohou být zavedeny jako součást jednotlivých konkrétních mechanismů. Například v rámci podpory mobilních počítačů se objevila volba (*Domácí adresa*). Samotná definice IPv6 obsahuje jen dvě: *Pad1* a *PadN*. Slouží ke vkládání „vaty“ – volného místa, které má sloužit k lepšímu zarovnání ostatních prvků s přihlédnutím k hranicím čtyřbajtových slov. Jedná se o vycpávky, které nenesou

žádnou aktivní informaci. Přehled doposud definovaných voleb najdete v tabulkách 2.2 a 2.3.

Typ	*Význam*	*Strana*
0	Pad1	25
1	PadN	25
5	Upozornění směrovače	26
194	Jumbo obsah	31

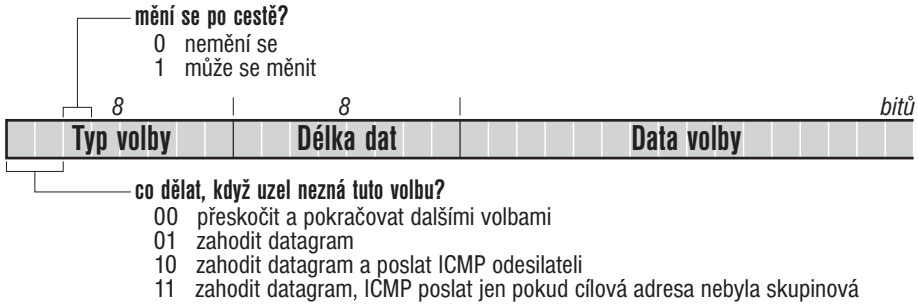
Tabulka 2.2: Volby pro všechny

Typ	*Význam*	*Strana*
0	Pad1	25
1	PadN	25
201	Domácí adresa	163

Tabulka 2.3: Volby pro příjemce

Pad1 *Pad1* vynechává 1 bajt. Tvar této volby je triviální: jedná se o jeden bajt s hodnotou 0, která identifikuje typ volby a zároveň říká, že to je vše.

PadN *PadN* umožňuje vynechat dva a více bajtů. První bajt opět určuje typ volby a má hodnotu 1. Za ním následuje jeden bajt obsahující délku volby, do níž se první dva bajty nepočítají. Následují data uvedené délky, jejichž hodnoty jsou nulové. Chcete-li tedy vynechat celkem 6 bajtů, bude mít *Délka dat* hodnotu 4 a za ní budou následovat čtyři nulové bajty „dat“.



Obrázek 2.5: Tvar voleb pro rozšiřující hlavičky

formát voleb Všechny volby musí dodržovat jednotný tvar. Odpovídá tomu, který jste viděli u volby *PadN*. První bajt identifikuje, o jakou volbu se jedná. Za ním pak následuje *Délka dat* (do níž se nepočítají první dva bajty) a po ní data. Jejich strukturu musí definovat dokument, který zavede danou volbu.

V rámci *Typu volby* byl pevně předepsán význam nejvyšších tří bitů. První dva určují, co se stane s datagramem, pokud zpracovávající uzel dotyčnou volbu nezná. Za nimi následuje bit, který indikuje, zda se volba může měnit během průchodu sítí. Konkrétní hodnoty najdete v obrázku 2.5.

upozornění směrovače

Jednou z „opravdových“ voleb je tak zvané *Upozornění směrovače (router alert)* definované v RFC 2711. Jedná se o volbu pro všechny, která má za cíl upozornit každý směrovač po cestě, že tento paket nese data, která by jej mohla zajímat.

Volba najde uplatnění například v rezervačním protokolu RSVP, který posílá řídicí pakety pro alokaci kapacit po cestě. Tyto pakety jsou určeny všem směrovačům. Právě *Upozornění směrovače* může napovědět, že paket nese zajímavou informaci. Bez něj by směrovač musel prohlížet všechny datagramy a zkoumat, jakému protokolu vyšší vrstvy patří. Když by narazil na RSVP paket, zabýval by se jím podrobněji. V opačném případě by jej poslal dále po cestě k cíli.

Díky *Upozornění směrovače* lze rychle odlišit datagramy potenciálně zajímavé od těch, které se mají prostě předávat dál. Formát volby najdete na obrázku 2.6. Obsahuje vlastně jedinou položku, která slouží k identifikaci protokolu, jehož data nese. Dosud definované hodnoty shrnuje tabulka 2.4.

Obrázek 2.6: Volba *Upozornění směrovače*

Hodnota	*Význam*
0	obsahuje MLD zprávu
1	obsahuje RSVP zprávu
2	obsahuje zprávu *Aktivní sítě*
3–35	úroveň vnoření agregovaných rezervací (RFC 3175)

Tabulka 2.4: Definované *Hodnoty* pro volbu *Upozornění směrovače*

Aby tato volba přinášela nějaký efekt, musí odpovídající protokol nařizovat její použití. Směrovač má právo ignorovat obsah všech datagramů, které nejsou adresovány jemu a neobsahují *Upozornění směrovače*. Chce-li určitý protokol získat jeho pozornost, musí k datagramu přihodit tuto volbu.

2.4 Směrování

Hlavička *Směrování* umožňuje předepsat datagramu určité body, kterými musí v daném pořadí projít. Zároveň slouží jako záznam, kterými z nich již

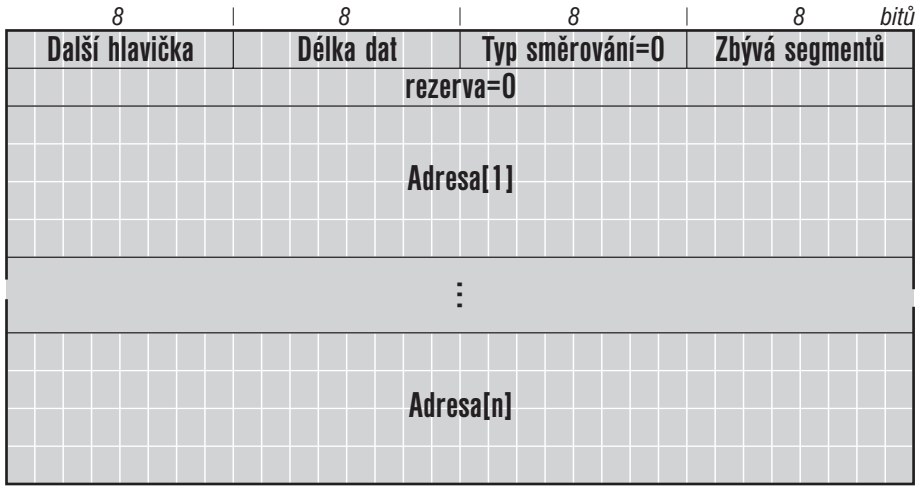
prošel. Tyto „průchozí body“ nemusí následovat bezprostředně za sebou. Mezi každými dvěma může datagram projít libovolným počtem směrovačů.

IPv6 ponechává prostor pro zavedení různých typů směrovacích hlaviček. Zatím byl definován jen jediný – typ 0, který zde popíši. V rámci mobility je navržen typ 2 (viz obrázek 11.14 na straně 163), který však zatím nebyl standardizován, jedná se jen o pracovní návrh. Nepřináší nic nového, naopak představuje zjednodušení typu 0.

Formát hlavičky *Směrování* typu 0 představuje obrázek 2.7. Pokud chce odesilatel, aby jeho datagram po cestě k cíli prošel určitými uzly, uvede jako jeho cílovou adresu IP adresu prvního z těchto průchozích uzlů. Do hlavičky *Směrování* pak zapíše postupně adresy zbývajících a na závěr konečný cíl datagramu. V položce *Zbývá segmentů* uvede jejich počet.

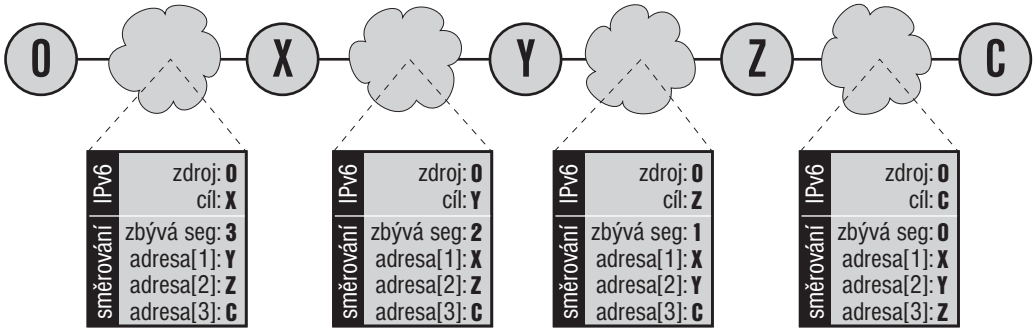
Když datagram dorazí na cílovou adresu uvedenou v základní IPv6 hlavičce a obsahuje hlavičku *Směrování* s nenulovým počtem zbývajících segmentů, vezme počet zbývajících segmentů jako index do tabulky průchozích adres. Určuje, kolikátá od konce je adresa následujícího průchozího bodu. Tuto adresu zapíše jako cílovou do základní hlavičky a dosavadní cílovou (tedy svoji) uloží místo ní do směrovací hlavičky. Následně zmenší počet zbývajících segmentů o jedničku a pošle datagram novému cíli (dalšímu na vytyčené cestě).

Položka *Zbývá segmentů* tedy odděluje, které z uvedených adres již byly navštíveny a kterými datagram ještě musí projít. Je-li nulová, znamená to, že datagram dorazil do svého cíle. Formát směrovací hlavičky typu 0 představuje obrázek 2.7.



Obrázek 2.7: Rozšiřující hlavička *Směrování* typu 0

Obrázek 2.8 ilustruje vývoj hodnot zajímavých položek při průchodu datagramu sítí. Jeho odesilatelem je *O* a koncovým adresátem *C*. Odesilatel předepsal, že datagram má projít uzly *X*, *Y* a *Z*.



Obrázek 2.8: Změny v hlavičkách datagramu

2.5 Fragmentace

Každá z podřízených technologií, které IPv6 používá pro přepravu svých datagramů, má jistou maximální velikost paketů, které dokáže přenášet. Tato konstanta se označuje zkratkou MTU (Maximum Transmission Unit). Například nejpopulárnější Ethernet má MTU = 1500 B.

Cílem fragmentace je umožnit IPv6 přepravovat datagramy větší, než je MTU používaných technologií. Základní myšlenka je prostá: odesilatel rozloží datagram do několika dostatečně malých částí a příjemce z nich poskládá původní datagram.

odlišnosti IPv4

Analogickou techniku používal i protokol IPv4, lišil se však v několika důležitých detailech. Zatímco v IPv4 mohl datagram fragmentovat libovolný směrovač po cestě (kdykoli měl být odeslán linkou, jejíž MTU bylo větší než velikost datagramu), *v IPv6 fragmentuje výlučně odesilatel*. Pokud má některý ze směrovačů odeslat datagram linkou s nedostačujícím MTU, zahodí jej a pošle odesilateli ICMP zprávu „příliš velký paket“, jejíž součástí je i MTU, které tento stav způsobilo. Druhou odlišností je, že zatímco IPv4 má všechny podklady pro fragmentaci zařazeny již do standardní hlavičky, IPv6 pro ni používá hlavičku rozšiřující a spíše se snaží, aby k fragmentaci vůbec nedocházelo.

formát hlavičky

Rozšiřující hlavička pro fragmentaci je identifikována kódem 44 v položce *Další hlavička* svého bezprostředního předchůdce. Její tvar vidíte na obrázku 2.9. Velikost je konstantní a kromě obvyklé *Další hlavičky* obsahuje tři informační položky.

Obrázek 2.9: Rozšiřující hlavička *Fragmentace*

Identifikace slouží k rozpoznání, které fragmenty patří k sobě. Jedná se o 32bitové celé číslo, které je v rámci dané dvojice odesilatel–příjemce pokud možno jednoznačné (každý další fragmentovaný datagram má číslo o jedničku vyšší než předchozí, po naplnění kapacity čítače se začne znovu od nuly). *Posun fragmentu* říká, kam tento fragment patří. Jednotkou jsou osmice bajtů od začátku fragmentovatelné části původního datagramu (viz níže). A konečně příznak *M* (z anglického „more fragments“) signalizuje, zda je tento fragment poslední (hodnota 0) nebo za ním následuje další (hodnota 1).

fragmentovatelná a nefragmentovatelná část

Má-li dojít k fragmentaci, vymezí se v původním datagramu dvě části: na začátku je tak zvaná *nefragmentovatelná část*, kterou tvoří standardní IPv6 hlavička a všechny po ní následující rozšiřující hlavičky až po *Směrování* (včetně). Tedy vše, co v pořadí rozšiřujících hlaviček předchází před fragmentací. Zbytek datagramu je považován za *fragmentovatelnou část* a pouze on je předmětem fragmentace.

postup fragmentace

Tato fragmentovatelná část se rozdělí na části, jejichž velikost je násobkem 8 B a je dostatečně malá na to, aby celková velikost výsledných datagramů nepřekročila požadované MTU. Tím z původního datagramu vznikne několik fragmentů – nových datagramů. Jejich hlavičky jsou sestaveny takto:

- Převezme se nefragmentovatelná část původního datagramu. Jedinými změnami, které se v ní pro jednotlivé fragmenty provedou, je úprava velikosti v základní hlavičce, aby odpovídala skutečné velikosti fragmentu, a změna hodnoty poslední *Další hlavičky* na 44.
- Za ni se přidá rozšiřující hlavička *Fragmentace*, jejíž hodnoty se naplní následovně:
  - vygeneruje se nový *Identifikátor* paketu a tato hodnota se přidělí všem jeho fragmentům
  - hodnota *Další hlavičky* se převezme z poslední *Další hlavičky* nefragmentovatelné části původního datagramu
  - *Posun* každého fragmentu se určí jako počet osmic bajtů, o které je jeho začátek vzdálen od začátku fragmentovatelné části původního datagramu; jelikož všechny fragmenty (kromě posledního) budou mít stejnou délku x, bude mít první fragment *Posun* nulový, druhý fragment ponese *Posun*=x, třetí *Posun*=2x atd.

  - posledním fragmentu se příznak *M* nastaví na 0, ostatním na 1
- Na konec se připojí dotyčný fragment (úsek fragmentovatelné části původního datagramu).

Vzniklé fragmenty jsou jako samostatné datagramy odeslány adresátovi. Ten je posbírá a z údajů ve fragmentační hlavičce dokáže složit původní datagram: podle *Identifikátoru* pozná, které fragmenty patří k sobě, pomocí *Posunutí* určí správné pořadí a zjistí případné chybějící části a konečně příznak *M* mu prozradí, zda má k dispozici všechny kousky.

Na základě těchto údajů příjemce poskládá původní datagram do podoby, kterou měl před fragmentací (tím zaniknou hlavičky *Fragmentace* jednotlivých částí) a ten pak dále zpracovává bez ohledu na to, že mu přišel po kouskách.

2.6 Velikost datagramů

Fragmentace těsně souvisí s velikostí odesílaných datagramů. Každý datagram navíc přináší určitou (byť malou) zátěž – musí mít své hlavičky, směrovače po cestě se musí rozhodovat, kudy jej poslat, a podobně. Ideálem je, aby datagramy byly pokud možno co největší, aby jich bylo co nejméně a snižovala se tak nadbytečná zátěž. Na druhé straně však datagramy musí být natolik malé, aby nikde po své cestě nepřekročily MTU a nedocházelo tudíž k fragmentaci.

objevování MTU cesty

O dosažení tohoto kompromisu se snaží algoritmus nazvaný objevování MTU cesty. Definuje jej RFC 1981: *Path MTU Discovery for IP version 6*.

Z pohledu teoretika nemá vůbec smysl mluvit o nějakých cestách v souvislosti s protokolem IP. Nabízí službu bez spojení, kdy je každý datagram směrován samostatně a nezávisle na ostatních. To znamená, že každý ze skupiny datagramů tvořících jeden soubor může dorazit k cíli jinou cestou. V praxi se však směrovací tabulky nemění příliš rychle a je vysoce pravděpodobné, že datagramy odeslané v krátkém časovém intervalu ke stejnému cíli budou putovat stejnou trasou. Na tomto pozorování ostatně stojí již letitý program *traceroute*.

Objevování MTU cesty má za cíl najít maximální velikost paketu, který lze poslat danému cíli. Postupuje jednoduše: nejprve pošle datagram, jehož velikost je rovna MTU rozhraní, kterým datagram odesílá. Celkové MTU jistě nemůže být větší. Pokud datagram úspěšně dojde, máme nalezeno MTU cesty.

Jestliže někde narazí na úsek s menším MTU, směrovač na jeho začátku datagram zahodí a pošle odesilateli ICMP zprávu „příliš velký datagram“. Její součástí je i hodnota MTU dotyčné trasy. Odesilatel si příslušně zmenší

svůj odhad MTU cesty a zkusí štěstí znovu s datagramem této velikosti. Celý proces se opakuje tak dlouho, dokud se datagramy nedostanou až k cíli.

Pokud komunikace trvá delší dobu, s vysokou pravděpodobností dojde ke změně cesty, případně i několikanásobné. Hledání MTU se snaží s touto skutečností vyrovnat. Pokud MTU cesty poklesne, odesilatel na to přijde hned – obdrží ICMP zprávu o příliš velkém datagramu. O případném zvětšení se však touto cestou nedozví. Proto by měl čas od času zopakovat celý algoritmus hledání MTU, aby zjistil, zda aktuální hodnota není vyšší, než se domnívá. V RFC se požaduje, aby interval mezi těmito zkouškami byl minimálně 5 minut, doporučená hodnota je 10 minut.

Ostatně vzhledem k tomu, že MTU na linkách podporujících IPv6 má být alespoň 1280 B a doporučuje se používat 1500 B nebo více, lze očekávat, že MTU cesty bude zpravidla 1500 B a prakticky se nebude měnit. Klient nesmí zmenšit MTU cesty pod 1280 B. Pokud mu někdo ohlásí nižší hodnotu, musí datagramy fragmentovat.

Objevování MTU cesty lze používat i pro skupinové adresy. V tomto případě může dostat na jeden datagram celou řadu ICMP zpráv. Bude se chovat podle očekávání – použije nejmenší ohlášenou hodnotu.

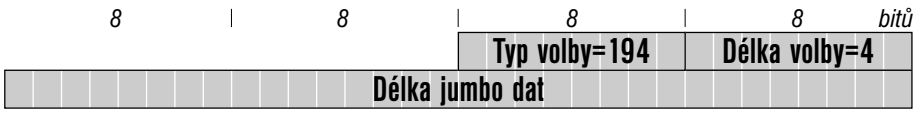
Implementace popsaného algoritmu je autory IPv6 důrazně doporučena, není však povinná. Jedná-li se o minimalistickou implementaci IPv6 (např. v ROM přenosného zařízení), může používat hodnotu 1280 B, aniž by se pokoušela zjistit, zda skutečné MTU cesty není vyšší.

2.7 Jumbogramy

Jelikož je délka nesených dat v IPv6 datagramu ukládána do 16bitové položky, je maximální dosažitelnou hodnotou 65 535 bajtů. Troufám si tvrdit, že případy, kdy by tento horní limit byl pociťován jako omezení, budou opravdu velmi velmi vzácné. Nicméně i pro ně nabízí IPv6 řešení. Jedná se o volbu *Jumbo obsah*, která umožňuje vytvářet datagramy o délce 65 536 až 4 294 967 295 B. Patří mezi *Volby pro všechny*, takže se jí bude zabývat každý směrovač po trase.

Použití je prosté: *Délka dat* v základní hlavičce se vynuluje a přidá se rozšiřující hlavička s volbami pro všechny obsahující *Jumbo obsah*. Nese položku *Délka jumbo dat*, která měří 32 bitů a umožňuje proto výše uvedený rozsah přípustných hodnot. Takto velké datagramy jsou označovány jako jumbogramy.

Použití jumbogramů má pochopitelně smysl jen v případě, kdy linková technologie umožňuje přenos takto velkých paketů. Jinými slovy pokud MTU dotyčné linky přesahuje 65 575 (maximální velikost nesených dat



Obrázek 2.10: Volba *Jumbo obsah*

plus IPv6 hlavička). Uzly, které nemají tak velké MTU, nemusí jumbogramy podporovat a ani této volbě rozumět.

UDP Příliš velké datagramy ale vadí i protokolům vyšší vrstvy. Například UDP má svou vlastní položku pro délku dat, která je také 16bitová. RFC 2675, které definuje jumbogramy, proto doporučuje, aby na strojích s jejich podporou byl pozměněn kód i ve vyšších vrstvách. Konkrétně pro UDP doporučuje, aby se u jumbogramů uváděla na úrovni UDP nulová délka a aby si kód pro UDP nechal sdělit skutečnou délku od IP vrstvy.

TCP TCP sice nemá ve svých hlavičkách délku, ale definuje volbu *Maximální délka segmentu (Maximum Segment Size, MSS)*, která – jak jinak – používá 16bitovou hodnotu. Doporučenou strategií je prohlásit 65 535 za nekonečno. Pokud jeden z partnerů dostane MSS s touto hodnotou, určí si skutečnou maximální délku segmentu z nalezeného MTU cesty (odečtením 60 B na IPv6 a TCP hlavičky).

Druhým délkovým údajem v TCP je délka urgentních dat. Autoři RFC 2675 považují za nepravděpodobné, že by se urgentní data používala v kombinaci s jumbogramy. Kdyby k tomu však přece jen došlo, doporučují i zde prohlásit 65 535 za nekonečno. Tato hodnota v položce *Urgent pointer* TCP hlavičky znamená „všechna data v tomto datagramu jsou urgentní“. Při odesílání TCP paketu s dlouhou urgentní částí je třeba jej rozdělit na dva tak, aby ve druhém byla délka urgentní části menší než 65 535.

Upřímně řečeno považuji jumbogramy spíše za zajímavou teoretickou konstrukci než za prakticky použitelný nástroj. MTU tak velká, aby ji umožňovala použít, se v současném Internetu nevyskytují.

2.8 Toky

Jedním z nových prvků IPv6 je koncepce toku. Idea je jasná: tok je proud datagramů, které spolu „nějak souvisí“. Často tok odpovídá transportnímu spojení (například TCP spojení mezi WWW klientem a serverem může být dobrým kandidátem pro tok), ale nemusí tomu tak nutně být.

Ohledně toků je zatím definitivní jen délka příslušné položky v IPv6 hlavičce. Vše ostatní se dosud vyvíjí a zatím jsou popsány jen základní principy. Obsahuje je návrh *draft-ietf-ipv6-flow-label-01*, ze kterého budu vycházet.

Hlavním cílem toku je usnadnit a urychlit zpracování datagramů v prvcích, kterými procházejí. V nich je uložena informace, která slouží jednak k rozpoznání toku, jednak obsahuje instrukce pro zacházení s datagramy tohoto toku. V návrhu se pro tuto informaci používá název *stav toku*.

Konkrétní postupy, jak vznikne a je spravován stav toku, byste v návrhu hledali marně. Naznačuje však možné přístupy k řešení tohoto problému:

- dynamický – stav toku si spravuje odesilatel (např. pomocí RSVP)
- kvazidynamický – řídí správa sítě
- statický – je konfigurován ručně
- algoritmický – vychází z nezávislých algoritmů (např. pro rozkládání zátěže)

Naproti tomu je jasně stanovena klasifikace. Datagram je přiřazen k určitému toku na základě tří údajů: IP adresy odesilatele, cílové IP adresy a značky toku. Tato trojice tvoří jednoznačný identifikátor, podle kterého příjemce i každý po cestě k němu rozpozná tok.

Přidělení značky toku má na starosti výlučně odesilatel datagramu. Po cestě se značka nesmí měnit. Odesilatel by měl související datagramy směřující ke stejnému cíli opatřovat stejnou značkou toku. Je také povinen vést si přehled o značkách, které momentálně používá, aby jeho chování bylo konzistentní.

Hodnota značky nemá definovánu žádnou vnitřní strukturu a je v podstatě libovolná. Doporučuje se, aby je odesilatel přiděloval náhodně. Jedinou zakázanou hodnotou je nula, která znamená, že dotyčný datagram nepatří k žádnému toku.

Podpora toků není povinná. Průchozí zařízení může brát na tok zřetel, nebo nemusí. V tom případě však musí informace související s tokem ignorovat a nijak do nich nezasahovat. Tím je zajištěno, že nic nepokazí těm strojům, které jsou za ním a věci rozumějí.