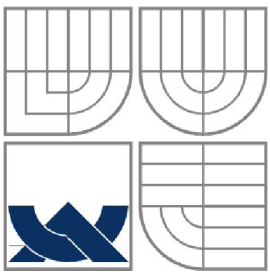
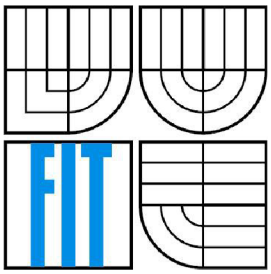


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MĚŘENÍ ENTROPIE V INTERNETOVÉ KOMUNIKACI

ENTROPY MEASUREMENT IN INTERNET COMMUNICATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID STEJSKAL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. VIKTOR PUŠ

BRNO 2012

Abstrakt

Tato práce se zabývá měřením entropie a jejím použitím při zhodnocení současně používaných internetových protokolů. Uvádí definici a výpočet entropie a její další související vlastnosti. Srovnává souvislost entropie dat protokolu s datovou režií daného protokolu. Dále se zabývá implementací nástroje pro měření entropie a možnostmi rozšíření funkcionality. Poté následuje uvedení výsledků měření a jejich zhodnocení. Sleduje vývoj využití protokolů a jejich polí.

Abstract

This work focuses on entropy measurement in Internet communication and elaboration on nowaday's protocols used in said communication. It elaborates on calculation of entropy and its properties. Our goal is to implement a utility for measuring the entropy, which is discussed in its own chapter. Later, results of the measurement are included and discussed in detail. Recent trends in protocol usage are also charted and discussed.

Klíčová slova

Entropie, měření, protokol, Internetová komunikace, binární vyhledávací strom

Keywords

Entropy, measurement, protocol, Internet communication, binary search tree

Citace

David Stejskal: Měření entropie v Internetové komunikaci, bakalářská práce, Brno, FIT VUT v Brně, 2012

Měření entropie v Internetové komunikaci

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Viktora Puše. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
David Stejskal
Datum (16.5.2012)

Poděkování

Děkuji panu Ing. Viktorovi Puši za ochotu vždy poradit, protože bez jeho pomoci bych si občas nevěděl rady.

© David Stejskal, 2012
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod

Sítě a zejména Internet jsou dnes prakticky nepostradatelné v běžném životě. Počet uživatelů však neustále roste a s ním i nároky na infrastrukturu, kterou je Internet realizován. Zvětšuje se objem dat přenášených za jednotku času. Proto se přenosové linky vylepšují, aby splnily požadavky, které jsou na ně kladeny. Základní požadavek je rychlost. Obvyklým postupem pro zvýšení propustnosti linky je zrychlení celé datové cesty. Druhým a často zanedbávaným krokem je zefektivnění využívaných protokolů tak, aby se přenášelo co nejméně dat.

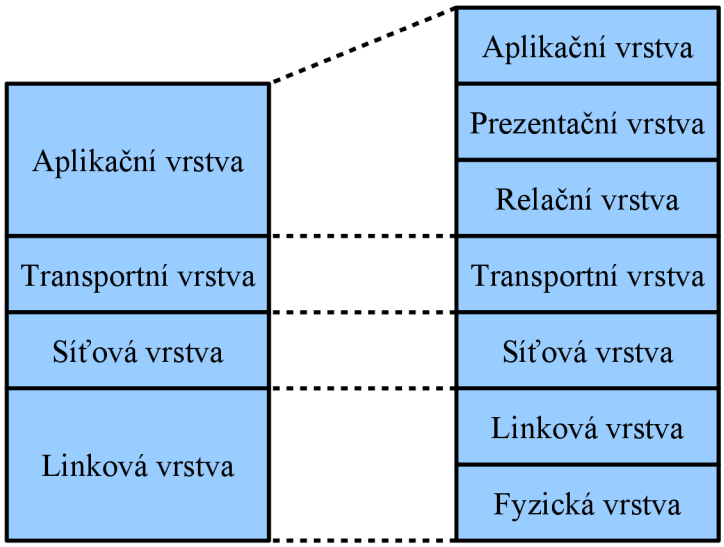
Našim cílem je zjistit a ukázat na místa v Internetové komunikaci, která nesou jen málo informace či žádnou Při návrhu protokolů se mohlo počítat s rozšířeními, která se postupem času nevyužila, či ztratila význam v důsledku jiné technologie, která přišla později. Tím pádem prakticky všechny protokoly v sobě mají jistý objem nepotřebných informací nebo nevyužitých bitů. Na tato místa chceme poukázat. Toho dosáhneme jak analýzou daného protokolu pro zformování předpokladů, tak praktickým ověřením pomocí programu, na jehož vytváření se také soustředíme.
Kapitola 2 se zabývá dnes nejpoužívanějšími protokoly. Popíšeme si je a nastíníme funkci hodnot obsažených v jejich hlavičkách. Porozumění funkce protokolů a jejich principů je důležité pro správné pochopení výsledků získaných měřením. V kapitole 3 vysvětlíme podstatu entropie jako pojmu v informační teorii. Řekneme si také, jak se bude entropie měřit v případě protokolů a jejích polí. Stanovujeme předpoklady výsledků a jejich důvody. Poté se v kapitole 4 zaměříme na implementaci nástroje pro měření entropie a vysvětlíme si principy použité v tomto programu. Zmíníme se o výkonu a nárocích programu. Kapitola 5 potom obsahuje výsledky měření jednotlivých polí. Srovnáváme předpoklady z kapitoly 3 a hledáme vysvětlení, pokud nejsou naplněny. S pomocí nástroje také zkoušíme najít závislosti Internetového provozu v čase.

2 Sítě a protokoly

Internet jako celosvětová síť se v dnešní době využívá pro více a více činností, které se do nedávné doby prováděly fyzicky či jiným způsobem. Také se vyvinulo mnoho systémů propojujících tuto rozsáhlou infrastrukturu. Každá část sítě má jiné požadavky a funkční parametry. Proto bylo navrženo množství protokolů a mechanismů umožňujících komunikaci uvnitř sítí a spolupráci mezi nimi.

Spektrum protokolů a jejich použití se liší v místě použití, např. v místních sítích, v dedikovaných sítích nebo v linkách poskytovatelů připojení (ISP).

V současnosti se v Internetu a v místních sítích využívá velké množství komunikačních protokolů. Pro přehlednost se protokoly kategorizují do vrstev. Kategorizuje se podle modelu *ISO/OSI* nebo *TCP/IP*. Oba rozděluje principy sítě na 7, resp. 4 vrstvy; každá z nich plní jistou funkci a nepotřebuje znalost funkce vrstvy nad ní nebo pod ní. Model *ISO/OSI* byl vytvořen jako standardní model pro počítačové sítě organizací *ISO*. *TCP/IP* model je jednodušší a vychází z funkce Internetu když byl navržen jako armádní síť, a také jak ho známe dnes.



Obr. 1: Relace TCP/IP modelu a ISO/OSI modelu

Obr.1 zobrazuje vztah TCP/IP modelu k ISO/OSI modelu. Zatímco ISO/OSI model je obecnější a abstraktnější, TCP/IP model vychází z funkce tehdejší sítě *ARPANET* vytvořené pro účely Ministerstva obrany Spojených Států. Slučuje některé vrstvy ISO/OSI modelu a zjednodušuje jej. TCP/IP model budeme používat pro kategorizaci protokolů, které si představíme.

2.1 Linková vrstva

Linková vrstva je nejnižší vrstvou v modelu TCP/IP. Zahrnuje v sobě přístup na síťové médium (UTP kabel, koaxiální kabel) a adresování v rámci jednoho síťového segmentu. Zde se můžeme setkat s protokoly jako *Frame Relay*, *ATM, Token Ring* a *Ethernet*. Právě Ethernet je jeden z nejpoužívanějších protokolů na linkové vrstvě v běžných sítích.

2.1.1 Ethernet

Ethernet je soubor síťových technologií definovaných v *IEEE 802.3*. Definuje jak fyzické propojení zařízení, tak způsob komunikace mezi nimi. Každé zařízení má 48-bitovou tzv. *MAC* (Media Access Control) adresu. Protokol zajišťuje komunikaci pouze v rámci jednoho segmentu, tj. mezi dvěma zařízeními. Z pohledu Ethernetu nelze, aby zařízení přeposílalo Ethernetové rámce bez použití protokolu vyšší vrstvy. Ethernet definuje jednoduchou hlavičku o 4 polích, která se dá dále rozšířit novějšími mechanismy jako jsou *VLAN značkování* (angl. *VLAN tagging*) *IEEE 802.1Q*.

2	1	6	6	2	46-1500	4
Preambule	Začátek rámce	Cílová adresa	Zdrojová adresa	Typ/Délka	Data +zarovnání	Kontrolní součet

Tab. 1: Formát hlavičky Ethernet [2]

Zdrojová a cílová MAC adresa (48 bitů)
Určují adresu příjemce a odesílatele, tedy dvě zařízení na síťovém segmentu.

Typ/Délka rámce (16 bitů)
Nese údaj o zapouzdřeném protokolu vyšší vrstvy. Protože bez jiného mechanismu nemáme možnost, jak zjistit protokol vyšší vrstvy, je definováno právě toto pole. Jeho hodnoty jsou přidělovány Tento bude velmi často protokol *IP*, můžeme se ale setkat i s jinými protokoly, např. *IPX/SPX*. V běžných sítích ale provoz *IP* převládá, proto jeho hodnota bude převážně konstantní.

2.2 Síťová vrstva

Síťová vrstva je zodpovědná za dopravu paketů mezi sítěmi. Na této vrstvě pracují směrovače, které zajišťují transport paketů mezi různými adresními prostory. Dnes je protokol IP dominantním v síti Internet.

2.2.1 IPv4

IP (Internet Protocol) verze 4 se od doby svého nasazení rychle rozšířil a lze říci, že téměř veškeré služby Internetu využívají tento protokol ke své funkci. Pro identifikaci zařízení v sítích definuje 32-bitovou *IP adresu* obvykle zapisovanou ve formátu čtyř oktetů v desítkové soustavě oddělených tečkami, např. 208.67.222.222. Pokud se pohybujeme v rámci veřejných adres (adresy viditelné a dosažitelné z Internetu), IP nám umožňuje s těmito adresami komunikovat. IP také podporuje tzv. fragmentaci datagramů. Pokud přenosová linka podporuje pouze jistou velikost rámců, umožňuje protokol IP rozdělit datagram tak, aby bylo možné rámec přenést po této lince po částech. Fragmenty se skládají do původního datagramu až v cíli.

Protokol definuje následující formát hlavičky:

<table>
<tr><th>4</th><th>8</th><th>16</th><th>17</th><th>18</th><th>19</th><th>32</th></tr>
<tr><td>Verze</td><td>Délka hlavičky</td><td>Typ služby</td><td colspan="4">Celková délka</td></tr>
<tr><td colspan="3">Identifikace</td><td>-</td><td>DF</td><td>MF</td><td>Posun fragmentu</td></tr>
<tr><td colspan="2">TTL</td><td>Protokol</td><td colspan="4">Kontrolní součet hlavičky</td></tr>
<tr><td colspan="7">Zdrojová adresa</td></tr>
<tr><td colspan="7">Cílová adresa</td></tr>
<tr><td colspan="7">Volby + zarovnání</td></tr>
<tr><td colspan="7">Data</td></tr>
</table>

Tab. 2: Formát IP datagramu [2]

Verze a délka hlavičky (4 + 4 bity)

Tato dvě pole označují verzi protokolu a délku hlavičky protokolu. Při vyhodnocování protokolu IPv4 a IPv6 se dá srovnávat pouze pole *Verze*, protože struktura verze 4 a 6 je dále rozdílná. Tudíž pro IPv4 bude mít vždy hodnotu 4.

Délka hlavičky je opět konstantní, až na ojedinělé případy, kdy hlavička obsahuje volitelné položky *Voleb*.

Typ služby

Toto pole může být použito k rozlišení provozu, u kterého požadujeme *QoS* (*Quality of Service*), tedy určení např. priority. Tímto můžeme upřednostňovat provoz, který vyžaduje rychlost doručení.

Identifikace (16 bitů)

Nelze očekávat, že každá přenosová linka dokáže přenášet Ethernetové rámce tak dlouhé, jako jsou při jejich vytváření. Některé linky mají limit na velikost rámců, tzv. MTU (Maximum Transfer Unit) – maximální velikost přenášené jednotky. Protokol IP proto zavádí mechanismus fragmentace, který umožňuje v zařízení rozdělit datagram na menší části do více rámců tak, aby takto rozdělený datagram bylo možno přenést po dané lince. Protože v sítí dozajista funguje mnoho zařízení, které využívají protokol IP a mohou fragmentovat datagramy, je třeba nějakým způsobem zajistit rozlišení jednotlivých fragmentů a určit, které patří kterému datagramu. K tomuto účelu slouží právě pole *Identifikace*. Fragmenty náležící stejnému datagramu mají stejnou hodnotu tohoto pole.

Fragmentace se potom dále řídí příznaky *DF* (*Don't Fragment* – nefragmentuj), *MF* (*More Fragments* – další fragmenty) a pole *Ofset fragmentu*.

Ofset fragmentu (13 bitů)
Jak bylo řečeno, slouží toto pole pro řízení mechanismu fragmentace. Říká, jaký je ofset dat v datech původního datagramu vzhledem k prvnímu fragmentu. Pro nefragmentované datagramy má pole hodnotu 0.

Time to Live (8 bitů)
Pole má velikost 1 bajt a určuje, kolik skoků (angl. *hops*) může paket projít, než je zahozen. Toto pole je většinou konstantní.

Protokol (8 bitů)
Udává protokol vyšší vrstvy, který je v IPv4 paketu zapouzdřen. Seznam platných hodnot je udržován organizací *IANA* (*Internet Assigned Numbers Authority*).

2.2.2 IPv6

Protokol IP verze 6 byl navržen a posléze specifikován jako odpověď na rychle se vyčerpávající adresový prostor v IP verze 4. Vzhledem k tomu, že vznikl mnohem později než IPv4, mohl se návrh vyvarovat nedostatků, které měl protokol IPv4. Návrh počítal i s možnými rozšířeními.
IPv6 rozšiřuje velikost zdrojové a cílové adresy na 128 bitů (oproti 32 bitům v IPv4), což by mělo zajistit dostatečný adresový prostor na dlouhou dobu. Mění se zápis adresy na 8 čtyřmístných hexadecimálních čísel oddělených dvojtečkou, IPv6 adresa tedy může být `2001:47ff:0000:0000:0055:3187:e45b:0070`.

4	12	16	24	32
Verze	Priorita	Značka toku		
Délka dat			Další hlavička	Limit skoků
Zdrojová adresa (128 bitů)				
Cílová adresa (128 bitů)				

Tab. 3: Formát hlavičky IP verze 6 [2]

Verze (4 bity)
Určuje verzi IP verze 6, tedy hodnota pole je 6.

Priorita (8 bitů)
Priorita dovoluje zdroji paketů specifikovat prioritu provozu.

Značka toku (10 bitů)
Umožňuje označit speciální toky dat, aby je bylo možno rozpoznat a patřičně směrovat na IPv6 směrovačích. Funkcí je podobné poli Typ služby v IPv4.

Délka dat (16 bitů)

Určuje délku zapouzdřených dat v paketu.

Další hlavička (8 bitů)

Má podobnou funkci jako pole Protokol v IPv4. Podle hodnoty je to buď protokol vyšší vrstvy zapouzdřený v paketu, nebo typ hlavičky bezprostředně následující po této hlavičce. Každá další hlavička obsahuje totéž pole se stejným účelem. Poslední hlavička obsahuje typ zapouzdřeného protokolu. Takto se mohou hlavičky řetězit, aniž by bylo potřeba jiných hodnot pro určení délky hlavičky, protože tento údaj není potřeba. Hodnoty jsou sdílené s hodnotami pro pole Protokol v IPv4.

Limit skoků (8 bitů)

Zabraňuje paketům, aby nekonečně kroužily v síti. Funkce je totožná s pole TTL v IPv4. Každý směrovač toto pole sníží o 1. Pokud hodnota dosáhne nuly, je paket zahozen.

Zdrojová a cílová adresa (128 + 128 bitů)

Identifikuje odesílatele a příjemce paketu jednoznačně v síti (Internetu).

Hlavička byla značně zjednodušena oproti verzi 4. Obsahuje hlavně informace, které využívají směrovače a jiná zařízení, všechny ostatní informace jsou přidány ve formě hlaviček připojených za IPv6 hlavičku. O jakou hlavičku se jedná je určeno polem *Další hlavička*.

2.3 Transportní vrstva

Transportní vrstva má za úkol přijímat pakety od nižší vrstvy a předávat je aplikacím v aplikační vrstvě. Může zajišťovat spolehlivé doručení dat, řídit jejich tok (předcházet a řešit zahlcení) a udržovat spojení s druhou stranou. V této vrstvě jde o adresování aplikace-aplikace.

2.3.1 Protokol TCP

Jak už název napovídá, protokol TCP (*Transmission Control Protocol*) se věnuje zejména spolehlivému doručení a řízení toku dat. Je tzv. *connection-oriented*, což znamená, že udržuje každé spojení jako relaci, dokud není ukončena. Mechanismus zajištění spolehlivého doručení využívá parametr *okno*, který vymezuje právě zpracovávané nepotvrzené segmenty. V průběhu komunikace se okno posouvá podle již potvrzených segmentů. V případě, že nedojde potvrzení do jisté doby, je segment znovu poslán. Příjemce si vede podobný záznam o správně přijatých segmentech. Tímto se zabezpečí, že druhá strana přijme všechny segmenty, ve správném pořadí a bez chyb.

Protokol TCP definuje následující formát hlavičky:

<table>
<tr><td>4</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>32</td></tr>
<tr><td colspan="8">Zdrojový port</td><td>Cílový port</td></tr>
<tr><td colspan="9">Sekvenční číslo</td></tr>
<tr><td colspan="9">Potvrzovací číslo</td></tr>
<tr><td>Posun dat</td><td>-</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Okno</td></tr>
<tr><td colspan="8">Kontrolní součet</td><td>Urgentní ukazatel</td></tr>
<tr><td colspan="9">Volby + zarovnání</td></tr>
<tr><td colspan="9">Data</td></tr>
</table>

Tab. 4: Formát TCP segmentu [2]

Zdrojový a cílový port (16 + 16 bitů)

Identifikují aplikace, kterým náleží zapouzdřená data. V případě např. komunikace klienta se serverem, bývá zdrojový port náhodně vybrán aplikací při vytváření spojení a cílový port je často některý ze známých (angl. *well-known*) portů (např. 80 pro HTTP).

Sekvenční číslo (32 bitů)

Sekvenční číslo označuje sekvence dat posílaných druhé straně. Má pole SYN hodnotu 1, je toto pole prvním číslem v sekvenci. Pokud je SYN nulové, je to sekvenční číslo akumulované od začátku komunikace. Vyjadřuje počet již odeslaných bajtů.

Potvrzovací číslo (32 bitů)

Pokud je bit ACK nastaven na 1, přijímací strana potvrzuje všechna přijatá data do tohoto sekvenčního čísla. Počáteční sekvenční číslo se potvrzuje stejným způsobem.

Ofset dat (4 bity)

Ofset dat určuje velikost hlavičky UDP segmentu. Je zavedeno kvůli nepovinným volbám. Stejně tak říká, od kterého oktetu od začátku segmentu začínají zapouzdřená data.

Okno (16 bitů)

Okno umožňuje odesílateli, aby specifikoval velikost své vyrovnávací paměti pro hlídání potvrzených segmentů.

TCP je předurčen pro aplikace, které potřebují zajistit spolehlivost přenosu a řízení toku. Je využíván v aplikacích, u kterých by výpadek segmentu znamenal komplikace při funkci programu nebo systému.

2.3.2 Protokol UDP

Protokol UDP (User Datagram Protocol) na rozdíl od TCP neřeší spolehlivé doručování. Nezaručuje, že všechna data budou přijata druhou stranou, poskytuje tzv. doručení s největším úsilím (angl. *best effort delivery*). Je to velmi jednoduchý protokol pro nenáročné aplikace.

Hlavička protokolu UDP je definována:

16	32
Zdrojový port	Cílový port
Délka	Kontrolní součet
Data	

Tab. 5: Formát UDP datagramu [2]

Zdrojový a cílový port

Kontrolní součet (checksum)
Slouží pro ověření neporušenosti dat. Toto pole je u protokolu UDP zajímavé tím, že je už z definice nepovinné.

Jak lze vidět z hlavičky, UDP je velmi jednoduchý protokol, který nepodporuje žádné řízení toku či kontrolu úspěšnosti a správného pořadí doručení. Je vhodný pro služby, kde je důležitější rychlost než kvalita spojení a kde nezáleží na několika ztracených paketech. Jde například o různé proudové přenosy (angl. *streams*), obrazové či zvukové. Je na aplikaci, aby tyto funkce zajistila ve své vlastní režii, pokud některé z nich vyžaduje.

2.4 Aplikační vrstva

Aplikační vrstva zajišťuje převod dat do správné reprezentace, správa sezení a předání dat aplikaci. Jedním z nejznámějších protokolů aplikační vrstvy je HTTP nebo POP3.
Protokoly aplikační vrstvy se nebudeme blíže zabývat, protože většinou postrádají uniformní definici hlavička-pole a jejich analýza je odlišná. Také zde nelze hovořit o entropii jako o počtu bitů potřebných k zakódování informace, poněvadž prvky protokolů často nemají pevné bitové velikosti a jejich hodnoty jsou též proměnné. Příkladem toho je protokol HTTP.

Technické informace získány z [2] a [3].

3 Entropie a její měření

3.1 Přenos informace

Přenos informace se obvykle realizuje přes různé druhy spojů a cest. Vzhledem k narůstajícím nárokům pro rychlost je jedna z možných cest minimalizace objemu přenesených dat. Data mohou být například zkomprimována nebo jejich objem zmenšen jejich analýzou. V každém protokolu jsou jistě data, která není nutné přenášet či k jejich přenesení stačí menší objem. K tomuto účelu lze využít entropii.

3.2 Entropie

V 70. letech 18. století Ludwig Boltzmann a J. W. Gibbs definovali entropii ve statistické termodynamice. Je definována jako míra neurčitosti systému, tj. jak jsou v něm rozprostřeny pravděpodobnosti jeho tzv. mikrostavů. Entropie jako pojem se rozšířila do dalších odvětví, jedním z nichž je informační teorie.

3.2.1 Entropie v informační teorii

Entropie v informační teorii byla popsána v r. 1948 Claudem E. Shannonem. Je definována jako míra nejistoty, která se váže k náhodné proměnné. Vyjadřuje, kolik bitů je nutných k bezeztrátovému zakódování dané informace. Tím pádem přímo říká, jaké množství užitečné informace obsahuje. Tuto vlastnost lze využít pro bezeztrátovou kompresi. [10]

Máme prostor zpráv M. Potom entropie M je:

$$H = -\sum (p_i) \log_n (p_i)$$

kde p_i je pravděpodobnost výskytu zprávy m_i z M.

Vzhledem k tomu, že logaritmus nulové pravděpodobnosti není definován, formálně definujeme že $0 \cdot \ln(0) \equiv 0$ [1]

Základ logaritmu n určuje, počet možností na jednu pozici proměnné a v jakých jednotkách je získaný výsledek. V našem případě měříme hodnoty uložené v počítači ve dvojkově soustavě, tudíž základ logaritmu bude 2 (bit má dva možné stavy). Jednotka výsledku bude počet *bitů*.

V případě, že všechny zprávy z prostoru M mají stejnou pravděpodobnost p_i, lze výpočet entropie redukovat na

$$H = \log_n(|M|)$$

kde $|M|$ je kardinalita prostoru M. V tomto případě je to počet všech zpráv. [10]

3.3 Měření entropie

Měření se bude řídit podle logické struktury paketů. To znamená, že entropii budeme počítat pro každé pole paketu jistého protokolu. Tímto můžeme zjistit jejich informační hodnotu. Druhý přístup by zahrnoval měření bez jakékoliv znalosti struktury paketů, zjištěné výsledky by však nebyly dobře interpretovatelné a jejich vypovídací hodnota by byla malá.

Entropie daného pole vlastně určuje, kolik bitů z celkové velikosti pole je statisticky využito pro užitečná data. V případě diskrétních hodnot, ze kterých je paket tvořen, bude měření vypadat asi takto: Zpracovávat se bude pouze předem definovaná množina protokolů. Program bude procházet pakety a pro každé pole protokolu bude uchovávat záznam pro každou unikátní hodnotu vyskytující se v tomto poli. Také bude ukládat informaci, kolikrát se daná hodnota v poli objevila a kolik paketů již bylo zpracováno. Poté se vypočítá pravděpodobnost výskytu každé unikátní hodnoty v poli.

Vycházíme z klasického vztahu pro pravděpodobnost náhodného jevu:

$$p = \frac{počet\ úspěšných\ pokusů}{celkový\ počet\ pokusů}$$

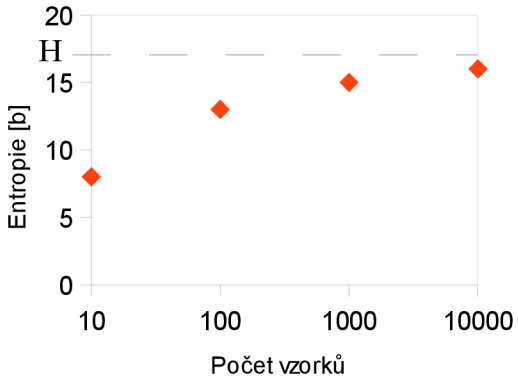
Pravděpodobnost výskytu unikátní hodnoty tedy vypočítáme:

$$p = \frac{počet\ výskytů\ hodnoty}{celkový\ počet\ hodnot\ v\ poli}$$

Potom pro každé pole ve formátu protokolu se entropie bude rovnat

$$H = -\sum p \cdot \log_2 p$$

Maximální entropie nastává, pokud se počet unikátních hodnot pole rovná počtu všech možnosti hodnot v daném poli. Tento případ však v praxi nikdy nenastane, protože při používání protokolu není možnost sledovat vygenerované hodnoty a často to ani neumožňuje význam pole. Vzhledem k tomu, že entropie je měřena z jistého vzorku provozu, je i přesnost vypočtené hodnoty závislá na kvalitě a počtu vzorků. Pokud entropii počítáme z nízkého počtu vzorků, mohou být výsledky velmi zkreslené.



Graf 1: Entropie v závislosti na počtu vzorků

Graf 1 ukazuje zpřesňování výsledků výpočtu entropie v závislosti na počtu měření. *H* označuje přesnou hodnotu entropie při nekonečně mnoho vzorcích s maximálním rozptylem hodnot. Tuto hodnotu nelze reálným měřením získat, bylo by možné ji však aproximovat.

3.4 Entropie a síťové protokoly

Jak jsme již uvedli, entropie určuje počet bitů potřebných k zakódování množiny dat. Nízká, resp. vysoká entropie ukazuje, že množina možných hodnot je malá, resp. velká. Pokud je entropie daného pole protokolu výrazně menší než rezervovaná délka, může to znamenat, že na daném místě dochází ke zbytečnému přenášení dat. Samozřejmě to ale může znamenat, že vzorek, ze kterého bylo měření provedeno, nebyl dostatečně vypovídající o daném poli, jelikož přesnost výpočtu se zvyšuje s počtem vzorků. Proto je třeba věnovat pozornost údaji, kolik hodnot bylo zpracováno.

Nyní si již dříve uvedené protokoly podrobněji rozebereme s ohledem na entropii.

3.4.1 Ethernet

Vezmeme v úvahu *Zdrojovou a cílovou MAC adresu*. Většina MAC adres je globálně unikátní, to znamená, že nelze vytvořit konflikt adres na 2. vrstvě v jakékoli síti. Počet možných hodnot je zde 2^{48} (délka 48 bitů), prakticky využita je pouze podmnožina těchto hodnot z důvodu nutnosti systému pro zajištění právě unikátnosti adres. Lze předpokládat, že počet možných adres v Ethernetových rámcích odchycených na jedné síti bude maximálně právě počet zařízení v této síti. Pole *Typ* značí zapouzdřený protokol vyšší vrstvy, jeho hodnoty a v důsledku i jeho entropie bude přímo záviset na protokolech, které jsou (byly) využívány v datovém vzorku. Předpokládáme, že IP verze 4 bude dominantním protokolem Síťové vrstvy.

3.4.2 IPv4

Když se zaměříme na pole *Verze*, zjistíme, že bez ohledu na data nebo parametry je toto pole je konstantní od doby, co byl IP protokol nasazen do praxe. Jelikož neexistuje žádná jiná verze IPv4, je zřejmé, že entropie tohoto pole bude vždy nulová. Podobně to může být i u *Velikosti hlavičky*. Vzhledem k tomu, že *Volby* se v IP už v podstatě nepoužívají, je možné, že *Velikost hlavičky* bude v každém měření konstantní. Tím pádem může být entropie nulová, podobně jako u pole *Verze*. K rozlišení typů provozu slouží *Typ služby*. Zatím se nedá s jistotou říct, jak bude využíváno. Pole *Identifikace* slouží k jednoznačné identifikaci IP fragmentů. Jelikož se protokol snaží o minimální kolize, velmi pravděpodobně budou hodnoty mít velký rozptyl a tím pádem i entropii. Uvažujeme-li pole *Protokol*, je pravděpodobné, že velká část vzorků bude obsahovat hodnoty určující protokoly TCP a UDP. Lze očekávat, že entropie tohoto pole se bude blížit ke 2. Bude ale zřejmě vyšší, jelikož TCP a UDP nejsou jediné užívané protokoly. *Posun fragmentu* určuje relativní pozici IP fragmentu vůči prvnímu fragmentu. Očekáváme, že na běžných linkách (bez velkých limitů na velikost rámců) nebude počet fragmentovaných paketů velký, nebude zde mnoho rozdílných hodnot (bude převládat hodnota 0) a entropie bude nízká. Pro řešení nekonečného kolování paketů v sítích slouží pole *Time to Live (TTL)*. Různá zařízení mohou mít různou výchozí hodnotu *TTL*. Je tedy možné, že narazíme na množství různých hodnot.

3.4.3 IPv6

Stejně jako IPv4, obsahuje i jeho následník pole *Verze*. Toto obsahuje opět konstantní hodnotu 6. Podle definice protokolu toto pole má vždy tuto hodnotu, proto jeho entropie bude nulová. Pole *Priorita*, podobně jako *Značka toku*, umožňují značení provozu a aplikaci pravidel pro jeho řízení a upřednostňování některých toků dat. Jeho entropie bude záviset na stupni rozšíření protokolu IPv6 a podporou v zařízeních. *Délka dat* bude záviset na přenášených datech. Pole *Další hlavička* značí protokol vyšší vrstvy nebo rozšiřující hlavičku. Rozmanitost těchto hodnot bude záviset na současně používaných protokolech na vyšší vrstvě. Podobně jako u IPv4, očekáváme rozšíření protokolů TCPv6 a UDPv6. Podobně jako u *TTL*, *Limit skoků* může obsahovat různé hodnoty.

3.4.4 TCP

Mezi poli *Zdrojový a cílový port* nemusíme dělat rozdíl při analýze, protože nerozlišujeme, od koho a komu byl segment poslán. Často jde o komunikaci klient-server a naopak, kdy klient volí náhodný port (1024 – 65535) a posílá segmenty na porty serveru (1 – 1024). Pro každé TCP spojení se vybírá jiný klientský port. Pokud zachytíme segmenty v rámci jednoho zařízení, může být zdrojový nebo cílový port stejný pro všechny segmenty. Entropie by byla nulová. My měříme entropii v rámci mnoha zařízení a spojení pro zajištění dobré vypovídací hodnoty zjištěných údajů. Očekáváme, že značná část TCP provozu bude směřovat na servery poskytující HTTP služby. Menší část budou dotazy pro DNS servery. Odhadujeme. že entropie každého pole nebude větší než polovina jejich velikostí v bitech, právě díky opakování typických portů služeb (angl. *well-known ports*). *Sekvenční a potvrzovací číslo* řeší značení a potvrzování odesílaných a přijímaných segmentů. Je to mechanismus pro zajištění spolehlivého doručení. Toto číslo se generuje náhodně při započetí spojení. Proto jeho hodnoty budou mít pravděpodobně větší rozptyl (i když čísla mohou na sebe

navazovat) a tím i větší entropii, blížící se velikosti pole. *Posun dat* značí velikost TCP hlavičky. Naše očekávání je, že se volby (jejichž přítomností se velikost hlavičky zvětšuje) často nepoužívají, a proto by mohla být její hodnota většinou konstantní s nízkou výslednou entropií.

3.4.5 UDP

Pro pole *Zdrojový a cílový port* platí stejné poznatky jako u stejnojmenných polích u protokolu TCP. Z hlediska hodnot očekáváme, že se na provozu bude významnou měrou podílet služba DNS. *Délka* určuje velikost celého UDP datagramu. Vzhledem k tomu, že data v UDP datagramech mohou být různě dlouhá, nemůžeme s velkou jistotou říci, zda se hodnoty budou lišit či nikoliv. Je možné, že při měření zjistíme hodnotu s dostatečným výskytem, kterou by se dalo označit za výchozí velikost datagramů.

3.4.6 Společné vlastnosti

Všechny uvedené protokoly disponují polem, které obsahují kontrolní součet buď hlavičky protokolu nebo dat. Algoritmus kontrolního součtu by měl pokud možno generovat stejné výsledky pouze pro stejná data. Funkce, které tyto výpočty provádí ovšem nejsou dokonalé, proto se můžeme setkat s různými daty generujícími stejné výsledky. Předpokládá se ale, že funkce má dostatečné velký obor hodnot, a proto se dá s jistou pravděpodobností očekávat, že pro každou rozdílnou datovou jednotku protokolu dostaneme rozdílný výsledek kontrolního součtu. Je pravděpodobné, že právě tato pole budou mít nejvyšší entropii ze všech, díky jejich účelu. V protokolech, jejichž pole vyjadřující adresy jsou určeny více bity (např. 16 a více) bude pravděpodobně v závislosti na funkci protokolu obsahovat množství unikátních hodnot. Dá se říci, že měření entropie (a tím informační hodnoty) těchto polí má spíše informativní charakter, protože hodnoty se síť od sítě mění, ale velikost pole sama o sobě musí zůstat nezměněna.

4 Implementace

Nyní si popíšeme důležité části aplikace pro měření entropie. Program je jednoúčelový nástroj pro měření entropie jistého protokolu v množině paketů. Je napsán v jazyku C. Důvodem pro tento jazyk je jeho nižší úroveň a proto vetší možnost optimalizace a tím pádem rychlosti. Umožňuje také rozšíření znalostí protokolů pomocí knihoven (zásuvných modulů).

4.1 Závislost na knihovnách

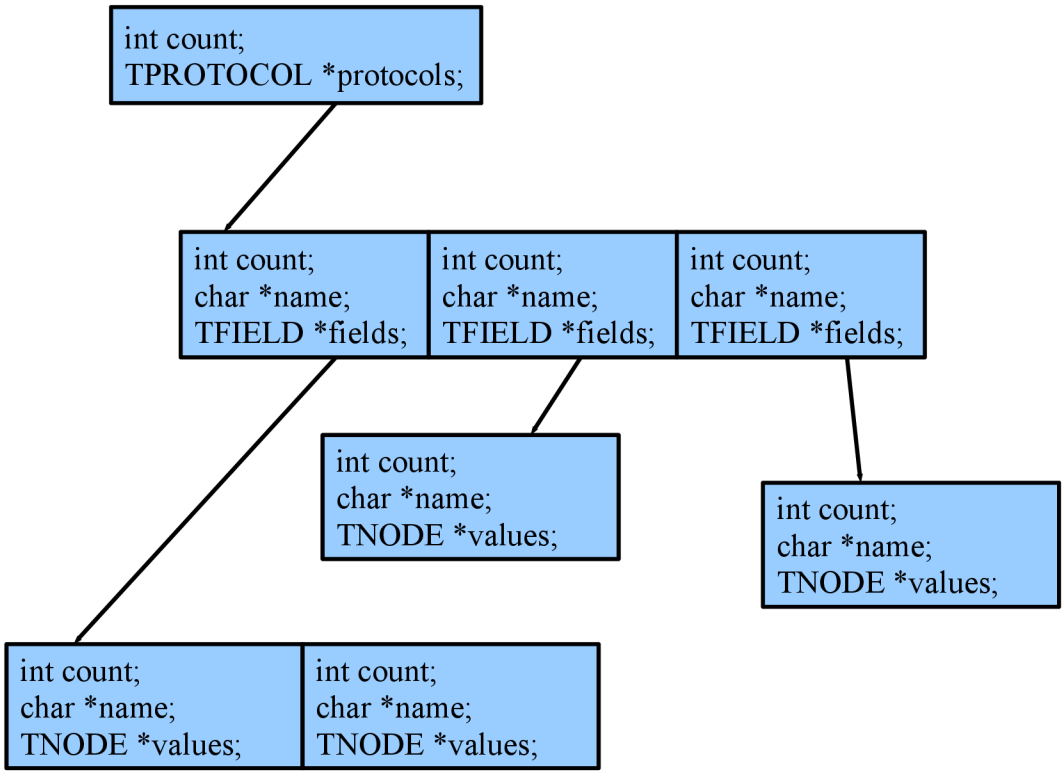
Program využívá knihovny *libPcap* [5] pro otevření datového souboru na jeho vstupu a procházení souborem. Tato knihovna je volně dostupná pro různé platformy. Důvodem pro použití knihovny je zejména fakt, že formát PCAP se může časem měnit a v tomto případě by program přestal fungovat. Očekává se aktualizace knihovny ze strany uživatele, tento požadavek ovšem nelze vynutit. Dále program užívá standardní knihovny obsažené v sadě kompilátorů GCC [9].

4.2 Zpracování souboru

Jak již bylo zmíněno, program využívá *libPcap* [5] pro procházení souborem. Program prochází pakety podle zadaných kritérií a sleduje zadané pole protokolů. Soubor je zpracován postupně, ale pouze jednou. Program si vede informace o protokolech a statistiky o výskytu hodnot
v paketech
v paměti.

4.2.1 Statistiky pro výpočet

Je třeba vést si informace o požadovaných protokolech a polích pro potřeby měření. Program potřebuje vědět, které protokoly má zpracovávat a o kterých polích si má ukládat statistiky. Pro uložení statistik a informací o protokolech, polích a hodnotách se využívá stromová struktura, kterou ilustruje *Obr.2*.



Obr. 2: Struktura pro ukládání informací o protokolech a jeho polích

Ve struktuře statistik jsou uloženy názvy protokolů a údaj o počtu jeho (zvolených) polí ve strukturách `TPROTOCOL`. Struktura protokolů obsahuje ukazatel na pole struktur informací o jeho polích `TFIELD`. Podle těchto údajů se volají funkce z knihoven (knihovna je identifikována názvem protokolu) a průchod statistikami při výpočtu.

Stejné informace se ukládají i o jednotlivých polích. Podle jména se získávají hodnoty polí protokolů. Položka `values` ukazuje na odlišnou strukturu držící hodnoty nalezené při procházení souboru. Pro každé sledované pole v protokolu se vede statistika, která se využívá ve finálním výpočtu. Ukládá se přečtená hodnota (pro účely vyhledávání, popř. rozšířeného výpisu). Dále struktura udržuje počet unikátních hodnot, na které se během zpracování narazilo, a počet hodnot celkem. Tyto údaje se posléze využívají při výpočtu entropie.

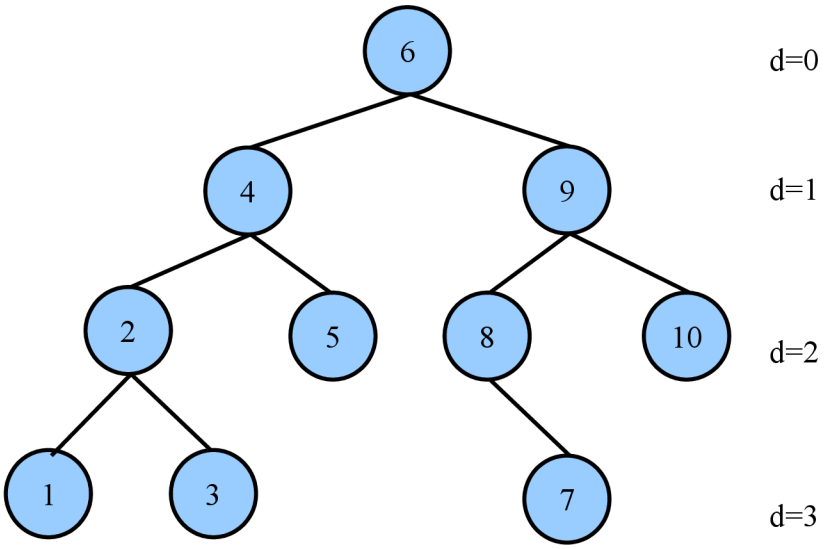
Výpočet entropie spočívá v procházení pole struktur pro statistiky polí a výpočet pravděpodobností výskytu každé hodnoty. Součet těchto pravděpodobností se rovná entropii pole v protokolu.

Ukládání hodnot

V prvotní implementaci byly hodnoty pro každé pole ukládány do pole o dané velikosti, které se případně potřeby zvětšilo. To se ukázalo jako velmi neefektivní, protože v případě velkého počtu zpracovávaných paketů a polí obsahujících většinou unikátní hodnoty často docházelo k prohledávání

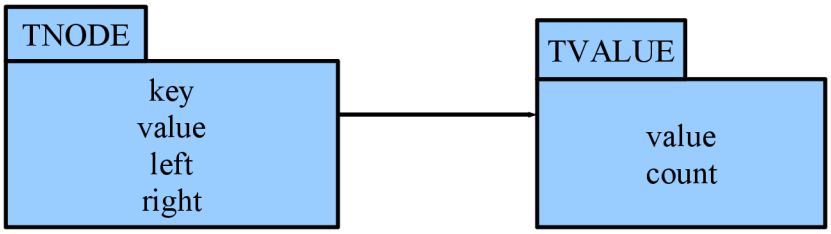
zbytečně velké oblasti paměti. To způsobovalo dlouhé doby běhu programu. Hodnoty jsou ukládány v binárním vyhledávacím stromu (BVS).

Každý uzel stromu obsahuje informace typické pro režii stromu (ukazatele na potomky), klíč, podle kterého se provádí vyhledávání (shodný s uloženou hodnotou) a také ukazatel na položku, která obsahuje informace o hodnotě a jejím počtu. Obvykle levý potomek má nižší hodnotu klíče než rodič a pravý vyšší.



Obr. 3: Ukládání hodnot jako strom

Vyhledávání se potom redukuje na porovnávání klíčů. Také nemusíme prohledávat části stromu, v druhém potomkovi. Např. pokud se porovnáním zjistí, že se bude pokračovat k levému potomkovi, už nemusíme prohledávat pravou stranu stromu, protože je jisté, že se tam hledaný uzel nenachází. V nejhorším případě prohledáme tolik prvků, jako je hloubka stromu *d*. [4]



Obr. 4: Detail uzlu stromu

Důvodem pro použití ukazatele na strukturu hodnoty místo zakomponování těchto údajů do uzlu stromu je oddělení definic uzlů stromu a definic statistických struktur v modulech. Pro zpracování dat uložených ve stromě je třeba nejdříve strom jistým algoritmem projít a data uložit do lépe přístupné struktury. Touto strukturou je právě pole ukazatelů na položky typu *TVALUE*. Operace serializace zvyšuje paměťovou náročnost, je však pro každé pole prováděna pouze jednou na konci měření, výkonnostní dopad je proto malý. Pro serializaci stromu využíváme metodu průchodu stromem *InOrder*. Její vlastnost je, že hodnoty jsou po průchodu uspořádány od nejmenší po největší.

Jistou nevýhodou binárního vyhledávacího stromu je oproti např. poli hodnot je vyšší paměťová režie při vytváření a nutnost průchodu stromu pro získání hodnot pro další zpracování. Nespornou výhodou je však rychlost vyhledávání a vkládání položek, což v tomto případě je pro nás rozhodující faktor při výběru struktury.

4.2.2 Rychlost a paměťové nároky

Měření rychlosti bylo provedeno na jednom sledovaném poli na počtu 1000000 datových jednotek s co největším počtem unikátních hodnot (tím pádem i uzlů stromu). Při průměrném počtu 700 000 unikátních hodnot program pracoval 14 sekund a spotřeboval 30 MB paměti.
Velikost paměti je zejména určena informacemi, které se ukládají v uzlu společně s informací o hodnotě jako režie spojená s binárním vyhledávacím stromem.

4.3 Podpora protokolů

Program již podporuje několik nejznámějších protokolů – Ethernet, IPv4, TCP a UDP. Další protokoly lze implementovat jako dynamické knihovny. Každá musí podporovat sadu funkcí nutnou ke správnému běhu programu. Nejsou-li k dispozici, program skončí s chybou.

Pokud se přidává nový protokol, je nutné, aby se tento fakt také promítl do protokolu, ve kterém je tento zapouzdřen. Nižší protokol určuje, jestli je zapouzdřený protokol podporován, a podle toho vrací jeho jméno.

4.4 Omezení a možná rozšíření

Program má také jistá omezení. Předpokládá se, že všechna data jsou odchycena na sítích Ethernet. Proto program nepodporuje jiné protokoly Linkové vrstvy než je Ethernet. Podporuje však *RAW IP*, což jsou soubory dat, které obsahují přímo IPv4 datagramy bez zapouzdření Ethernetem. Dále program nedokáže správně zpracovat pole o velikosti vetší než 64 bitů. Je to z důvodu ukládání hodnot v 64 bitové podobě pro co největší použitelnost pro různá pole.

Program by bylo možné rozšířit pro podporu dalších exotičtějších typů linek, např. Token Ring, ATM nebo FDDI. Dále by mohla být rozšířena kompatibilita s protokoly proměnných polí. To vyžaduje přepsání kódu zpracovávajícího pole na začátku procházení souboru. Šlo by zefektivnit ukládání hodnot specifikací datového typu, aby se jednobitové hodnoty neukládaly do 64 bitového celočíselného datového typu. Otázku ale je, zda se vyplatí rozlišovat hodnoty menší než 32 bitů, pokud daná architektura nepracuje s menší paměťovou jednotkou než 32 bitů. Potom by to znamenalo spíše výkonovou ztrátu.

5 Výsledky měření

Doposud jsme si definovali pojem entropie, rozebrali často využívané protokoly a uvedli některé předpoklady výsledků měření. Nyní budeme provádět vlastní měření. Zdrojem dat budou soubory paketů odchycených na vysokorychlostních páteřních linkách z projektu *MAWI Working Group* (http://mawi.wide.ad.jp) a z *CAIDA – Cooperative Association for Internet Data Analysis* (http://www.caida.org). Pro měření jednotlivých polí budou použita data ze stejného měsíce (týdne) a výsledná entropie bude počítána jako průměr dílčích za účelem odstranění možných anomálií a špiček, které se jinak v Internetovém provozu mohou vyskytnout. Měření se bude provádět na vzorcích po 100, 10000 a 1000000 paketů daného protokolu pro názornost možné změny výsledků a zvětšování vypovídací hodnoty. Očekáváme potvrzení (vyvrácení) předpokladů, které jsme si stanovili v kapitole 3.4.

Nejdříve budeme měřit některá pole protokolů jako taková. Vyslovíme předpoklady a porovnáme je s výsledky měření, popř. se pokusíme vysvětlit, proč se předpoklad nenaplnil.

5.1 Ethernet

Zdrojová, cílová MAC adresa (48 + 48 bitů)

Počet paketů	Počet unikátních hodnot	Entropie
100	2	0,746355
10 000	2	0,955410
1 000 000	2	0,952985

Tab. 6: Výsledky pro Zdrojovou a cílovou MAC adresu

Pozn.: Výsledky jsou uvedeny pouze pro zdrojovou MAC adresu, jsou však shodné s výsledky pro cílovou MAC adresu.

Dalo by se předpokládat, že v poli zdrojové nebo cílové adresy bude více než dvě unikátní hodnoty. Připomeňme si ale funkci Ethernetu: rámce jsou doručovány pouze v jednom síťovém segmentu (zařízení-zařízení). Pokud jsou tato zařízení síťový přepínač (angl. *switch*) nebo počítače, není možné, aby se mezi adresami objevila jiná než adresa jednoho nebo druhého zařízení. Toto je možné sledovat pouze v síti, kde pracuje síťový rozbočovač (hub), který rozesílá rámce všem, protože si nevytváří asociace port-MAC adresa.

Mohli bychom říci, že přenášet plných 48 bitů adresy nemá smysl, když se v moderních sítích v těchto adresách střídají dvě adresy v rámci jednoho segmentu. Na první pohled by na adresování mohl stačit jeden bit. Nastává zde ale problém, jak síťový adaptér pozná, která hodnota je jeho a která patří adaptéru na druhé straně média. Pravděpodobně by to vyžadovalo ruční konfiguraci či nějaký mechanismus „domluvy“ mezi síťovými adaptéry. Další problém jsou rozbočovače a přepínače.

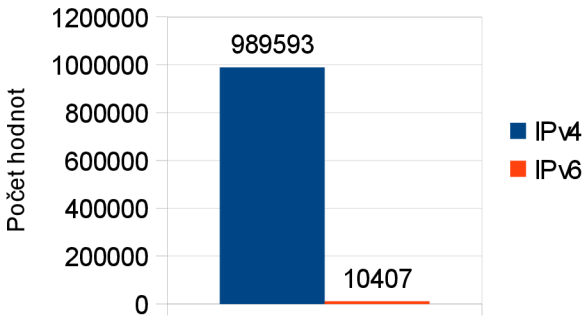
Rozbočovač rozesílá Ethernetové rámce všem připojeným zařízením. V této situaci by zařízení nemohla poznat, komu daný rámec patří, protože více zařízení by mělo stejné adresy. Přepínače si vedou tabulky adres pro přepínání, proto by měly obdobné potíže, protože by nevěděly, kam mají daný rámec poslat. Nakonec docházíme k závěru, že je potřeba používat adresy, které pokud možno nebudou mít konflikty s jinými adresami v síti. Proto existují MAC adresy, které se skládají z OUI (*Organizationally Unique Identifiet*) – unikátní identifikátor organizace a části adresy specifické pro síťový řadič. Čísla OUI spravuje IEEE, specifická čísla pro řadič si volí výrobce. Tímto je zajištěna globální jedinečnost adresy a vyhnutí se konfliktům na Linkové vrstvě.

Typ(16 bitů)

Pole *Typ* určuje zapouzdřený protokol Síťové vrstvy. Můžeme se proto dozvědět, jaké protokoly jsou (byly) používány.

Počet paketů	Počet unikátních hodnot	Entropie
100	1	0
10 000	2	0,099825
1 000 000	2	0,079737

Tab. 7: Výsledky pro pole Typ



Graf 3: Počty IP datagramů verze 4 a 6 [8]

Ve vzorcích se vyskytovaly dvě unikátní hodnoty, a to jedna identifikující protokol IPv4, a druhá pro protokol IPv6 (ve značně menší míře). Z těchto výsledků můžeme říci, že IP je opravdu nejpoužívanější protokol Síťové vrstvy v celém Internetu, ale IPv6 se začíná rozšiřovat do běžné praxe. Mohlo by se automaticky předpokládat, že každý Ethernetový rámec zapouzdřuje IPv4 nebo IPv6 paket a pouze určit zapouzdřenou verzi. Existuje však mnoho jiných protokolů Síťové vrstvy, buď pro specifické účely či méně využívané, které by tímto nebylo možné zapouzdřit do Ethernetového rámce. To, že jsme v měření na žádné nenarazili, neznamená, že nejsou používány.

5.2 IPv4

Verze a Délka hlavičky (4 + 4 bity)

Jak jsme již uvedli v předpokladech, je velmi pravděpodobné (a v případě pole *Verze* jisté), že tato pole přenáší stále stejné údaje. To jsme i zjistili při měření jejich entropie.

Počet paketů	Počet unikátních hodnot	Entropie
100	1	0
10 000	1	0
1 000 000	1	0

Tab. 8: Výsledky pro pole Verze a Délka hlavičky

Z uvedených výsledků můžeme vyvodit závěr, že pole *Verze a Délka hlavičky* neplní v podstatě žádnou funkci. Pro pole *Verze* to navíc vyplývá to také z faktu, že to není proměnné pole. Rozpoznání verze protokolu je na Linkové vrstvě zajištěno protokolem *Ethernet*, který určuje protokol vyšší vrstvy (*Typ*). Jiná situace nastává u *Délky hlavičky*. Nezjistili jsme žádné praktické využívání *Voleb*, což neznamená, že se již nepoužívají. Nicméně z našich výsledků usuzujeme, že toto pole je zbytečné přenášet, protože již neplní žádnou funkci.

Typ služby

Typ služby může sloužit k rozlišení typů provozu za účelem jeho upřednostňování před jiným provozem či speciálnímu zpracování.

Počet paketů	Počet unikátních hodnot	Entropie
100	2	0,160990
10 000	8	0,231085
1 000 000	14	0,216449

Tab. 9: Výsledky pro pole Typ služby

Z výsledků je zřejmé, že se Typ služby prakticky nepoužívá. Je možné, že mechanismus není příliš vhodný pro dnešní praktické aplikace.

Celková délka

Toto pole obsahuje délku celého IP datagramu včetně hlavičky a dat. Pravděpodobně bude obsahovat různé hodnoty podle různých přenášených dat.

Počet paketů	Počet unikátních hodnot	Entropie
100	22	2,962971
10 000	443	4,484552
1 000 000	1 469	4,442974

Tab. 10: Výsledky pro pole Celková délka

Lze pozorovat, že počet unikátních hodnot roste rychleji než entropie pole. Znamená to, že zatímco počet unikátních hodnot roste, některé z nich se objevují častěji než jiné. To bude pravděpodobně proto, že některá data mohou být stejná pro různé komunikace nebo některé protokoly mají neměnné formáty.

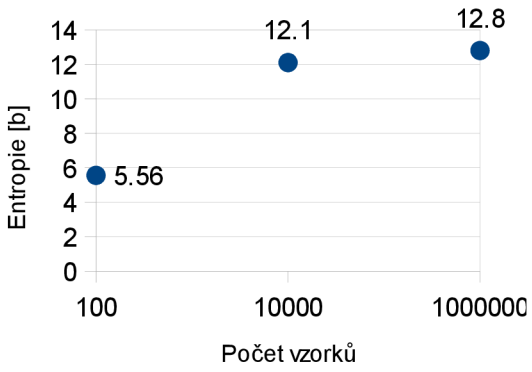
Identifikace (16 bitů)

Pole identifikace má zajistit jednoznačnou identifikaci IP fragmentů v síti. To znamená, že toto číslo se bude jistým způsobem náhodně generovat, aby pravděpodobnost kolize byla minimální. Lze tedy předpokládat, že entropie může dosahovat hodnot bitové velikosti pole.

Počet paketů	Počet unikátních hodnot	Entropie
100	92	6,378630
10 000	8 230	12,104335
1 000 000	65 536	13,786161

Tab. 11: Výsledky pro pole Identifikace

Jak lze vidět z měření, počet unikátních hodnot vzhledem k počtu paketů je poměrně vysoký. Se vzrůstajícím počtem vzorků se také přibližujeme bitové velikosti pole. Této hodnoty však entropie nedosáhne, jelikož není možné zabránit duplicitě hodnot v poli identifikace. To proto, že hodnota tohoto pole je generována různými zařízeními, které nemají informace o již použitých hodnotách. 16 bitů informace dokáže držet 2^{16} = 65536 hodnot. 1000000 datagramů byl dostačující na to, aby se vyskytly všechny možné hodnoty v tomto poli. Entropie je však nižší než 16 bitů, protože počty unikátních hodnot byly nerovnoměrně rozloženy a tím se snížila jejich náhodnost.



Graf 4: Rostoucí entropie s počtem vzorků

Graf 4 zobrazuje závislost entropie pole Identifikace na počtu změřených vzorků. Jak je vidět, hodnota entropie s velkým počtem vzorků stoupá rychleji než na začátku měření. Znamená to, že v množině dosavadní hodnot, resp. v poměrech mezi počty hodnot, již nedochází k výrazným změnám.

Ofset fragmentu (13 bitů)

Hodnoty *Ofsetu fragmentu* budou závislé také na parametrech linky, na které provádíme odchytávání paketů. Linka s menší dovolenou velikostí rámců bude vyžadovat více fragmentace IP paketů než linka s vyšší hranicí.

Počet paketů	Počet unikátních hodnot	Entropie
100	0	0
10 000	2	0,000982
1 000 000	3	0,000192

Tab. 12: Výsledky pro pole Ofset fragmentu

V tabulce je vidět, že Ofset fragmentu má velmi málo unikátních hodnot. Největší zastoupení (více než 98%) měla hodnota 0, což znamená žádná fragmentace nebo také první fragment. Protože ale ostatních hodnot bylo velmi málo, znamená to, že naprostá většina provozu sestávala z nefragmentovaných IP datagramů. Můžeme tvrdit, že entropie tohoto pole už se zvyšujícím se počtem datových jednotek nebude zvyšovat, ale spíš snižovat. To zejména proto, že počet nefragmentovaných datagramů narůstá mnohem rychleji, než počet fragmentovaných a tím se snižuje informační hodnota. Je pravděpodobné, že v dnešní době rychlých spojů a vyspělejších přenosových technologií už je fragmentace k vidění spíše zřídka. Je to ale nutný mechanismus k zachování kompatibility s linkami s různými MTU, které by jinak byly zdrojem ztrát datagramů, protože nepodporují větší datové jednotky.

TTL (8 bitů)

Počet paketů	Počet unikátních hodnot	Entropie
100	17	2,487440
10 000	86	4,337010
1 000 000	165	4,390774

Tab. 13: Výsledky pro pole Time to Live

Při měření se v tomto poli objevovaly různé hodnoty *TTL*. Náš předpoklad, že by mohlo být možné zjistit některé často využívané hodnoty pro tento údaj se nesplnil. Důvodem je vlastní funkce pole, protože je dekrementováno o 1 při každém průchodu směrovačem. Tím pádem hodnota naměřená v jistém bodě bude záviset na počáteční hodnotě a počtu směrovačů, kterým daný paket prošel. Dále relativně vysoká entropie znamená, že pole je hojně využíváno (což je zřejmé už z jeho funkce), proto můžeme říct, že velikost pole je oprávněná a dobře využita.

Počet paketů	Počet unikátních hodnot	Entropie
100	26	4,021039
10 000	90	4,574122
1 000 000	250	4,614936

Tab. 14: Výsledky pro pole Time to Live (lokace 2)

Z nějakého důvodu je počet unikátních hodnot a entropie znatelně vyšší než v předchozích výsledcích. Je možné, že předchozí data byla odchytávána na lince, kde bylo více spojů a pakety byly směrovány tak, aby se rozložila zátěž. Bude také záležet na umístění bodu odchytu dat vzhledem k topologii sítě. Jeden z bodů mohl přenášet více provozu mezi různými zařízeními než druhý. Např. rychlá linka u ISP oproti jedné lince účastníka.

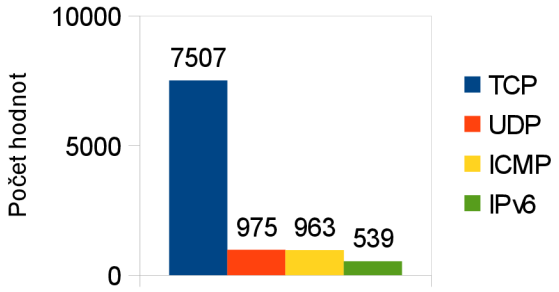
Protokol (16 bitů)

U pole, které určuje zapouzdřený protokol, se dá očekávat rozmanitost hodnot. Je ale pravděpodobné, že některá hodnota (protokol) bude převládat.

Počet paketů	Počet unikátních hodnot	Entropie
100	4	0,792824
10 000	5	1,140152
1 000 000	7	1,149355

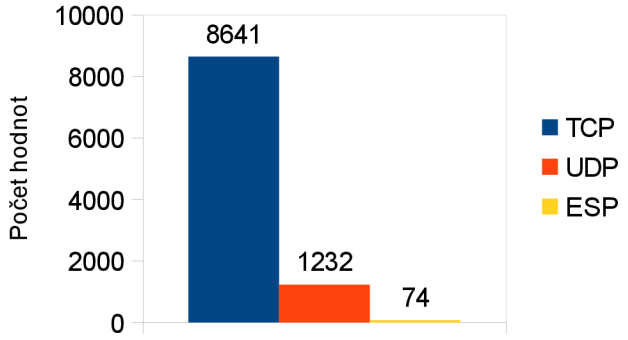
Tab. 15: Výsledky pro pole Protokol

Následující grafy ukazují poměr protokolů Transportní vrstvy přítomných v měřených paketech. Vlastní hodnoty pole jsou již interpretovány na názvy protokolů Transportní vrstvy pro 100 a 10000 hodnot.



Graf 5: Výskyty protokolů v IP datagramu [6]

Zajímavé je, že na 4, resp. 5 unikátních hodnot připadá entropie 0,7 – 1,2 bitu. Je to způsobeno tím, že protokol TCP je ve všech případech zastoupen mnohonásobně více, než ostatní protokoly. Kdyby všechny protokoly měly přibližně stejné procento výskytu, entropie by byla mnohem vyšší (až 2,3 pro 5 unikátních hodnot). Čím více nerovnoměrné jsou pravděpodobnosti výskytů hodnot, tím více se snižuje entropie, protože se snižuje i náhodnost daných hodnot (můžeme předpovědět výskyt jisté hodnoty s vyšší pravděpodobností). Dále se potvrzuje náš předpoklad, že protokol TCP (mnohem méně pak UDP) je nejpoužívanější protokol Transportní vrstvy v dnešním veřejném Internetu. Přítomnost IPv6 zapouzdřeného v IPv4 značí pomalé rozšiřování IPv6 provozu, který je ale zatím nutné přenášet přes IPv4 sítě.



Graf 6: Výskyty protokolů v IP datagramu (jiná lokalita) [6]

Zde můžeme vidět, že podobné výsledky lze zjistit i v jiných lokalitách. Zde se ale místo IPv6 zapouzdřeného v IPv4 objevuje hlavička *ESP* (*Encapsulating Security Payload*), který je součástí technologie *IPsec*. TCP je opět dominantním protokolem síťové vrstvy.

Kontrolní součet hlavičky (16 bitů)

Kontrolní součet hlavičky by měl být pokud možno unikátní pro každou rozdílnou IP hlavičku.

Počet paketů	Počet unikátních hodnot	Entropie [b]
100	99	6,610522
10 000	8 946	12,988141
1 000 000	65 535	15,794387

Tab. 16: Výsledky pro pole Kontrolní součet hlavičky

Graf 7: Entropie Kontrolního součtu

Toto pole má vysokou entropii již díky své funkci - detekci chyb. Lze zde sledovat typický efekt entropie a potažmo celé statistiky, že čím více vzorků použijeme, tím více se výsledný údaj blíží jisté hodnotě, která v případě entropie přesně vyjadřuje informační hodnotu daného pole. Za 1 milion hodnot se objevily až na jednu všechny možné hodnoty v poli (2^{16} = 65 536). Entropie je téměř rovna velikosti pole. Přestože se hodnoty zákonitě musely opakovat, musely být rovnoměrně rozloženy. Entropie by byla znatelně nižší, kdyby některých hodnot bylo více než jiných.

Zdrojová a cílová adresa

Protože IP datagram potřebuje vědět, kam má dorazit, potřebuje znát adresu. IP sítě často propojují různé další sítě, proto je pravděpodobné, že i adres bude mnoho.

Počet paketů	Počet unikátních hodnot	Entropie [b]
100	25	4,175660
10 000	1 010	6,642623
1 000 000	35 672	7,470968

Tab. 17: Výsledky pro pole Zdrojová a Cílová adresa

Unikátních adres v tomto poli není mnoho, ale jsou náhodně rozprostřeny, proto entropie roste. Nemůžeme očekávat, že narazíme na velký počet unikátních IP adres, protože většina světa komunikuje na jiných linkách.

5.3 IPv6

IPv6 byl navržen jako nástupce IPv4 a proto očekáváme, že bude mít lepší parametry než IPv4. Vzhledem k tomu, že prozatím není příliš rozšířen, nemáme k dispozici mnoho dat pro měření.

Verze (4 bity)

Počet paketů	Počet unikátních hodnot	Entropie [b]
436744	1	0

Tab. 18: Výsledky pro pole Verze

Výsledek měření odpovídá předpokladům v kapitole 3.4. Entropie je nulová, protože ve všech paketech byla pouze jedna unikátní hodnota, a to 6.

Třída provozu (8 bitů)

Počet paketů	Počet unikátních hodnot	Entropie [b]
436744	3	0,015680

Tab. 19: Výsledky pro pole Třída provozu

Počet unikátních hodnot společně s entropií napovídá, že většina paketů obsahuje jednu výchozí hodnotu a ostatní jsou v menšině. Určování priority provozu se tedy příliš neuplatňuje.

Značka toku (20 bitů)

Počet paketů	Počet unikátních hodnot	Entropie [b]
436744	4014	0,863407

Tab. 20: Výsledky pro pole Značka toku

Ve výsledcích dominovala hodnota 0, což znamená neoznačený provoz. Zbylé hodnoty byly zastoupeny v desítkách, maximálně stovkách. Může to znamenat, že se tento mechanismus zatím nerozšířil. Nevylučuje to však, že se tak nestane s větším rozšířením protokolu jako takového.

Délka dat (16 bitů)

Počet paketů	Počet unikátních hodnot	Entropie [b]
436744	1194	4,595455

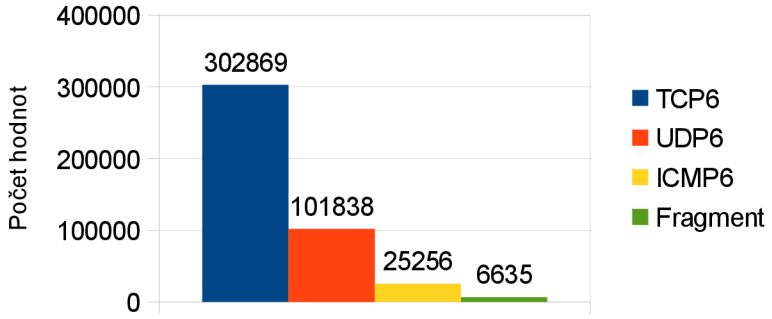
Tab. 21: Výsledky pro pole Délka dat

K výsledkům lze pouze dodat, že se přenáší rozmanitá data. Jistá část provozu však bude často podobná (např. se budou přenášet stejná data), proto entropie tohoto pole není velká. Je možné, že by byla větší při použití více vzorků.

Další hlavička (8 bitů)

Počet paketů	Počet unikátních hodnot	Entropie [b]
436744	7	1,265047

Tab. 22: Výsledky pro pole Další hlavička



Graf 8: Výskyty v poli Další hlavička [6]

Lze sledovat podobné využití protokolů jako v IP verze 4, tzn. protokol TCP je nejpoužívanějším protokolem, za ním UDP. Můžeme také sledovat jistý počet hlaviček fragmentů. To znamená, že fragmentace se stále používá, ač málo.

Limit skoků (8 bitů)

Počet paketů	Počet unikátních hodnot	Entropie [b]
436 744	64	3,506541

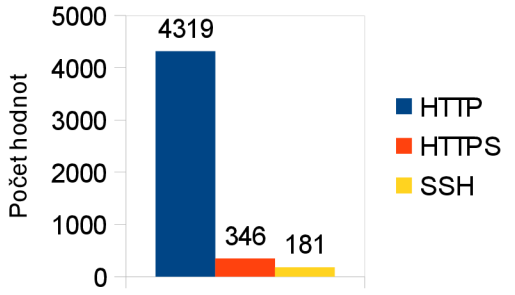
Tab. 23: Výsledky pro pole Limit skoků

5.4 TCP

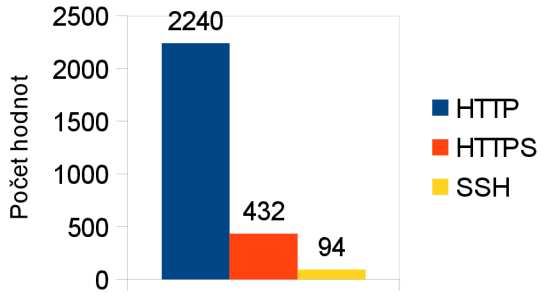
Zdrojový a cílový port (16 bitů)

Počet paketů	Počet unikátních hodnot	Entropie
100	21	2,225458
10 000	1 064	5,233166
1 000 000	17 773	6,110984

Tab. 24: Výsledky pro pole Zdrojový a Cílový port



Graf 9: Porty v poli Zdrojový port [7]



Graf 10: Porty v poli Cílový port [7]

Pro obě pole platí podobné poměry používaných protokolů Aplikační vrstvy. Jak jsme předpokládali, velká část *TCP* provozu se skládá právě z *HTTP* a jeho zabezpečené verze *HTTPS*. Dále ze známých portů se výrazněji podílela na provozu služba SSH, což je zabezpečená alternativa Telnetu. Zbylé porty s většími počty byly porty nad číslo 1024, což bude tvořit komunikaci ke klientovi (od klienta), protože se porty na straně klienta často volí náhodně právě z hodnot vyšších než 1024.

Sekvenční číslo (32 bitů)

Počet paketů	Počet unikátních hodnot	Entropie
100	87	6,169340
10 000	7 155	11,743136
1 000 000	659 313	17,129509

Tab. 25: Výsledky pro pole Sekvenční číslo

Na entropii je vidět, že pole obsahuje mnoho unikátních hodnot. Samozřejmě jednu hodnotu nalezneme vícekrát díky účelu a způsobu jejího vytváření.

Potvrzovací číslo (32 bitů)

Potvrzovací číslo by mělo mít podobné vlastnosti jako *Sekvenční číslo*, protože s ním přímo souvisí.

Počet paketů	Počet unikátních hodnot	Entropie
100	35	3,844868
10 000	3 386	9,280495
1 000 000	283 454	13,109180

Tab. 26: Výsledky pro pole Potvrzovací číslo

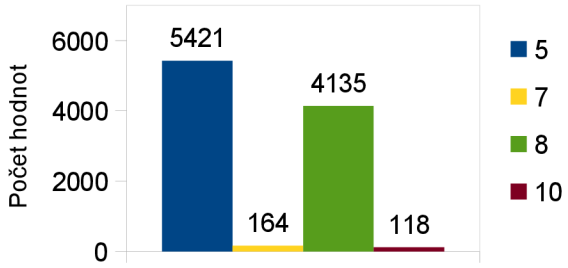
Zajímavé je, že počet unikátních hodnot je menší než poloviční oproti *Sekvenčnímu číslu*. Pravděpodobně proto, že v TCP komunikaci se toto číslo může odeslat vícekrát v jedné relaci.

Posun dat (4 bity)

Posun dat určuje velikost hlavičky TCP segmentu. Lze tedy očekávat, že nebude velká rozmanitost hodnot, protože není tolik možností rozšiřování této hlavičky.

Počet paketů	Počet unikátních hodnot	Entropie
100	3	0,586110
10 000	9	1,208819
1 000 000	11	1,240485

Tab. 27: Výsledky pro pole Posun dat



Graf 11: Hodnoty Posunu dat

Hodnota posunu bez přídavných voleb na konci hlavičky je 5. Proto lze usoudit, že existují volby (ty rozšiřují TCP hlavičku), které jsou dnes často používány.

Okno
f

Pole *Okno* používají komunikující strany k domluvě vyrovnávací paměti pro příjem a sledování potvrzených paketů.

Počet paketů	Počet unikátních hodnot	Entropie
100	19	2,711211
10 000	735	6,049296
1 000 000	14 347	6,791751

Tab. 28: Výsledky pro pole Posun dat

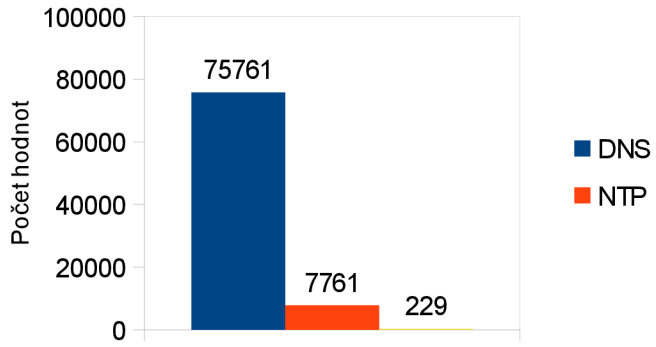
Zdá se, že hodnota entropie dále neroste a pohybuje se pod 7 bity. Pravděpodobně existují běžné hodnoty, které se používají častěji. Exotičtější hodnoty pole jsou zastoupeny v menšině, proto je entropie menší.

5.5 UDP

Zdrojový a cílový port (4 bity)

Počet paketů	Počet unikátních hodnot	Entropie
100	25	2,980993
10 000	962	4,396097
1 000 000	37 191	5,058967

Tab. 29: Výsledky pro pole Zdrojový a Cílový port



Graf 12: Protokoly aplikační vrstvy v UDP datagramu [7]

Jak jsme předpokládali, překlad doménových adres hraje důležitou roli v síťovém provozu. To lze také vyvodit z výsledků pro protokol TCP. Služba HTTP funguje ve většině případů ve spolupráci právě se službou DNS. V mnohem menší, ale podstatné míře se objevuje i protokol NTP (*Network Time Protocol*) sloužící k synchronizaci času. Lze proto vyvodit závěr, že se dnes zařízení běžně synchronizují svůj čas po síti.

Délka

Entropie tohoto pole bude hlavně záviset na rozmanitosti přenášeného provozu a protokolech vyšší vrstvy.

Počet paketů	Počet unikátních hodnot	Entropie
100	39	4,593305
10 000	566	6,110897
1 000 000	1 461	6,464661

Tab. 30: Výsledky pro pole Délka

Ve výsledcích entropie zůstává při větších počtech vzorků na jisté hodnotě, můžeme tedy říct, že dále příliš neporoste, protože se sice objevují nové hodnoty, ale jsou v malém počtu, proto entropii jen málo ovlivňují.

Kontrolní součet

Zde očekáváme podobné vlastnosti jako u předchozích kontrolních součtů v protokolech.

Počet paketů	Počet unikátních hodnot	Entropie
100	9	2,682020
10 000	875	8,979241
1 000 000	45 984	14,828583

Tab. 31: Výsledky pro pole Kontrolní součet

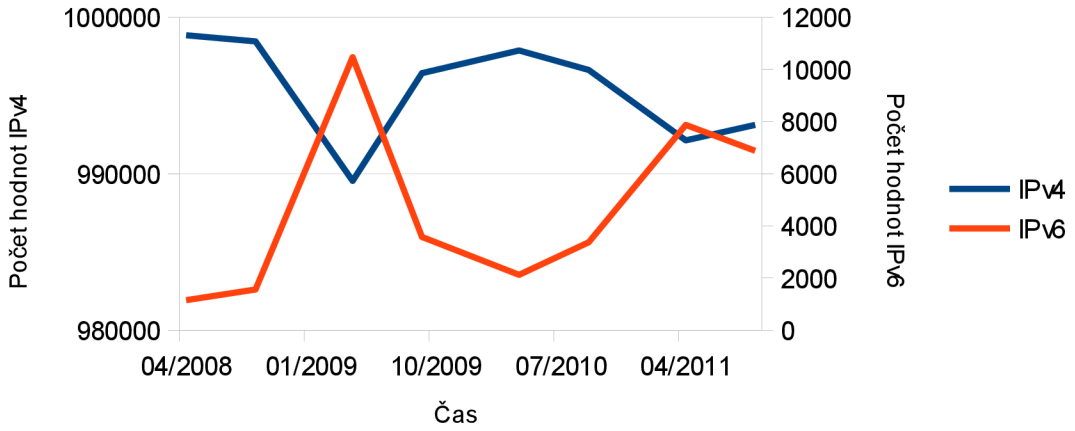
Předpoklad se opět potvrdil, a to že kontrolní součty obsahují různé hodnoty, což je velmi pravděpodobně způsobeno různým provozem. Můžeme tedy říct, že v UDP datagramech se data často neopakují.

6 Trendy v komunikacích

Pomocí změřené entropie (podpořenou výčtem vyskytujících se hodnot) se můžeme pokusit zjistit změny složení Internetového provozu a určit, čím je to způsobeno. Sítě se v průběhu času mění. Zřejmě nasazení nové technologie bude mít dopad na celkové složení a objem přenášených dat a tím pádem i na vlastnosti dat, tedy i entropii. Data budeme měřit ze dvou měsíců od roku 2007 do roku 2011, popř. prvního čtvrtletí 2012. Výsledky budou v rámci měsíců zprůměrovány. Předpokládáme, že využití IP verze 6 nebude příliš stoupat. Komunikační infrastruktury IPv6 podporují, ale chybí využití ze strany uživatelů linky. Očekáváme, že se časem zvětšuje objem provozu protokolu HTTP (přenášen pomocí TCP), protože se neustále zvyšuje počet webových stránek a také uživatelů disponujících Internetovým připojením.

6.1 Ethernet

U protokolu Ethernet neočekáváme v podstatě žádné velké rozdíly, protože neobsahuje informace, jejichž využití by měnilo průběhem času. Můžeme ale zkusit sledovat pole Typ, které nám řekne, jaké protokoly se používají v Síťové vrstvě. Vzhledem k nasazování IP verze 6, mohou nám tyto údaje poskytnout náhled do vývoje této situace.

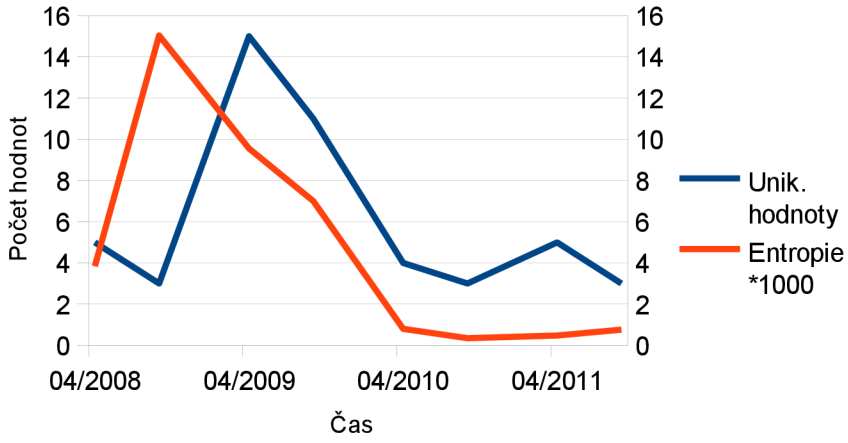


Graf 13: Výskyty IP datagramů verze 4 a 6 v čase [8]

Zřejmě, pokud se využívá buď IPv4 nebo IPv6, budou se grafy výskytu těchto protokolů v Ethernetu doplňovat.

6.2 IPv4

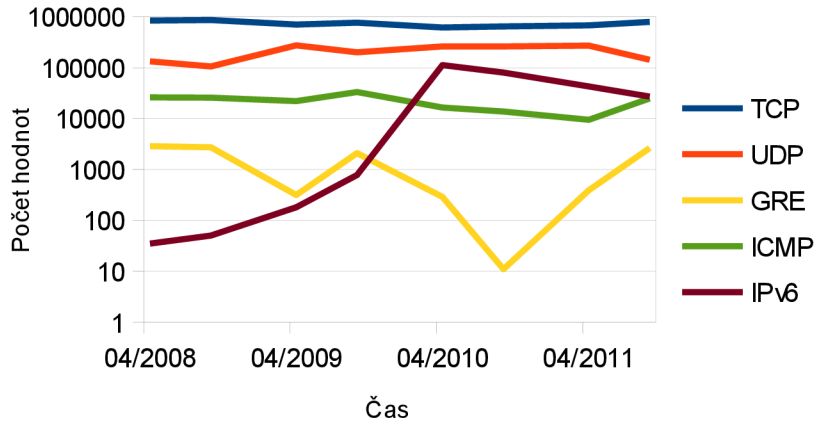
IPv4 obsahuje několik údajů, které slouží specifickému účelu. Je možné, že objevíme nějakou závislost. K zajímavým poznatkům můžeme dojít zejména při zkoumání polí obstarávajících fragmentaci a rozlišení služeb (DSCP). Dále se podíváme na užití protokolů vyšší vrstvy.



Graf 14: Unikátní hodnoty a entropie pole Posun fragmentu v čase

Pro názornost a snazší srovnání násobíme entropii konstantou 1000, protože se pohybuje ve velmi malých číslech. Z grafu lze pozorovat značný pokles entropie *Posunu fragmentu* v průběhu posledních 5 let. Lze usoudit, že v době, kdy k poklesu dochází, přestávalo být nutné IP datagramy fragmentovat. Pravděpodobně díky vylepšení přenosových linek vedoucích do této linky, popř. změně parametrů této linky.

Dále se podíváme na protokoly Transportní vrstvy zapouzdřené v IP datagramech

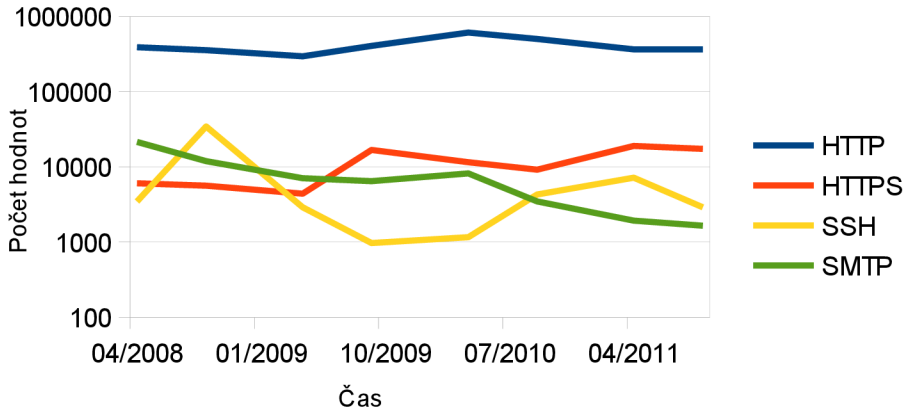


Graf 15: Výskyty protokolů v IP datagramu v čase [6]

Z grafu vychází, že protokoly TCP a UDP se využívají relativně stejně jako v předchozích letech. Je zde ale patrný nárůst využití IPv6 zapouzdřeného v IPv4 jako mechanismu pro přenášení IPv6 datagramů přes IPv4 sítě.

6.3 TCP

V protokolu TCP můžeme sledovat využití služeb Aplikační vrstvy. Ty jsou indikovány poli *Zdrojový port* a *Cílový port*.

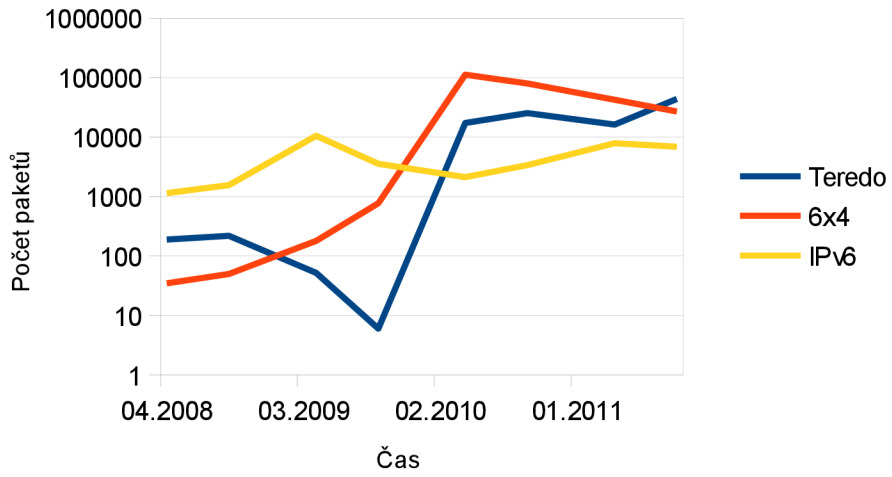


Graf 16: Protokoly přenášené v TCP [7]

Vidíme, ze provoz zabezpečeného HTTP se pomalu zvyšuje, nejspíše důsledkem jeho častějšího nasazování v nových i stávajících webech. Počet SMTP dat se naopak snižuje, pravděpodobně z důvodu zastarávání služby E-mail jako moderního komunikačního prostředku a jeho nahrazování sociálními službami, které komunikaci značně usnadňují. Dále mnoho uživatelů již nepoužívá klasické E-mailové klienty ve prospěch webových rozhraní, kde jsou odeslané E-maily v podobě HTTP(S) přenosů.

6.4 Rozšiřování IPv6

Protokol IP verze 6 se začal navrhovat, když bylo zřejmé, že adresový prostor IP verze 4 rychle ubýval a mohlo by dojít k jeho vyčerpání. Proto byla velikost adres zvětšena, aby k tomuto nemohlo dojít v blízké budoucnosti. Nicméně nový protokol nebylo možné nasadit ihned. Podpora pro IPv6 se přidávala (a stále přidává) postupně, aby nebylo nutně vyměnit všechno vybavení najednou. Byly vytvořeny protokoly a principy, jak umožnit přenášení IPv6 přes IPv4 a jak překládat adresy z těchto adresních prostorů. Už jsme se setkali s IPv6 zapouzdřeným v IPv4. Tento princip se používá u mechanismů jako např. *6in4* nebo *6to4*. Používá se však i zapouzdření v protokolu Transportní vrstvy, čehož využívá např. *Teredo*.



Graf 17: Přechodové mechanismy IPv6 a IPv6 [6][7][8]

Graf 17 ukazuje závislost mezi mechanismy dovolujícími přenos IPv6 paketů přes IPv4 sítě a rozšíření IPv6 zapouzdřeného v Ethernetu. Je zde vidět rostoucí užití těchto přechodových mechanismů. To naznačuje připravenost některých sítí (uživatelů) IPv6 používat, nicméně v tomto brání některé části infrastruktury, které IPv6 zatím nepodporují. Použití samotného IPv6 zatím relativně stagnuje.

6.5 Shrnutí poznatků

Nyní si srovnáme definované velikosti polí a naměřenou entropii. Sečtením entropie zjistíme, na kolik by se maximálně teoreticky dala zredukovat hlavička. To ale neplatí pro každé pole a ukážeme si proč.

Ethernet

Pole	Počet bitů [b]	Entropie [b]
Zdrojová MAC adresa	48	0,95
Cílová MAC adresa	48	0,95
Typ	16	0,08
Celkem	112	1,98

Dle měření by se celá Ethernetová hlavička dala zredukovat na pouhé dva bity. Entropie je v tomto mírně zavádějící, protože nemůžeme přenášet např. půl bitu. Takže by to musely být tři bity. MAC adresy by sice mohly být jeden bit, způsobilo by to ale potíže při používání sítě, viz detailněji

v kapitola 5.1. *Typ* by mohl být jeden bit (IPv4/IPv6), ale tím by se omezila použitelnost Ethernetu v místech, kde se používá jiný protokol na Linkové vrstvě.

IPv4

Pole	Počet bitů [b]	Entropie [b]
Verze	4	0
Délka hlavičky	4	0
Typ služby	8	0,22
Celková délka	16	4,44
Identifikace	16	13,79
Posun fragmentu	13	0
TTL	8	4,39
Protokol	8	1,15
Kontrolní součet	16	15,79
Zdrojová adresa	32	7,47
Cílová adresa	32	7,47
Celkem	157	54,72

V protokolu IPv4 by docela jistě bylo možné některá pole zredukovat. Podle měření nemají pole *Verze* a *Délka hlavičky* žádný význam. Některá pole jako např. *TTL* také nemůžeme zmenšit, protože fungují jako čítač a jejich rozsah je důležitý zachovat. IP adresy je třeba zachovat také, protože se snažíme o adresovatelnost zařízení v globálním Internetu. Ve skutečnosti nastává případ, že pole (a tudíž počet hodnot) je příliš malý.

IPv6

Pole	Počet bitů [b]	Entropie [b]
Verze	4	0
Typ provozu	8	0,02
Značka toku	20	0,86
Délka dat	16	4,60
Další hlavička	8	1,27
Limit skoků	8	3,51
Celkem	64	10,26

V IPv6 opět vystupuje pole *Verze*. Vzhledem k tomu, že už v IPv4 bylo toto pole konstantní, musí mít v protokolech IP jistý význam. Je možné, že jde o kompatibilitu s protokoly nižší vrstvy, které nemají informaci o tom, jaký protokol přenáší. Pokud by toto byl ten případ, tak by možná bylo

výhodnější tyto málo používané protokoly upravit tak, aby nesly informaci o zapouzdřeném protokolu.

TCP

Pole	Počet bitů [b]	Entropie [b]
Zdrojový port	16	6,11
Cílový port	16	6,11
Sekvenční číslo	32	17,13
Potvrzovací číslo	32	13,11
Posun dat	4	1,24
Okno	16	6,79
Kontrolní součet	16	15,94
Celkem	132	71,66

Z výsledků vyplývá, že TCP je poměrné dobře navržen z hlediska využití jeho polí. Mnoho z nich má relativně vysokou entropii v porovnání s jejich specifikační délkou. Celkové výsledky jsou velmi dobré vzhledem k ostatním protokolům.

UDP

Pole	Počet bitů [b]	Entropie [b]
Zdrojový port	16	5,06
Cílový port	16	5,06
Délka	16	6,47
Kontrolní součet	16	14,83
Celkem	64	31,42

UDP má podobné vlastnosti jako TCP, tj. využití polí je poměrné dobré. Nejvyšší entropii má opět pole Kontrolní součet díky svému účelu a principu.

Shrnuli jsme si měřené protokoly a zhodnotili jejich entropie. Samozřejmě musíme mít na paměti, že entropie je pojem teoretický a je třeba brát jako ukazatel informační hodnoty. Pokud měřením zjistíme jistou hodnotu entropie, neznamená to, že je pole jednoznačně nevyužito. Je třeba přihlédnout i k účelu pole, případně k vývoji jeho využití. Některá pole mohou mít zanedbatelnou entropii a přesto plnit svůj účel.

7 Závěr

V této práci jsme si ukázali, jak měřit entropii. Aplikovali jsme tyto poznatky na pole Internetových protokolů. Lze takto měřit v podstatě jakýkoli protokol, u kterého můžeme definovat data nad nimiž chceme entropii měřit. Takto by bylo možno měřit entropii např. u hlavičky protokolu HTTP, když si správně nadefinujeme, co považujeme za stejné hodnoty, protože syntaxe je složitější a už se neomezuje na „jednoduché“ celočíselné hodnoty.

Popsali jsme si používané protokoly a uvedli výsledky měření. Některé výsledky bylo možné dopředu předvídat, jiné odhalily zajímavé poznatky.

Protokoly časem zastarávají. Vlastnosti prostředí, ve kterém se užívají se často mění, protokoly však často zůstávají bez úprav dlouhou dobu. Tím může být ztížena komunikace, protože např. nepodporuje vlastnost, která je právě aktuální. Občas se také může stát, že se při návrhu protokolu počítá s nějakým mechanismem či principem, který ale časem zastará nebo ztratí na významu. Dále v protokolech mohou existovat rezervované bity pro budoucí účely, které ale nikdy nemusí být využity. Příkladem toho jsou rezervované bity v protokolu IPv4. V důsledku nasazovaní jeho verze 6 je velmi pravděpodobné, že již ani nedojde k jejich využití. Pokud by se skutečnému účelu polí věnovalo při návrhu protokolů více pozornosti, bylo by možné celkově zefektivnit komunikaci. Mnoho dnes používaných protokolů se nezměnilo od své specifikace v dobách, kdy se s masovým rozšířením počítalo spíše okrajově. Bral se ohled na technologie a protokoly, které časem zastaraly a již se nepoužívají. Jelikož ale jakákoli dodatečná změna může být značně nákladná, může být ponechání již bezvýznamných dat v protokolu často schůdnější cestou.

Entropie nám také může ukázat možnost komprese dat. Zrychlování přenosů nemusí vždy znamenat zrychlování přenosové linky, ale také zmenšováním dat přenášených mezi zařízeními. Mimo odstraňování nepotřebných údajů může stát za zvážení i komprese dat polí. Samozřejmě toto zvětší výkonové požadavky na zařízení, faktem ale je, že největší zpoždění nastává na přenosových linkách. Příkladem užití komprese v praxi je protokol *FAST/FIX*. Používá se pro přenosy finančních informací (burzy), kde je vyžadováno co nejmenší zpoždění. Proto se minimalizoval i objem přenášených dat.

Lze chápat, že návrh protokolu, který má být užíván po celém světě je obtížné. Je třeba zvážit mnoho faktorů; v této práci pravděpodobně nemůžeme postihnout všechny z nich. I přesto je důležité, aby se otázce přenášení nedůležitých dat věnoval jistý prostor při návrhu protokolů.

Literatura

[1] Cover, T. M., Thomas, J. A.: *Elements of Information Theory*. 2nd Edition. 2006. 776 s. ISBN 0-471-24195-4

[2] Javvin Technologies Inc. *Network Protocols Handbook*. Second Edition. 2007. 341 s. ISBN 978-0-9740945-2-6

[3] Tanenbaum, A. S.: *Computer Networks*. Fourth Edition. 2003. 891 s. ISBN 0-13-066102-3

[4] Töpfer, P.: *Algoritmy a programovací techniky*. 1. vydání. 1995. 298 s. ISBN 80-85849-83-6

[5] *Tcpdump/Libpcap public repository* [online]. 20.9.2010, <http://www.tcpdump.org/>

[6] Protocol Numbers [online]. 1.11.2011, <http://www.iana.org/assignments/protocol-numbers/>

[7] *Service Name and Transport Protocol Port Number Registry* [online]. 7.5.2012, <http://www.iana.org/assignments/service-names-port-numbers/>

[8] *IEEE Public Ethertype List* [online]. 13.5.2012, <http://standards.ieee.org/develop/regauth/ethertype/eth.txt>

[9] *GCC, the GNU Compiler Collection*. 9.4.2012, <http://gcc.gnu.org/>

[10] *Entropy in thermodynamics and information theory*. 3.5.2012, <http://en.wikipedia.org/wiki/Entropy_in_thermodynamics_and_information_theory>