Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Semestrální práce z předmětu

Počítačové sítě

Zachytávání a analýza IP paketů

Jiří Kučera, A08N0092P

kalwi@students.zcu.cz

23. 6. 2009

Zadání první samostatné úlohy - RPC

Navrhněte program pro zachycování a analýzu paketů IP. Analyzujte zejména volitelné parametry. K ověření funkčnosti využijte vhodný program (např. ping s možností nastavení délky paketu) nebo vlastní program, datagramovou aplikaci s možností nastavovat IP parametry pomocí funkce setsockopt(). Dále realizujte program pro syntézu IP datagramů, dovolující testovat libovolné parametry IP záhlaví a nastavovat libovolnou délku paketů. K zobrazení výsledků využijte vhodného grafického rozhraní (FLTK). Naprogramujte pod operačním systémem Linux v programovacím jazyce C (C++) nebo Java.

Programátorská dokumentace

Aplikace je naprogramována v jazyce Java s použitím knihovny Jpcap, což je nadstavba pro nativní knihovnu PCAP (potažmo WinPcap). Existují dvě odlišné, ale stejnojmenné knihovny Jpcap, a to:

http://sourceforge.net/projects/jpcap

a

http://netresearch.ics.uci.edu/kfujii/jpcap/doc/

První z nich nabízí bohatší API, ovšem nenašel jsem možnost odesílat pakety (pouze zachytávat), navíc poslední verze je z roku 2004. Druhá (poslední verze z 2007) je co do počtu tříd o poznání chudší, ale umožňuje odesílat pakety, proto jsem použil tuto. Nicméně v implementaci jsem nalezl spousty chyb; některé z nich bylo možno obejít, jiné však ne. Nalezené chyby a způsob jejich obejití uvedu níže v dokumentaci.

Protože jak zachytávání, tak odesílání RAW paketů na síťové úrovni je možno jen s pomocí podpůrných knihoven (Jpcap), sloučil jsem analyzátor i syntetizér paketů do jedné aplikace.

Aplikace je rozdělena do dvou vrstev – síťovou a prezentační, každá z vrstev se nachází v samostatném balíku.

Síťová vrstva (balík psi.ip.net)

V této vrstvě se nachází následující třídy:

Captor

Spouští a zastavuje vlákno pro zachytávání paketů. Zachycené pakety posílá prezentační vrstvě.

Sender

Umožňuje odesílat pakety na síť. Protože pakety se vkládají do Ethernetovských rámců, pro odeslání paketu vyžaduje zadat i zdrojovou a cílovou MAC adresu.

IPv4Packet

Reprezentuje zachycený paket. Knihovna Jpcap sice obsahuje třídu `IPPacket`, která slouží ke stejnému účelu, ovšem ta obsahuje chyby v implementaci. Třída `IPv4Packet` tedy slouží jako fasáda zaobalující původní chybu a skrývající chyby.

FakeOptionsGenerator

Generuje falešné volitelné parametry IP paketu pro účely ladění analyzátoru. Další důvod existence této třídy je uveden v sekci s popisem chyb knihovny Jpcap.

Prezentační vrstva (balík psi.ip.gui)

Vrstva obsahuje tyto třídy:

Controller

Aplikační rozhraní mezi síťovou a prezentační vrstvou. Zachytává události z GUI a spouští přidružené akce. Taktéž uchovává zachycené IP pakety ze síťové vrstvy.

Třídy `Controller` a `Captor` jsou propojeny přes návrhový vzor Observer-Observable, což zjednodušuje zasílání paketů ze síťové vrstvy do prezentační, a zároveň je tak zajištěna synchronizace mezi vlákny.

MainWindowFrame

Třída zobrazující hlavní okno aplikace. Definuje všechny ovládací/vizualizační prvky a nastavuje jejich hodnoty podle hodnot v zobrazovaném paketu.

PacketsTableModel

Třída implementující rozhraní `TableModel`. Poskytuje tabulce zobrazující pakety data ve vhodné reprezentaci.

SenderWindowFrame

Třída zobrazující okno pro odesílání paketů. Při odeslání vloží údaje z formuláře do IP paketu a ten nechá odeslat síťovou vrstvou.

Parser

Pomocná třída pro parsování části hlavičky IP paketu s volitelnými parametry a dále datové části. Tyto údaje převádí na čitelný textový řetězec.

Chyby nalezené v knihovně Jpcap a jejich řešení

1) Knihovní třída IPPacket nevrací všechny parametry hlavičky IP paketu.

Jedná se o tyto parametry:

- Internet Header Length
- Type of Service
- Flags
- Header Checksum

Toto jsem obešel vytvořením obalující třídy (`IPv4Packet`), která chybějící hodnoty dopočítá z pole obsahujícího hlavičku v binární podobě.

2) Datová část zachyceného paketu je poškozená.

Obsahuje-li zachycený paket data o délce více než 12 bytů, nejsou tato data vrácena buďto vůbec, nebo jen jejich úsek. To může být způsobeno chybnou implementací knihovny Jpcap, nebo dokonce WinPcap. Tuto chybu nelze obejít žádnými programovacími prostředky.

Pozn.: Odesílané pakety obsahují všechna data nepoškozená.

3) Knihovna Jpcap neumožňuje u odesílaných paketů nastavit všechny parametry.

Jedná se o tyto parametry:

- Version
- Fragment Offset
- Options

Nemožnost nastavení parametru Version je víceméně logická, neboť jiná hodnota než 4 by v IPv4 sítích neměla smysl.

S nemožností nastavení Fragment Offset je nutno se smířit, nicméně tato hodnota není z hlediska analýzy jednoho paketu tak zajímavá, takže tato chyba není nikterak fatální.

Nemožnost nastavit odesílaným paketům volitelné parametry (pole Options) je poměrně hrubá chyba znesnadňující důsledné splnění zadání. Krom toho v síti, ve které jsem aplikaci ladil, nepřicházely žádné pakety obsahující volitelné parametry. Proto jsem pro účely ladění implementoval možnost vložit do zachycených paketů falešné volitelné parametry, a to ještě před analýzou paketu, takže analyzátoru se pak takový paket jeví, jako by volitelné parametry obsahoval. Pochopitelně je pak také přepočítána délka hlavičky i celého paketu. Obsahuje-li zachycený paket volitelné parametry, pak jsou tyto ponechány tak, jak jsou. Falešné volitelné parametry generuje výše zmíněná třída `FakeOptionsGenerator`.

Uživatelská dokumentace

Ke spuštění aplikace je třeba mít nainstalovány knihovny Pcap (WinPcap v případě OS MS Windows) a Jpcap. V případě 64b systému MS Windows není možno použít 64b Javu, ale jen 32b.

Aplikace se spustí příkazem

```
java -jar IPAnalyzer.jar
```

Po spuštění se zobrazí okno analyzátoru (Obrázek 1: Okno analyzátoruObrázek 1). V horní části okna se nacházejí ovládací prvky pro zachytávání:

Start – spustí zachytávání

Stop – zastaví zachytávání

Clear – zahodí všechny zachycené pakety

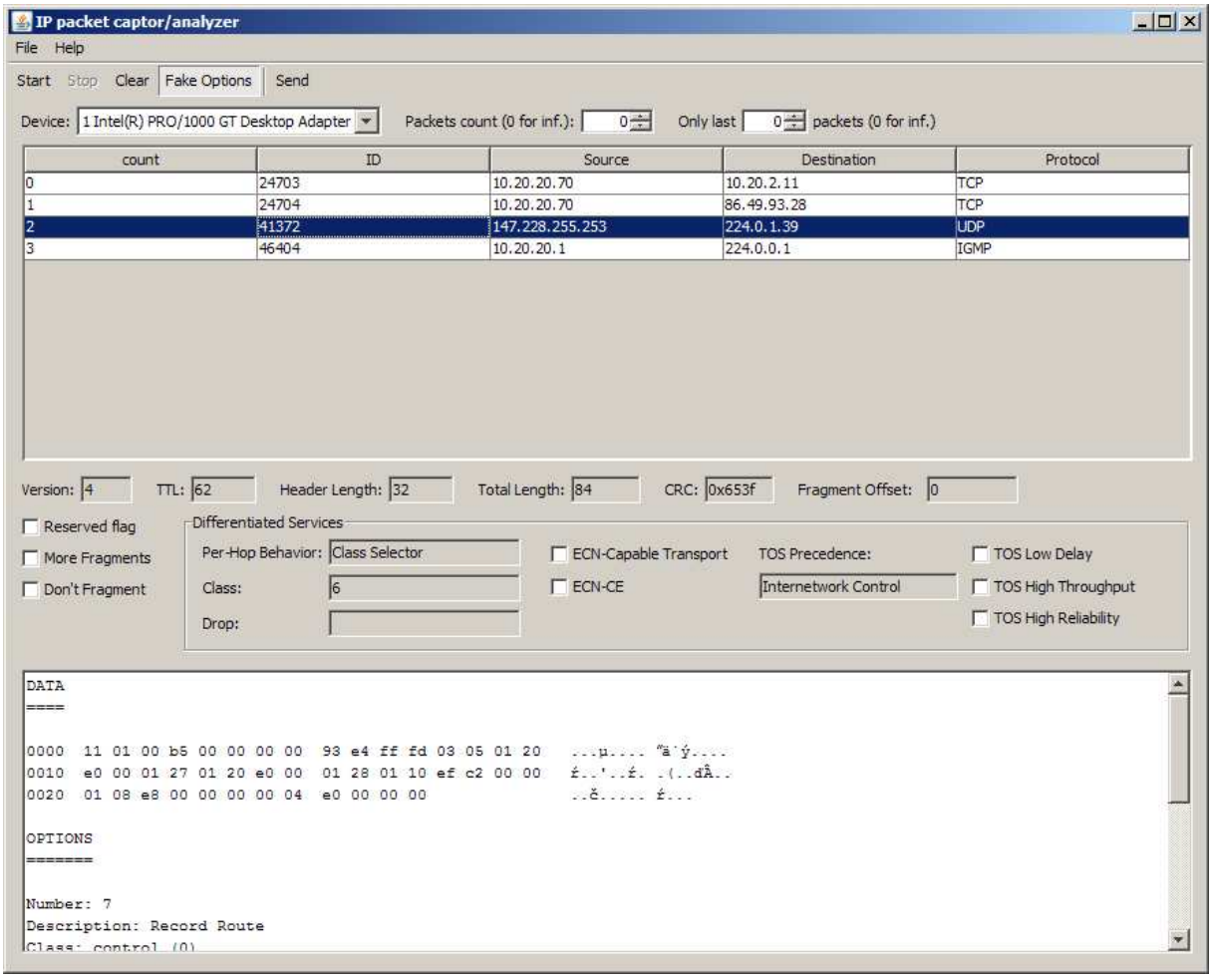
Fake Options – zapne vkládání falešných volitelných parametrů do zachycených paketů

Device – umožňuje vybrat síťové rozhraní, na kterém se bude zachytávat.

Packets count – limituje počet zachycených paketů, po dosažení tohoto limitu se zachytávání vypne; nastavením nulové hodnoty se tato volba deaktivuje

Only last … packets – v tabulce se zobrazí pouze stanovené množství paketů, starší pakety budou postupně mizet s tím, jak se zachytávají pakety nové, nastavením nulové hodnoty se tato volba deaktivuje

Pod ovládacími prvky se nachází tabulka se zachycenými pakety. Kliknutím na paket se zobrazí detailní informace o paketu ve formuláři v dolní části okna.



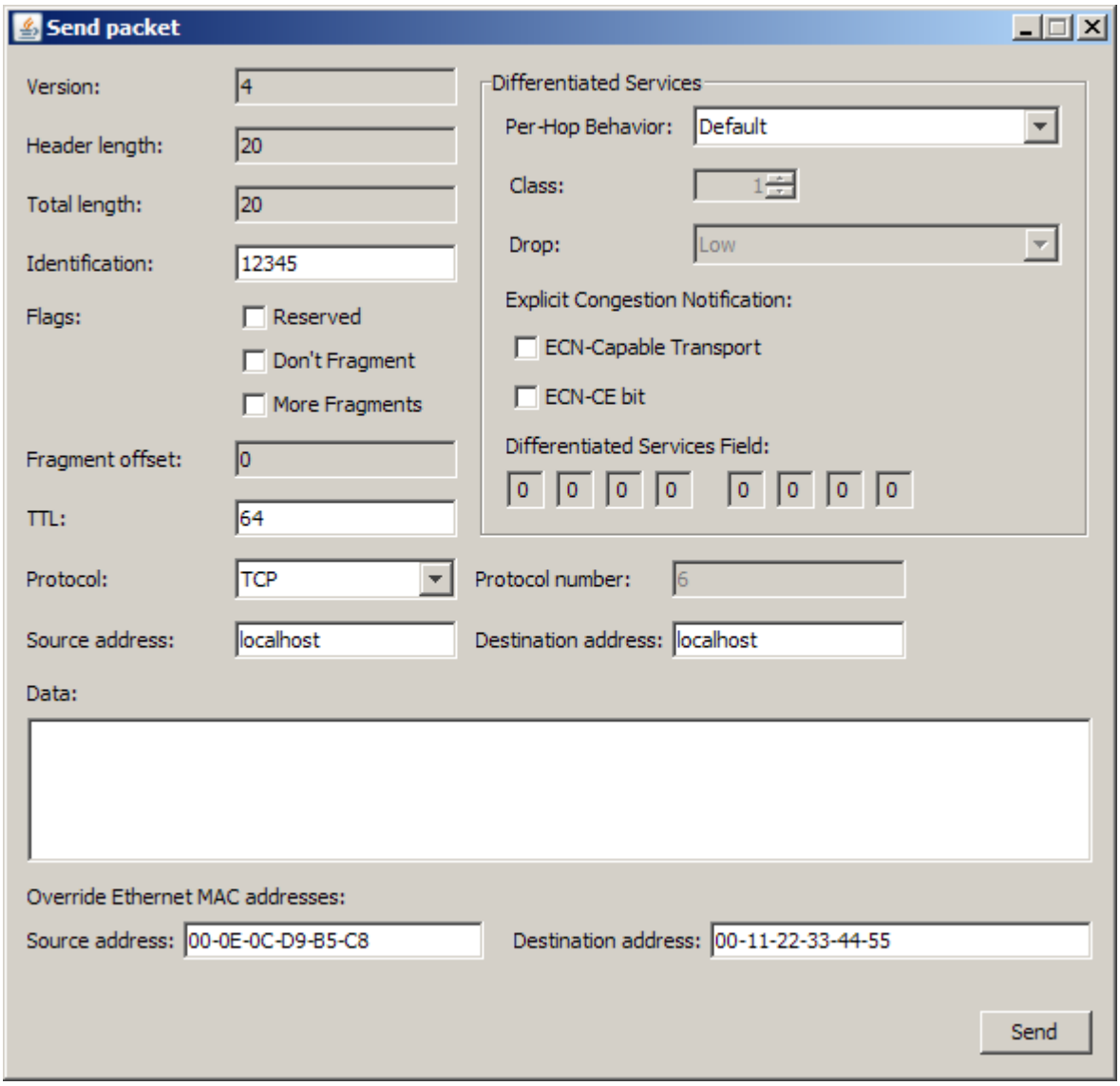
Obrázek 1: Okno analyzátoru

Okno pro odesílání paketů (Obrázek 2) se zobrazí po stisknutí tlačítka **Send**.

V okně se nachází formulář se všemi parametry protokolu IP. Ty položky, které není možné nastavit (buď z důvodu chybějící podpory v Jpcap, nebo nesmyslnosti volby), jsou znepřístupněny.

Délku paketu je možno ovlivňovat naplněním textové oblasti ve spodní části obrazovky daty. Rovnou se přepočítá hodnota v textovém poli zobrazujícím délku paketu.

Oktet TOS se nastavuje podle RFC 2474 jako Differentiated Services, protože původní TOS se již dnes nepoužívá (ovšem v analyzátoru se zobrazí i hodnoty reprezentované jako TOS).



Obrázek 2: Okno pro odesílání paketů

Tlačítkem **Send** v dolní části okna se paket odešle na stejné síťové rozhraní, které je nastavené v analyzátoru.

Závěr

Aplikaci jsem vyvíjel pod operačním systémem MS Windows Server 2008 x64 SP2 s knihovnou WinPcap 4.1 beta5 a její nadstavbou Jpcap 0.7 a JDK 1.6.0_13-b03 x86.

Jako nejvhodnější způsob ladění aplikace (flexibilnější než výstup do souboru) mi přišlo souběžné zachytávání paketů programem Wireshark a následné porovnávání hodnot v zachycených paketech. Tento způsob ladění jsem použil i pro syntetickou část – odesílané pakety jsem kontroloval jak v analyzéru, tak ve zmiňovaném Wiresharku.

V použité knihovně Jpcap jsem nalezl několik chyb. Většinu z nich se povedlo nějakým způsobem obejít. Fatální chybou jsou poškozená data v přijímaných paketech. Programem Wireshark jsem ověřil, že naštěstí alespoň data odesílaná jsou v pořádku.

Doba trvání vývoje aplikace včetně ladění a psaní dokumentace trvala zhruba týden až deset dní (po 12 – 16 hod/den).

Použité materiály

RFC 790 – Assigned Numbers

RFC 791 – Internet Protocol

RFC 2474 – Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers

RFC 3168 – The Addition of Explicit Congestion Notification (ECN) to IP

http://en.wikipedia.org/wiki/IPv4

http://en.wikipedia.org/wiki/Differentiated_services