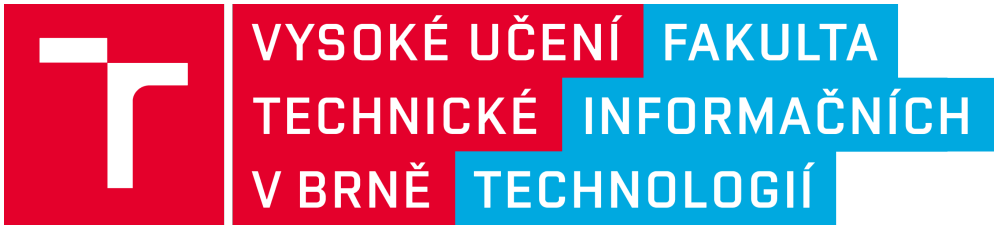

Počítačové komunikace a sítě

Varianta ZETA: Sniffer paketů

Tereza Burianová (xburia28)

25. dubna 2021

Obsah

1 Protokoly

Protokoly jsou jistá pravidla pro komunikaci zařízení. Mezi takové vlastnosti patří například adresát zprávy nebo porty pro odlišení své komunikace. [**?**]

1.1 TCP

TCP, neboli *Transmission Control Protocol*, je protokol umožňující oboustranné odesílání a přijímání dat. Kromě dat obsahuje také hlavičku, která je reprezentována datovou strukturou *tcphdr*. [4]

1.2 UDP

Protokol UDP je od TCP rozdílný hlavně v doručování tzv. ”bez záruky”. Skládá se totiž z několika nezávislých zpráv. Jeho hlavička je v kódu reprezentována strukturou *udphdr*. [5]

1.3 ICMP

ICMP je od předchozích dvou protokolů velmi odlišný. Neslouží totiž k přenášení dat, ale k hlášení errorů. Jedním z nich může být například hlášení o tom, že packet nebyl úspěšně doručen. Není možné využívat určitý port. [6]

1.4 ARP

ARP slouží ke zjištění linkové adresy příjemce. V Ethernetu, který je hlavním zaměřením tohoto projektu, je získávána MAC adresa. [1]

2 Implementace

2.1 Zdroje

Program je sestaven převážně pomocí manuálu knihovny *pcap*[3]. Tato stránka podrobně popisuje postup při práci s pakety a také obsahuje odkaz na manuál ke každé z použitých funkcí. Další zdroje kódu či znalostí potřebných k vytvoření snifferu jsou vypsané přímo v komentářích kódu.

2.2 Parsování argumentů

První částí programu je parsování argumentů zadaných uživatelem při jeho spouštění. K tomu slouží funkce *parseargs*, která ukládá potřebná data do struktury *s_args* definované v hlavičkovém souboru.

2.3 Příprava před čtením dat

Před samotným čtením dat je třeba komunikaci otevřít pomocí funkcí *pcap_create* a *pcap_activate*. V případě přítomnosti argumentů zadaných uživatelem se pak ve funkci *set_filter* sestrojí a nastaví filtr pro čtené pakety. Implementace je omezena na TCP a UDP protokoly.

2.4 Zpracování paketů

Pakety jsou procházeny funkcí *pcap_loop*, která pak pro každý volá funkci *handler*. Ta zjišťuje protokol aktuálního paketu a podle něj volá funkci *tcp_packet* nebo *udp_packet*. Tyto funkce zjišťují vlastnosti paketu, jako jsou IP adresy, porty nebo délka, a vypíše je. Metoda *print_content*, volaná v každé z těchto funkcí, je převzaná[2] a upravená podle potřeb tohoto projektu, její implementace ale není kompletní.

2.5 Testování

Testování probíhalo pomocí nástroje *Wireshark*.

2.6 Help

Při zadání *-h* nebo *–help* program vypíše pomoc.

Reference

[1] Address Resolution Protocol. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Address_Resolution_Protocol

[2] How to code a Packet Sniffer in C with Libpcap on Linux. [online]. Dostupné z: https://www.binarytides.com/packet-sniffer-code-c-libpcap-linux-sockets/?fbclid=IwAR0qY12qCUFkhUJEiXQ83rbDOEijX1PDT5jQPIeaLY-cS67P3JFqk9f9cmk

[3] Man page of PCAP. [online]. Dostupné z: https://www.tcpdump.org/pcap3_man.html

[4] Transmission Control Protocol. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Transmission_Control_Protocol

[5] User Datagram Protocol. [online]. Dostupné z: https://cs.wikipedia.org/wiki/User_Datagram_Protocol

[6] What is the Internet Control Message Protocol (ICMP)? [online]. Dostupné z: https://www.cloudflare.com/learning/ddos/glossary/internet-control-message-protocol-icmp/