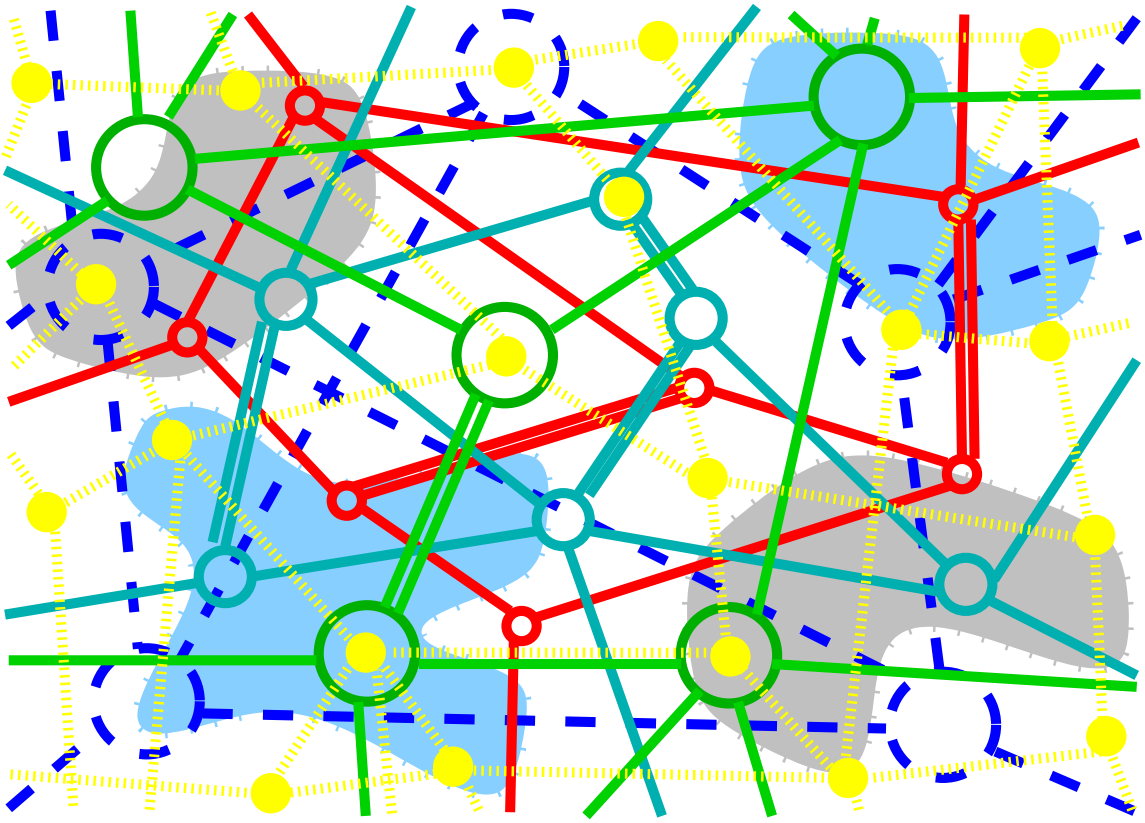
Instrukcja do laboratorium sieci komputerowych

Monitorowanie i analiza ruchu w sieci

dr inż. Piotr Arabas mgr inż. Jerzy Sobczyk dr inż. Edward Śliwa



30 lipca 2024

1 Ćwiczenie 1 Monitorowanie i analiza ruchu w sieci

1.1 Przygotowanie do zajęć

1. W czasie wykonywania ćwiczenia niezbędna będzie wiedza z zakresu następujących tematów:
   - budowa ramki EthernetII
   - podstawowe protokoły rodziny TCP/IP: ARP, IP, UDP, TCP
   - nagłówki wymienionych wyżej protokołów, znaczenie podstawowych pól
   - nawiązywanie i zamykanie połączeń TCP
2. Przed przystąpieniem do ćwiczenia należy zapoznać się z opisem (np. instrukcje dostępne w sieci www) programów wykorzystywanych w ćwiczeniu, w szczególności:
   - **Wireshark** – w zakresie umiejętności rozpoczęcia/zakończenia przechwytywania ramek na wybranym interfejsie sieciowym, sposobów elementarnej filtracji przechwytywanych pakietów, przeglądania plików z wcześniej przechwyconymi pakietami oraz interpretacji wyników
   - **netcat** (komenda `nc`) – w zakresie wysyłania/odbioru prostych pakietów w protokołach UDP i TCP
   - komenda `ip` z pakietu **iproute2** – w zakresie sprawdzania parametrów konfiguracyjnych interfejsu sieciowego

1.2 Logowanie do stanowiska roboczego

- Zalogować się na stanowisku roboczym. W ćwiczeniu pracujemy jako użytkownik `student`. Jeśli zajdzie konieczność użycia komend wymagających większych uprawnień, to (za zgodą prowadzącego) należy skorzystać z konta `ppkroot`. Hasła do obu kont zostaną podane w czasie zajęć.
- Po zalogowaniu na stanowisko robocze sprawdzić parametry interfejsu sieciowego (nazwę interfejsu, adres MAC, nr IP, maskę sieci, nr sieci, adres rozgłoszeniowy skierowany, nr IP bramy domyślnej). Parametry te należy zamieścić w sprawozdaniu z wykonania ćwiczenia.
- Program Wireshark można znaleźć w zakładce *Applications→Internet*. Pozostałe programy (np. `nc`, `ip`) są uruchamiane z linii komendy – zatem aby z nich skorzystać należy otworzyć okno terminala (*Applications→Terminal emulator*).

1.3 Protokół ARP

1. Usunąć ewentualne wpisy z tablicy ARP
2. Uruchomić program przechwytujący pakiety na interfejsie sieciowym komputera. W ćwiczeniu do przechwytywania pakietów wykorzystywany jest interfejs `eni`.
3. Wysłać (za pomocą programu `ping`) pojedynczy pakiet ICMP do sąsiedniego komputera.
4. Zakończyć przechwytywanie pakietów i przeanalizować uzyskane wyniki, zwracając uwagę tylko na pakiety przechwycone przed wysłaniem komunikatu ICMP (sugerujemy zapisanie wyników analizy do pliku w celu wykorzystania w następnym punkcie ćwiczenia).
   Odpowiedzieć na następujące pytania:
   - w jaki sposób nasz host znajdzie adres MAC odbiorcy pakietu wysyłanego komendą `ping`?
   - jakie adresy MAC nadawcy/odbiorcy znajdują się w nagłówku warstwy łącza danych zapytania i odpowiedzi ARP?
   - jakie adresy MAC i numery IP nadawcy/odbiorcy znajdują się w zapytaniu, a jakie w odpowiedzi ARP?

5. Powtórzyć punkty 2 do 4 (bez usuwania istniejących wpisów w tablicy ARP). Czy można zaobserwować jakieś różnice?
6. Powtórzyć punkty 1 do 4, wysyłając pakiet ICMP do komputera poza siecią laboratorium. Jak w tym przypadku wyglądają odpowiedzi na pytania z punktu 4?
7. Wysłać pakiet ICMP na adresy IP, pod którymi nie działają żadne hosty. Czy wystąpi różnica, a jeśli tak to jaka, między przypadkami adresu z zakresu sieci laboratorium i sieci poza laboratorium?

1.4 Protokoły IP i ICMP

1. Wykorzystując zbiór przechwyconych w poprzednim punkcie ramek prześledzić wymianę datagramów IP związanych z wysłaniem pakietu ICMP (komenda `ping`). Zwrócić uwagę w szczególności na najbardziej istotne pola nagłówka:
   - *Version*, *Header length*, *TTL*, *Source/destination address*, *Protocol*
   - pola związane z fragmentacją: *Identification*, *Total length*, *Flags*, *Fragment offset*
2. Wyjaśnić znaczenie pól nagłówka ICMP w zapytaniu i odpowiedzi.
3. Przechwycić wymianę datagramów IP wysyłając za pomocą komendy `ping` datagram o długości 3000 oktetów. Zwrócić uwagę na wartości pól nagłówka związanych z fragmentacją datagramu.

1.5 Protokół UDP

1. Upewnić się, że na sąsiednim stanowisku działa usługa DAYTIME
2. Rozpocząć przechwytywanie pakietów na interfejsie sieciowym komputera.
3. Wysłać do sąsiedniego komputera na adres usługi DAYTIME (za pomocą programu `nc`) pojedynczy pakiet UDP (zawartość obszaru danych pakietu może być dowolna, w szczególności pusta).
4. Zakończyć przechwytywanie pakietów i przeanalizować uzyskane wyniki, zwracając uwagę na wartości pól nagłówka UDP w zapytaniu i odpowiedzi.

1.6 Protokół TCP

1. Rozpocząć przechwytywanie pakietów na interfejsie sieciowym komputera.
2. Wysłać do sąsiedniego komputera na adres usługi DAYTIME (za pomocą programu `telnet` lub `nc`) pojedynczy pakiet TCP (zawartość obszaru danych pakietu może być dowolna, w szczególności pusta).
3. Zakończyć przechwytywanie pakietów i przeanalizować uzyskane wyniki. Zwrócić szczególną uwagę na:
   - etap nawiązania połączenia TCP (potrójne uzgodnienie) – które pola nagłówka TCP grają tu istotną rolę?
   - etap wymiany danych,
   - zakończenie połączenia TCP – które pola nagłówka TCP są tu istotne?
4. Jakie opcjonalne pola pojawiają się w nagłówkach IP i TCP w pakietach obserwowanych w czasie ćwiczenia? Wyjaśnić ich znaczenie.

1.7 Sprawozdanie z wykonania ćwiczenia

- Z wykonania ćwiczenia należy przygotować sprawozdanie w postaci pliku .pdf.
- W sprawozdaniu dla kolejnych punktów ćwiczenia należy zamieścić istotne informacje i dane, w szczególności komendę, która spowodowała wygenerowanie ruchu sieciowego, zdekodowane ramki (można ograniczyć się do istotnych w danym punkcie pól), i ewentualne komentarze lub wnioski.
- Sprawozdanie należy zamieścić na serwerze `studia`.

1.8 Uwagi

- Ćwiczenia wykonywane są jednoosobowo. Potrzebne w ćwiczeniu numery IP sąsiednich stanowisk pochodzą z zakresu 192.168.96.201 ÷ 192.168.96.216.
- Przy określaniu adresów warstwy sieciowej można używać zarówno numerów IP (w notacji kropkowo-dziesiętnej) jak i nazw symbolicznych – prostych (jednoczłonowych) i domenowych (wieloczłonowych). Sugeruje się jednak użycie numerów IP - nie będzie wówczas generowany ruch DNS, który może nieco zaciemniać uzyskane wyniki.
- Przy korzystaniu z programu **Wireshark** pożyteczne może okazać się włączenie filtrów - uniknie się w ten sposób rejestracji nadmiernej liczby nieistotnych z punktu widzenia ćwiczenia pakietów.
- Zalecamy zapis przechwyconych (niezdekodowanych) pakietów sieciowych do pliku, następnie zapisanie tak uzyskanych plików na własny nośnik lub własne konto na serwerze sieciowym. Pozwoli to – w przypadku braku czasu – na dokończenie sprawozdania w domu.
- Przenoszenie uzyskanych wyników do sprawozdania może być wykonane w jeden z poniższych sposobów:
  - metodą „copy-paste" z okna terminala (nie zalecamy tej metody – jest żmudna, podatna na błędy i zwykle nie pozwala na usunięcie nieistotnych informacji),
  - przez przekierowanie uzyskanych wyników do pliku/plików tekstowych.
- Niektóre z programów (np. `ifconfig`, `arp`) oraz niektóre opcje programu `ip` wymagają uprawnień administratora – zalecamy w takim przypadku zalogowanie się jako użytkownik `ppkroot`.

1.9 Pożyteczne komendy i ich opcje

Polecenia systemu Linux	
`arp`	Komenda służąca do wyświetlenia i manipulacji wpisami w tablicy ARP. Bez dodatkowych opcji wyświetlana jest zawartość tablicy ARP.
`arp -n`	Opcja `-n` powoduje, że adresy warstwy sieciowej wyświetlane są jako numery IP a nie nazwy symboliczne (zalecamy korzystanie z tej opcji)
`arp -d` *ip*	Opcja `-d` umozliwia usunięcie wpisu odnoszącego się do podanego numeru *ip*
`ip`	Podstawowa komenda systemowa pozwalająca m.in. wyświetlić i ustawić parametry interfejsów sieciowych. W tym ćwiczeniu potrzebna jest tylko do uzyskania informacji o parametrach interfejsów sieciowych.
`ip address show`	Wyświetlenie informacji o parametrach interfejsów sieciowych
`ip neighbour`	Wyświetlenie informacji zawartości tablicy ARP
`ip neighbour flush dev` *dev*	Usunięcie wpisów z tablicy arp związanych z interfejsem *dev*
`ifconfig -a`	Starsza komenda o funkcjonalnosci zbiiżonej do `ip address show` (zalecamy jednak użycie komendy `ip` – w nowszych wersjach systemu Linux komenda `ifconfig` nie zawsze jest dostępna).
`nc`	Uniwersalne narzędzie diagnostyki sieci, w niniejszym ćwiczeniu służy do wysyłania/odbioru prostych pakietów w protokołach UDP i TCP.
`nc` *ip port*	Nawiązanie połączenia TCP z hostem o numerze *ip* na porcie *port* (ew. dane są przekazywane przez standardowe wejście)
`nc -u` *ip port*	Wysłanie datagramu UDP do hosta o numerze *ip* na port *port* (ew. dane są przekazywane przez standardowe wejście)
`telnet` *ip port*	Systemowa komenda przydatna do testowania połączenia TCP.
`ping`	Systemowa komenda wysyłająca komunikaty ICMP ECHO REQUEST i oczekująca na odpowiedzi.
`ping -c` *n*	Argument opcji `-c` *n* jest liczbą wysyłanych żądań ICMP ECHO REQUEST (bez podania tej opcji żądania będą wysyłane aż do chwili przerwania przez użytkownika przez Ctrl-C)
`ping -s` *rozm*	Argument *rozm* jest rozmiarem pakietu (domyślnie 56 oktetów)