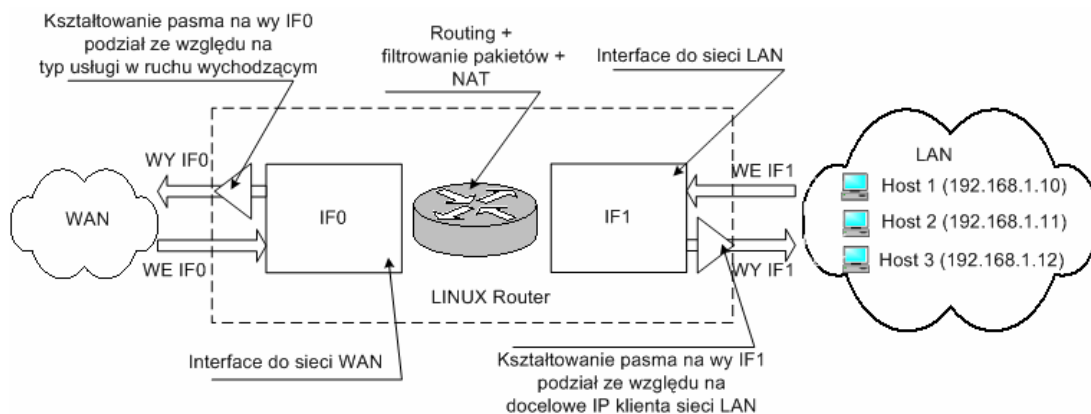


„Kontrola ruchu peer-to-peer w sieciach dostępowych”

Problemy powodowane przez użytkowników programów typu p2p są dosyć powszechne dla administratorów sieci dostępowych. Zaledwie kilka osób używających p2p w jednym czasie jest w stanie skutecznie przyblokować sieć, która w normalnych warunkach jest w stanie obsłużyć kilkudziesięciu (w zależności od szybkości łącza) klientów. Rozwiązanie tego problemu jest oczywiście możliwe, jednak wymaga zrozumienia podstawowych zagadnień dotyczących kontroli ruchu sieciowego, które staram się poniżej przybliżyć.

Pierwsza istotna rzeczą jest to, że kontrolować można jedynie ruch wychodzący z interfejsu, natomiast nie ma możliwości wprost kontrolować ruchu przychodzącego. Jeśli wziąć pod uwagę interfejsy WAN i LAN routera, to możemy kształtować jedynie pasmo na WY IF0 (WAN) oraz na WY IF1 (LAN). Pamiętając o tym że ruch odbywa się w obu kierunkach, należy na odpowiednim interfejsie kształtować pasmo w określonym kierunku transmisji. Łatwo zauważyć, że kształtując pasmo na WY IF1 kształtujemy ruch wchodzący (download) do sieci LAN oraz kształtując pasmo na WY IF0 kształtujemy ruch wychodzący do Internetu (upload).



Rys. 1 – Podstawowa forma kontroli przepływu na routerze linuxowym

W celu zapewnienia każdemu użytkownikowi odpowiedniej szybkości łącza, należy stworzyć odpowiednie limity na WY IF1, będą to limity dotyczące download, a kryterium podziału będzie IP użytkownika. Z przeciwnym kierunkiem transmisji nie można uczynić podobnie wprost w przypadku korzystania z maskarady IP, ponieważ kształtujemy pasmo na WY IF0, a więc adresy klientów z sieci LAN są już „zamaskowane” adresem publicznym routera. Pozostaje więc podział pasma ze względu na typ usługi. I tak można wydzielić fragmenty dla poszczególnych usług (usługi interaktywne, poczta, www, ftp), a kryterium podziału będzie nr portu na jakim działa dana usługa. Gdyby pominąć ten podział, wszystkie usługi były by traktowane jednakowo, a należy pamiętać że niektóre usługi powinny być uprzywilejowane, jeśli chcemy zapewnić odpowiednią wydajność łącza. Szczególnie istotne jest zapewnienie pierwszeństwa tzw pakietom potwierdzeń ACK, mówiących o poprawności transmisji. Jeśli pakiety ACK utkną gdzieś po drodze, wówczas download, w tym przeglądanie stron www będzie mocno spowolniony lub wręcz niemożliwy,

Łatwo zauważyć wady powyższego rozwiązania:

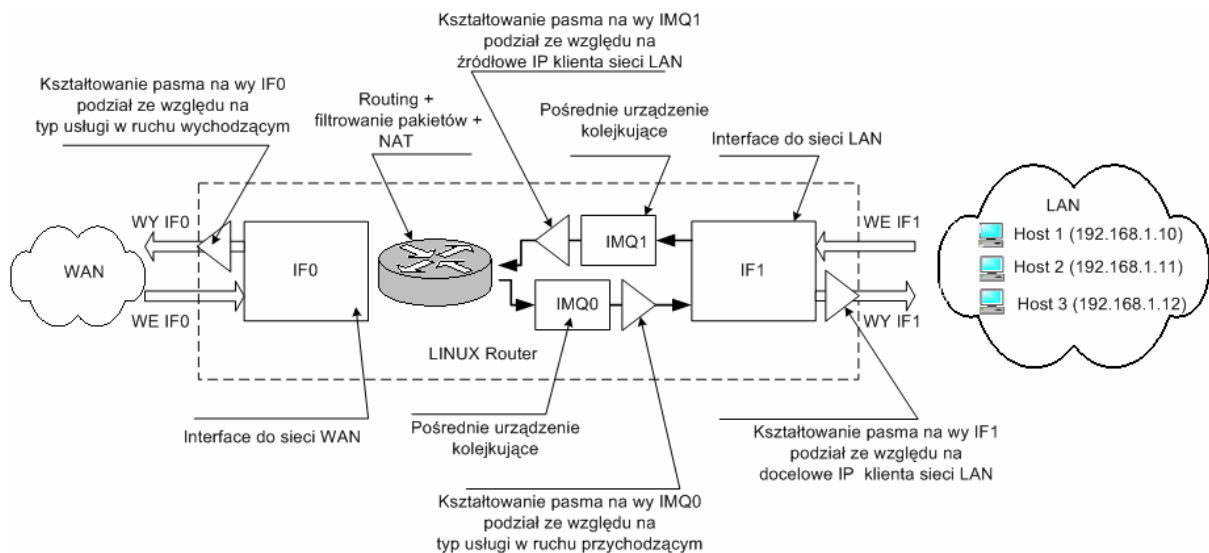
- Nie ma możliwości limitowania transmisji upload per user
- Brak wpływu na to jaką część pasma zużywają użytkownicy na poszczególne usługi w ramach swoich limitów download.

Stąd biorą się sytuacje, w których kilku użytkowników korzystających z p2p i przyznanych im limitów zapycha całe łącze w trybie download.

Rozwiązaniem obu problemów, było by zastosowanie podziału dwustopniowego, w obu kierunkach transmisji zarówno ze względu na typ usługi jak i IP użytkownika.

Poza tym, konieczny jeszcze jest mechanizm klasyfikujący ruch p2p, trzeba pamiętać że trudno go sklasyfikować na podstawie portów tak jak można to zrobić z innymi usługami. Ponieważ korzysta on w zasadzie z dowolnych portów, nawet tych dedykowanych do standardowych usług, co powoduje szereg kłopotów.

Dwustopniowy podział jest możliwy dzięki zastosowaniu mechanizmu tzw pośredniego urządzenia kolejującego IMQ. IMQ jest dodatkowym interfejsem emulowanym programowo, które pozwala na ruch w jednym tylko kierunku. Podłączając dwa takie urządzenia zaraz za interfejsem IF1 w systemie, każde dla jednego kierunku transmisji, uzyskuje się możliwość dodatkowego kontrolowania pasma. W konfiguracji jak na rysunku, sterując przepływem danych na WY IMQ1, kontroluje się de facto ilość danych przychodzących do interfejsu IF1 od strony LAN. (upload) Podobnie sterując przepływem na WY IMQ0 w rzeczywistości kontroluje się transfer do interfejsu IF0 od strony WAN (download).



Rys. 2 – Kontrola przepływu na routerze linuxowym z wykorzystaniem IMQ

Dzięki temu rozwiązaniu eliminuje się możliwość tzw zapchania łącza od strony we IF1 (nadmierny upload użytkowników sieci) poprzez przypisanie każdemu użytkownikowi indywidualnego limitu na upload. Po drugie uzyskuje się możliwość dowolnego kształtowania pasma download ze względu na rodzaj usługi, można więc przeznaczyć na ruch p2p np. 25% pasma, wobec czego nie ma możliwości żeby kilku użytkowników p2p korzystających jednocześnie wysycało pasmo.

Przykład:

Łącze 1Mbit

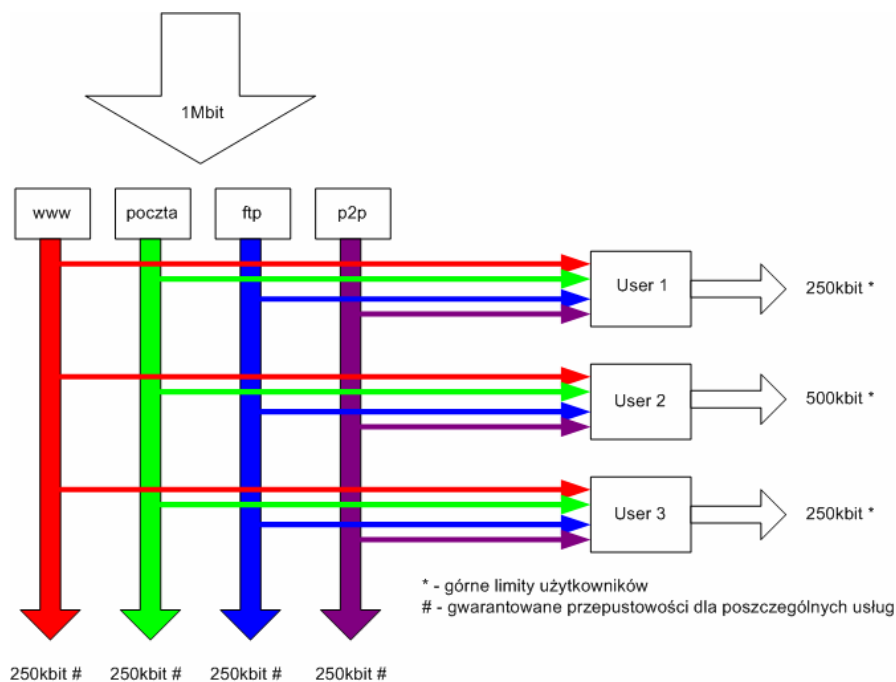
Limity użytkowników:

- User 1 250kbit
- User 2 500kbit
- User 3 250kbit

Łatwo zauważyć, że w sytuacji gdy użytkownicy 1 -3 jednocześnie będą korzystać z p2p, są w stanie wysycać całe łącze, efekt – pozostałym użytkownikom ze zrozumiałych względów strony będą się otwierać bardzo wolno.

Teoretycznie mechanizm kolejkowania htb w momencie zapotrzebowania innych użytkowników na pasmo powinien zmniejszyć limity użytkowników 1-3, tak aby inni użytkownicy mogli skorzystać z łącza. Trzeba jednak pamiętać że nie dzieje się to natychmiast, i w pierwszym momencie strony będą się ładować bardzo powoli lub wcale.

Wydzielając na usługę p2p fragment pasma o przepustowości 250kbit, unika się takiego niebezpieczeństwa, ponieważ poza indywidualnymi limitami użytkowników jest jeszcze limit zbiorowy na usługę p2p, który chroni innych użytkowników, gwarantując przepustowość dla każdej z usług. Tak więc w sytuacji gdy wszyscy trzej użytkownicy będą chcieli skorzystać z p2p to pomimo swoich wysokich limitów nie zajmą oni całego pasma, do ich dyspozycji będzie właśnie 250kbit, pozostała część łącza pozostanie zarezerwowana dla innych usług.



Rys. 3 – Dwustopniowy podział pasma – typ usługi/użytkownik

W przypadku łącza symetrycznego podany mechanizm można zastosować dla obu kierunków transmisji, dla łącz asymetrycznych raczej niewskazane jest limitowanie uploadu per user, a jedynie ze względu na rodzaj usługi.

Klasyfikacja i kontrola ruchu p2p jest możliwa dzięki zastosowaniu mechanizmów nieobecnych w standardowym jądrze systemu linux. Mowa tu o module ipp2p oraz l7 filter. Są to filtry iptables klasyfikujące pakiety na podstawie ich nagłówków. Dodatkowo konieczne jest zastosowanie pakietu łat znanych pod nazwą patch-o-matic. Należy jeszcze wspomnieć, że dzięki mechanizmowi klasyfikującemu ruch p2p można go zablokować całkowicie dla niektórych protokołów.

Implementacja wymienionych składników (ipp2p, imq, l7 filter, patch-o-matic) wiąże się z samodzielną kompilacją jądra. Dodatkowo konieczne jest odpowiednie skonfigurowanie, ponieważ nie są to gotowe działające programy, a jedynie narzędzia służące zaawansowanym użytkownikom.