Ochrona sieci operatorów internetowych

Dobre praktyki

Łukasz Bromirski
lbromirski@cisco.com

Wrocław, 21 października 2006

Agenda

- **Ochrona sieci operatora**
  - **...i słów parę o ochronie samych urządzeń**
- **Ochrona klientów i ich usług**
- **Parę innych, luźnych uwag dotyczących metodyki ochrony sieci**
- **Q&A**

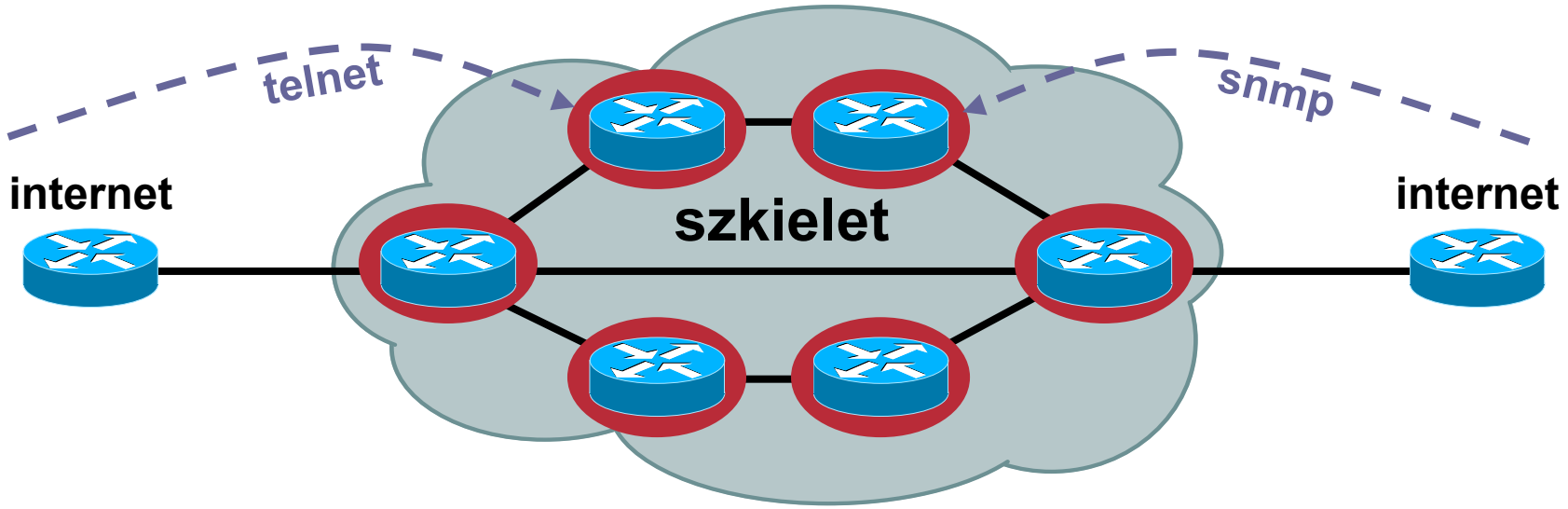
Disclaimer

Sesja zawiera jedynie wybrane elementy dobrych praktyk.

Nie stanowi kompendium, a jedynie zestaw luźno powiązanych zagadnień dających się poruszyć w ciągu 45 minut.

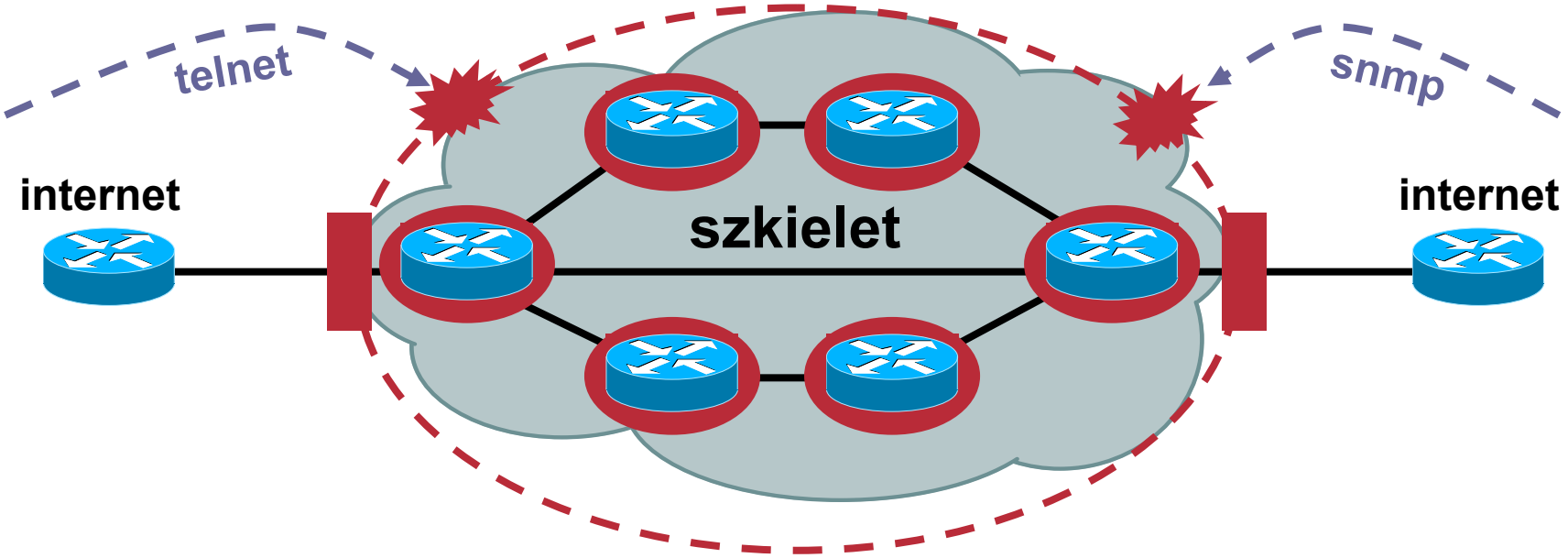
Ochrona sieci operatora

Jak nie projektować sieci operatora...



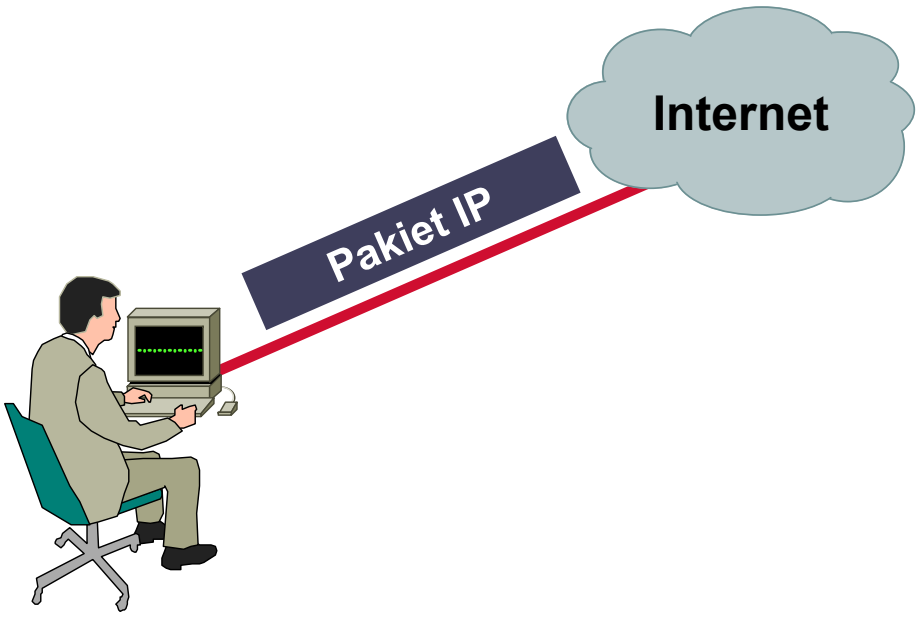
- **Routery zabezpieczane indywidualnie**
- **Każdy router dostępny i pośredniczący w wymianie ruchu od klienta do Internetu**

Jak projektować sieć dla operatora?



- **Podstawowa sprawa to separacja ruchowa:**
  - **ruch nasz**
  - **ruch naszych klientów**

Wszystko zaczyna się od pakietu...

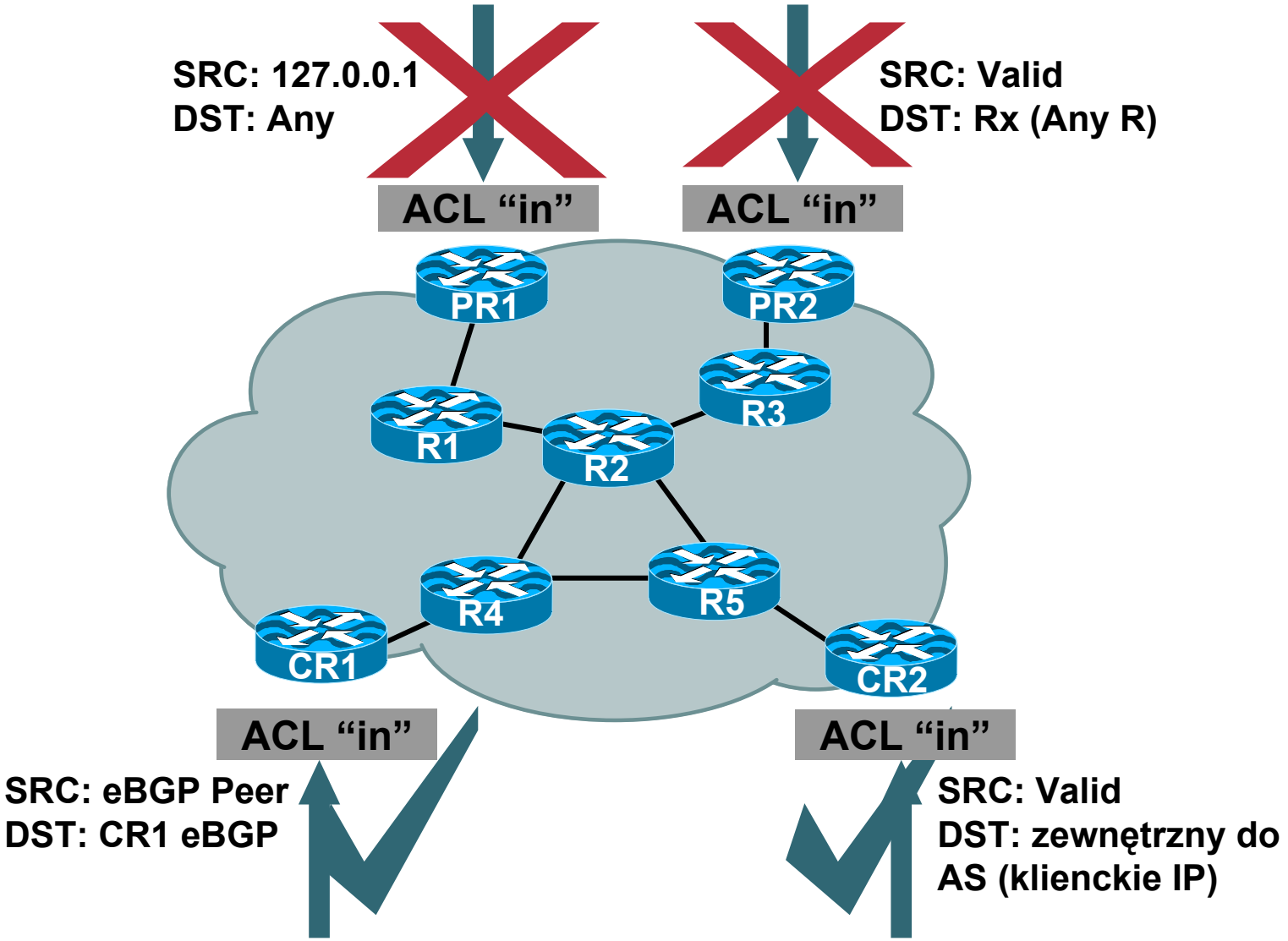


- **W momencie w którymś pakiet zostanie stworzony i wysłany do Internetu, ktoś, gdzieś, musi zrobić jedną z dwóch rzeczy:**
  - **Dostarczyć pakiet**
  - **Odrzucić pakiet**

ACL „infrastrukturalne”

- **Dlaczego Twoje routery powinny zajmować się całym ruchem?**
- **Stwórz listę protokołów które pozwalają Twojej sieci pracować i wykorzystaj separację ruchową**
  - **Na przykład: peeringi eBGP/iBGP, GRE, IPsec, OSPF, etc.**
  - **Wydzielenie osobnych przestrzeni adresowych IP dla swojej infrastruktury i sieci klienckich (w tym adresów do obsługi NAT) zdecydowanie pomaga**
  - **Pomaga również sumaryzacja podsieci – krótsze reguły ruchowe**

ACL „infrastrukturalne” w akcji...



ACL „infrastrukturalne”
Przykład

```
! Nasze własne IP od innych sieci? Nie ma mowy:
access-list 101 deny ip our_CIDR_block any
! 0.0.0.0 czy 127/8 jako sieci źródłowe?! Nie ma mowy:
access-list 101 deny ip host 0.0.0.0 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
! RFC1918 – chyba żartujesz:
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.0.15.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
! eBGP z konkretnego IP z zewnątrz:
access-list 101 permit tcp host peerA host peerB eq 179
access-list 101 permit tcp host peerA eq 179 host peerB
! Pozostała infrastruktura – brak dostępu z zewnątrz:
access-list 101 deny ip any core_CIDR_block
! Pozostały ruch – przechodzący kliencki
access-list 101 permit ip any any
```

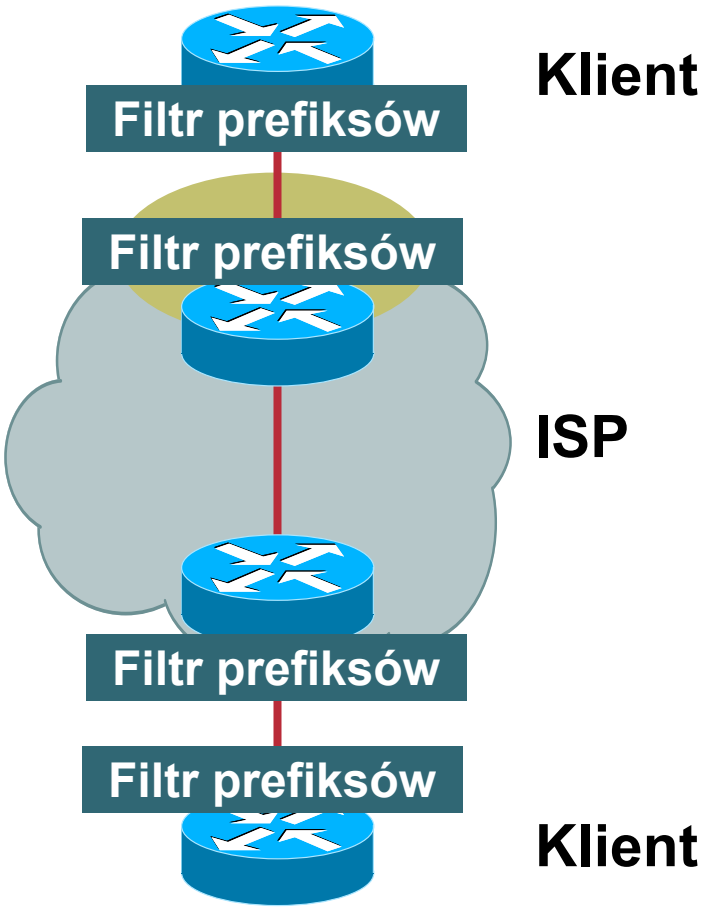
Ochrona protokołów routingu

- **Współdzielony klucz w pakietach routingu**
  - **Czystym tekstem – chroni tylko przed błędami w konfiguracji**
  - **Message Digest 5 (MD5)—zapobiega potencjalnym atakom w warstwie protokołu routingu**
- **Wiele kluczy per proces**
- **BGP, IS-IS, OSPF, RIPv2 i EIGRP obsługują**
- **Często nie używane**
  - **„Nie mieliśmy żadnych ataków”**
  - **„To obciąża zasoby”**

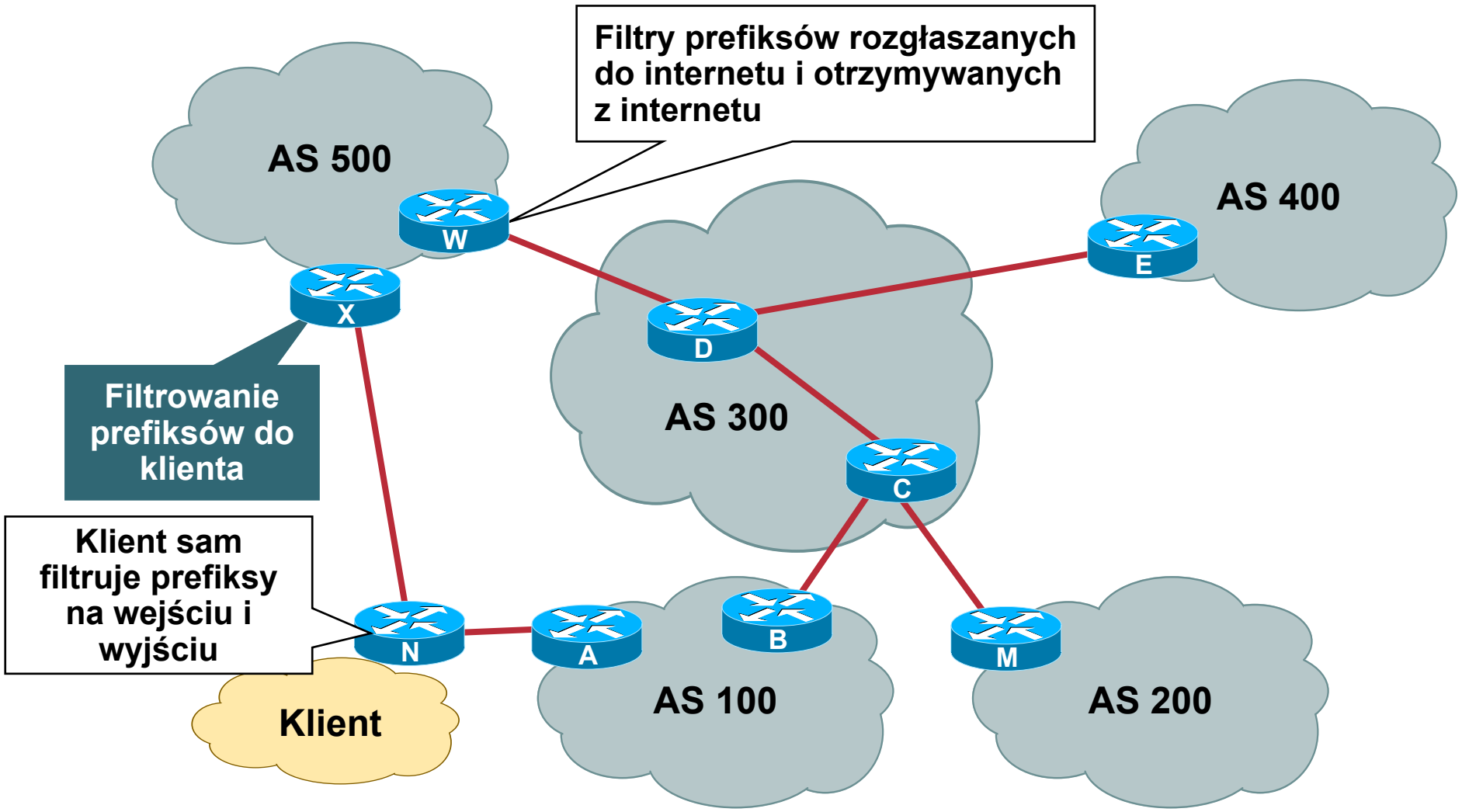
Filtrowanie prefiksów

- **Prefiksy otrzymywane od operatorów i wysyłane do operatorów (i klientów) należy kontrolować**

  ...dodatkowy bonus to prawidłowe działanie mechanizmów typu uRPF



Gdzie filtrować prefiksy?



Ochrona urządzeń sieciowych operatora

Mechanizmy bezpieczeństwa na routerach

- **Wiele organizacji publikuje własne zalecenia dotyczące najlepszych praktyk**
- **...skorzystaj:**

  http://www.first.org/resources/guides/

  http://www.sans.org/resources/policies/

  http://www.ietf.org/html.charters/opsec-charter.html
- **Dokumenty te opisują ‘hardening’ platformy, nie kompleksowe podejście do zapewnienia sieci bezpieczeństwa**

Tradycyjne metody
Hardening routerów

- Wyłączenie nieużywanych usług

```
no service tcp-small-servers
no cdp run
```

- ACL do VTY
- ACL na dostęp do SNMP
- Widoki SNMP
- Wyłączyć dostęp RW

  ...lub używać SNMPv3

- Wygasanie sesji, które umarły

```
service tcp-keepalives-in
```

- Polityka QoS na interfejsach skierowanych w stronę brzegu sieci (klientów i sieci zewnętrznych)
- Wykorzystaj systemy AAA (Authentication, Authorization i Accounting)
- Wyłączenie nieużywanych mechanizmów sieciowych, włączonych domyślnie na interfejsach sieciowych urządzenia

Tradycyjne metody
Hardening routerów

- Testy na adresie źródłowym (RFC2827/BCP38, RFC3704/BCP84)

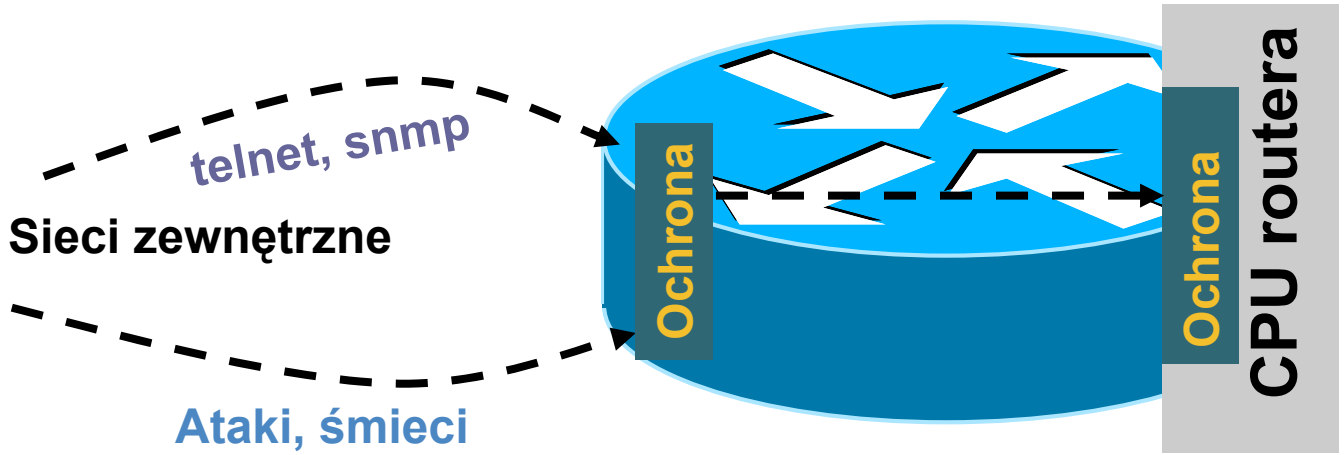
```
ip verify unicast source reachable-via {any|rx}
cable source-verify [dhcp]
ip verify source [port-security]
```

- Wyłącz source-routing

```
no ip source-route
```

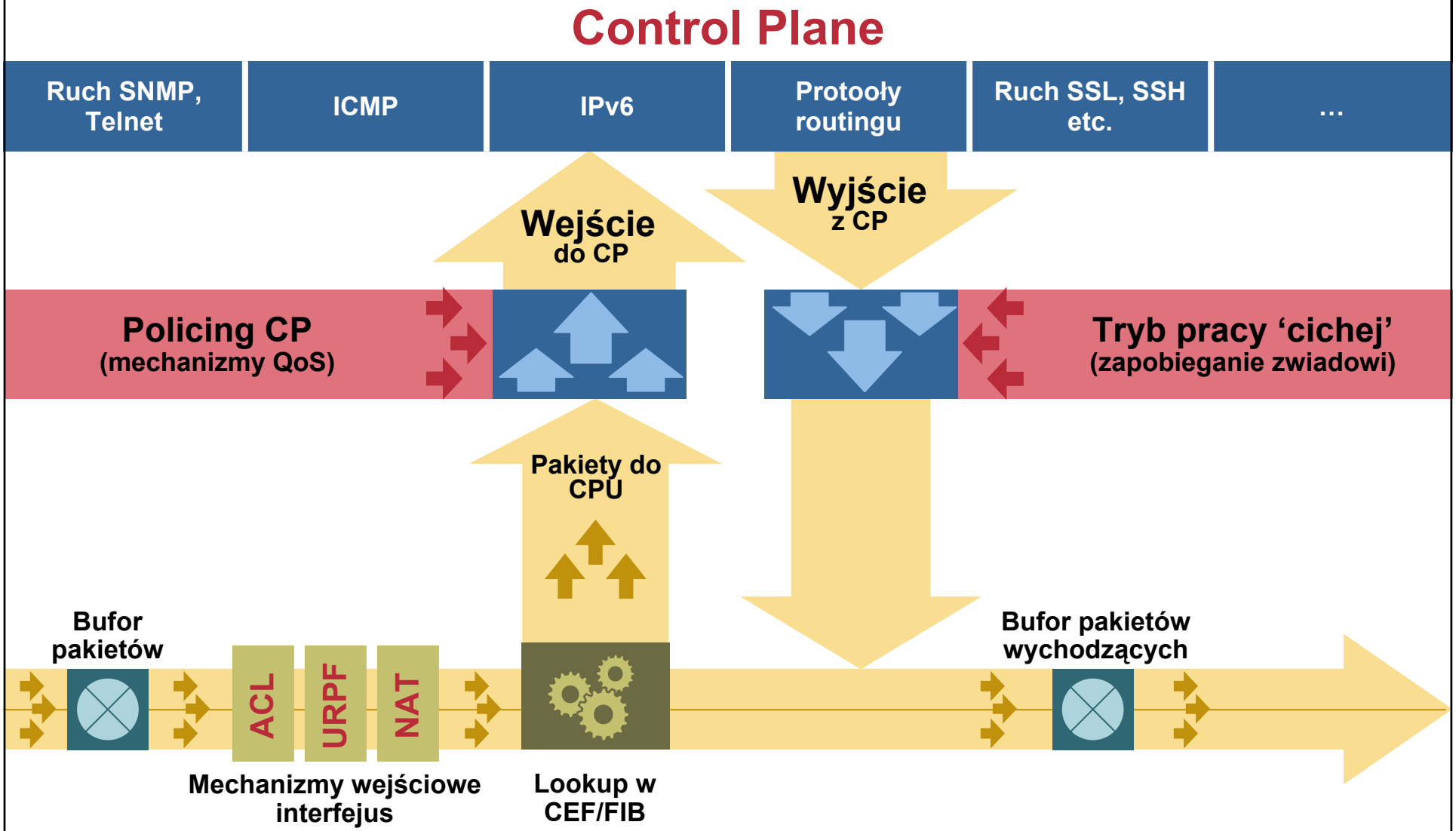
- Filtrowanie prefiksów na peerach eBGP
- BGP dampening (!)
- MD5 na sesjach BGP i IGP
- Inne mechanizmy specyficzne dla platformy:
  - CoPP
  - Przydział czasu obsługi przez CPU ruchu i innych procesów
  - Selective Packet Discard

Control plane vs data plane



- **Na platformach programowych konieczny podział na część realizującą funkcje routera i część zapewniającą forwarding pakietów**
- **W Cisco IOS – mechanizm CoPP**

Ochrona control plane

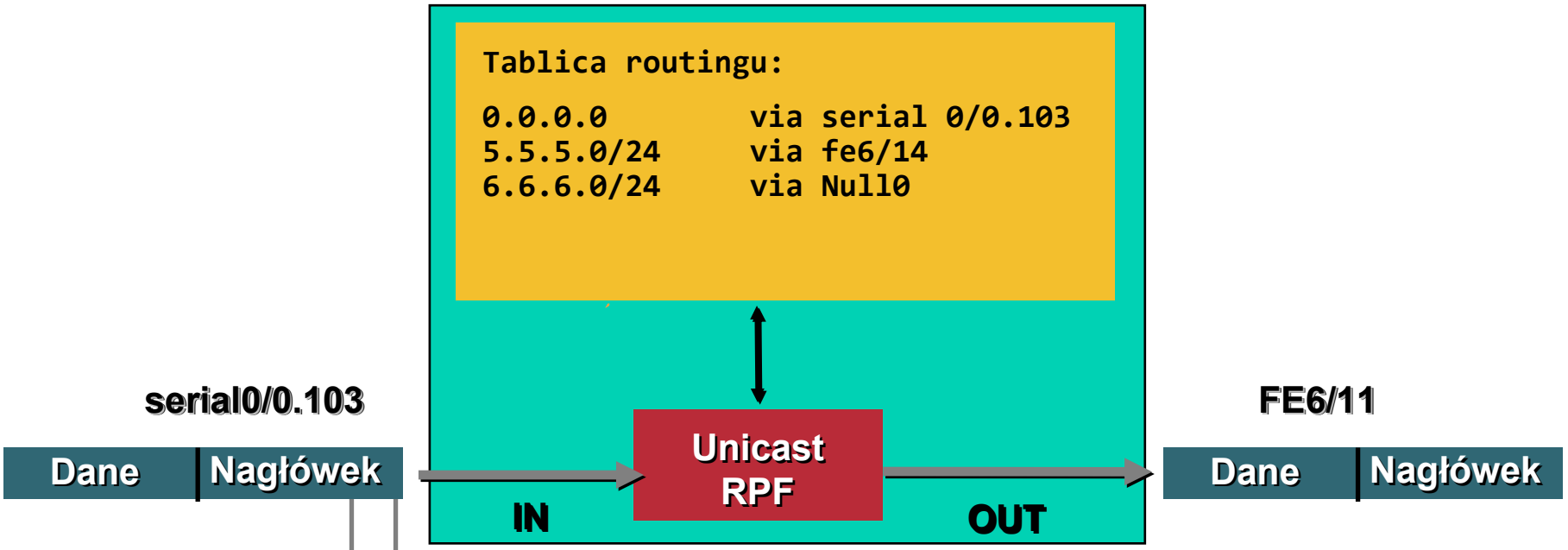


Ochrona klientów i usług

Interfejs w stronę klienta

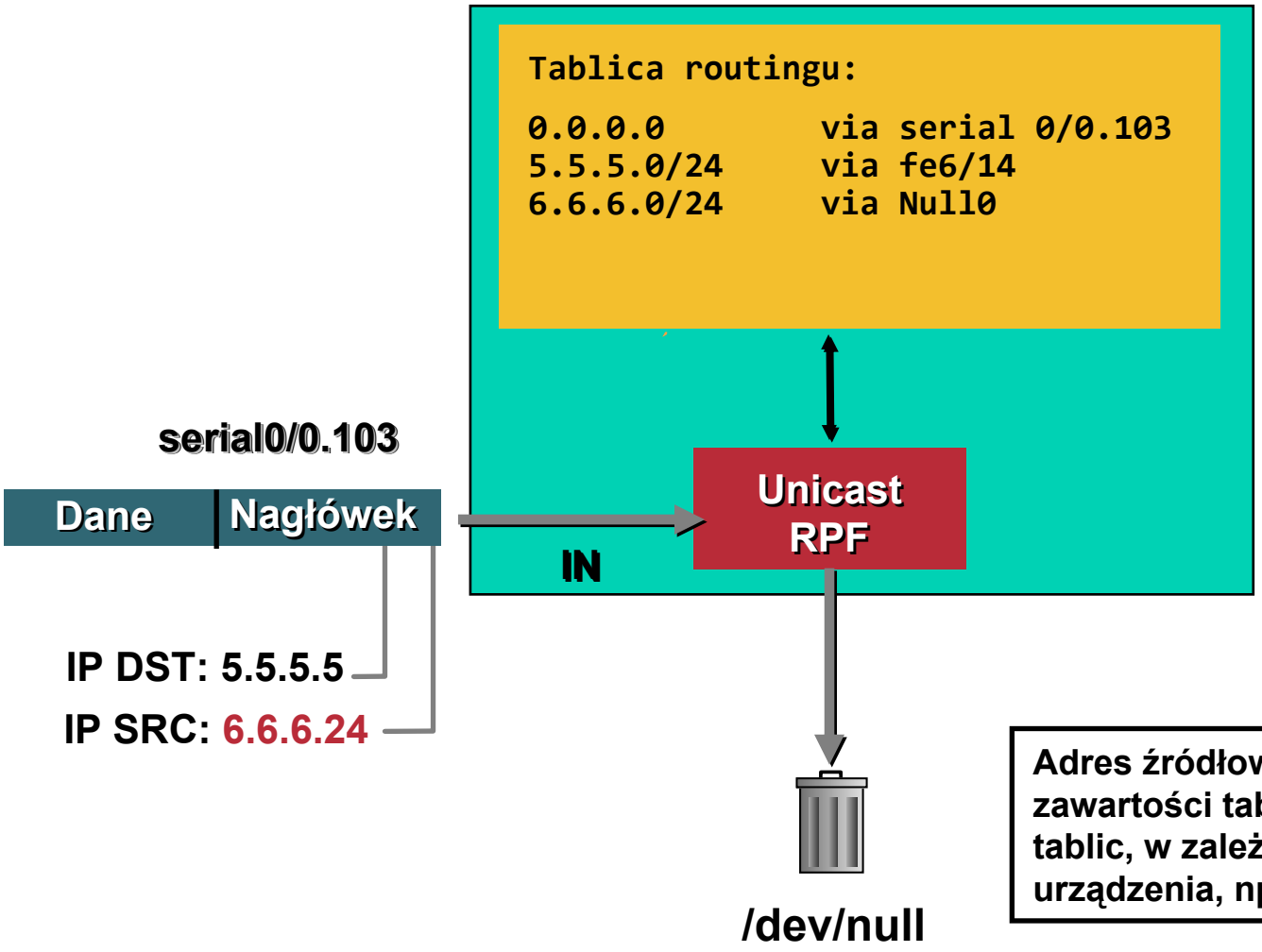
- **Absolutnie koniecznie:**
  - **uRPF (!)**
- **Inne pomysły do rozważenia:**
  - **wycięcie ruchu do/z TCP/UDP 135-139**
  - **wycięcie ruchu do/z TCP 445 (SMB over TCP)**
  - **adresacja P2P /30 lub wręcz /31**
  - **mechanizm QoS – rate limiting per protokół, lub ilość nawiązywanych sesji na sekundę**
  - **oznaczanie pakietów w IP DSCP dla konkretnych klas usług**

Unicast RPF
Tryb strict



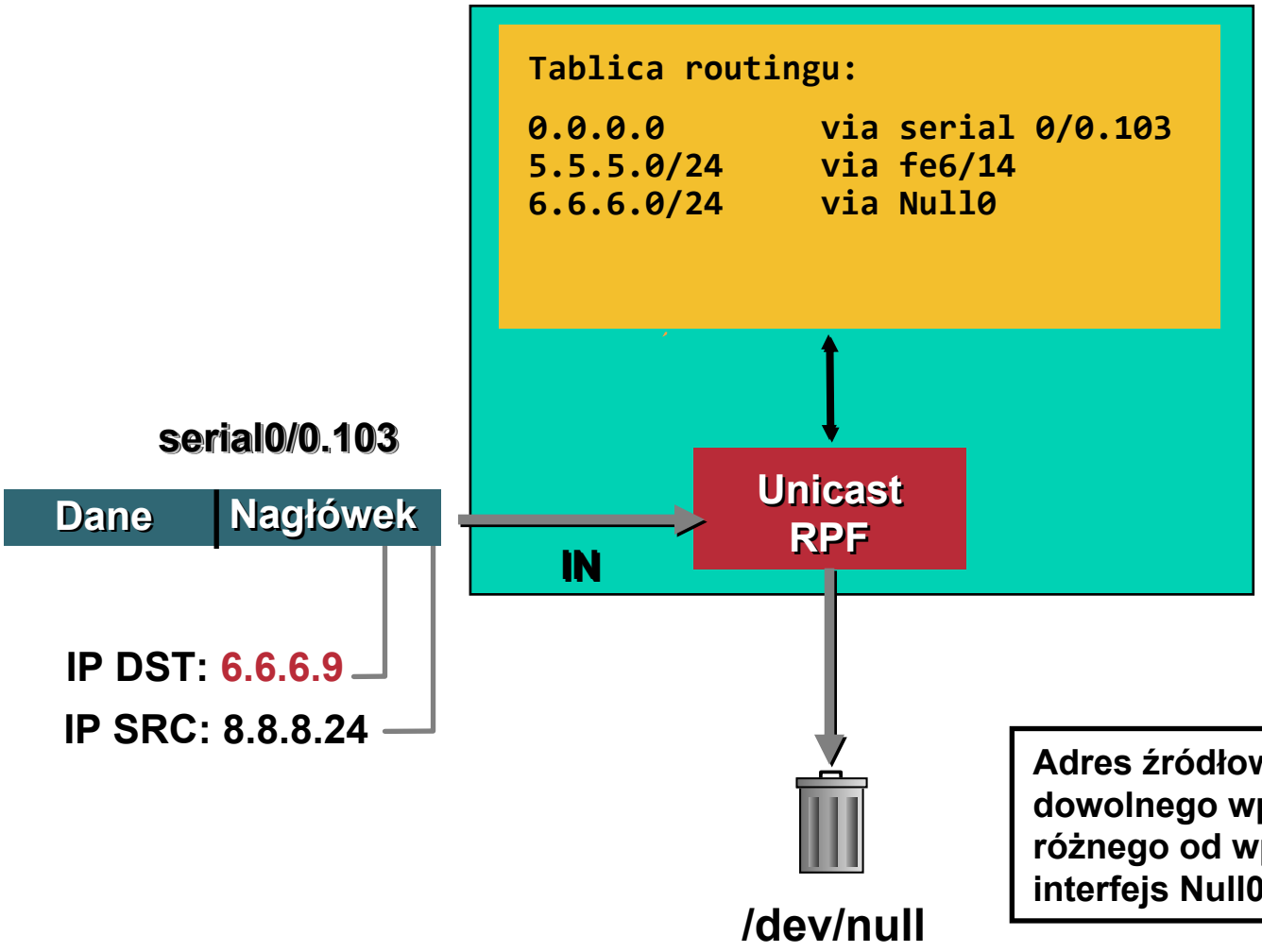
Adres źródłowy musi pasować do zawartości tablicy routingu (i innych tablic, w zależności od architektury urządzenia, np. CEF/adj)

Unicast RPF
Tryb strict



Adres źródłowy musi pasować do zawartości tablicy routingu (i innych tablic, w zależności od architektury urządzenia, np. CEF/adj)

Unicast RPF
Tryb loose



Adres źródłowy musi pasować do dowolnego wpisu w tablicy routingu różnego od wpisu kierującego pakiet na interfejs Null0

Konfiguracja uRPF

- **W zależności od systemu operacyjnego (i często konkretnego filtra pakietów) konfiguracja uRPF:**

  FreeBSD, tryb „strict/loose”:

  ```
  deny log ip from any to any not [verrevpath|versrcpath] in via em0
  ```

  Cisco, tryb „strict/loose”:

  ```
  ip verify unicast source reachable via [rx|any] [allow-default]
  ```

  Linux, tryb „strict/loose”:

  ```
  echo [1|2] > /proc/sys/net/ipv4/conf/(all|ethX)/rp_filter
  ```

  JunOS, tryb „strict/loose”:

  ```
  [edit interface ge-0/0/0 unit 0 family inet]
          rpf-check { mode loose; }
  ```

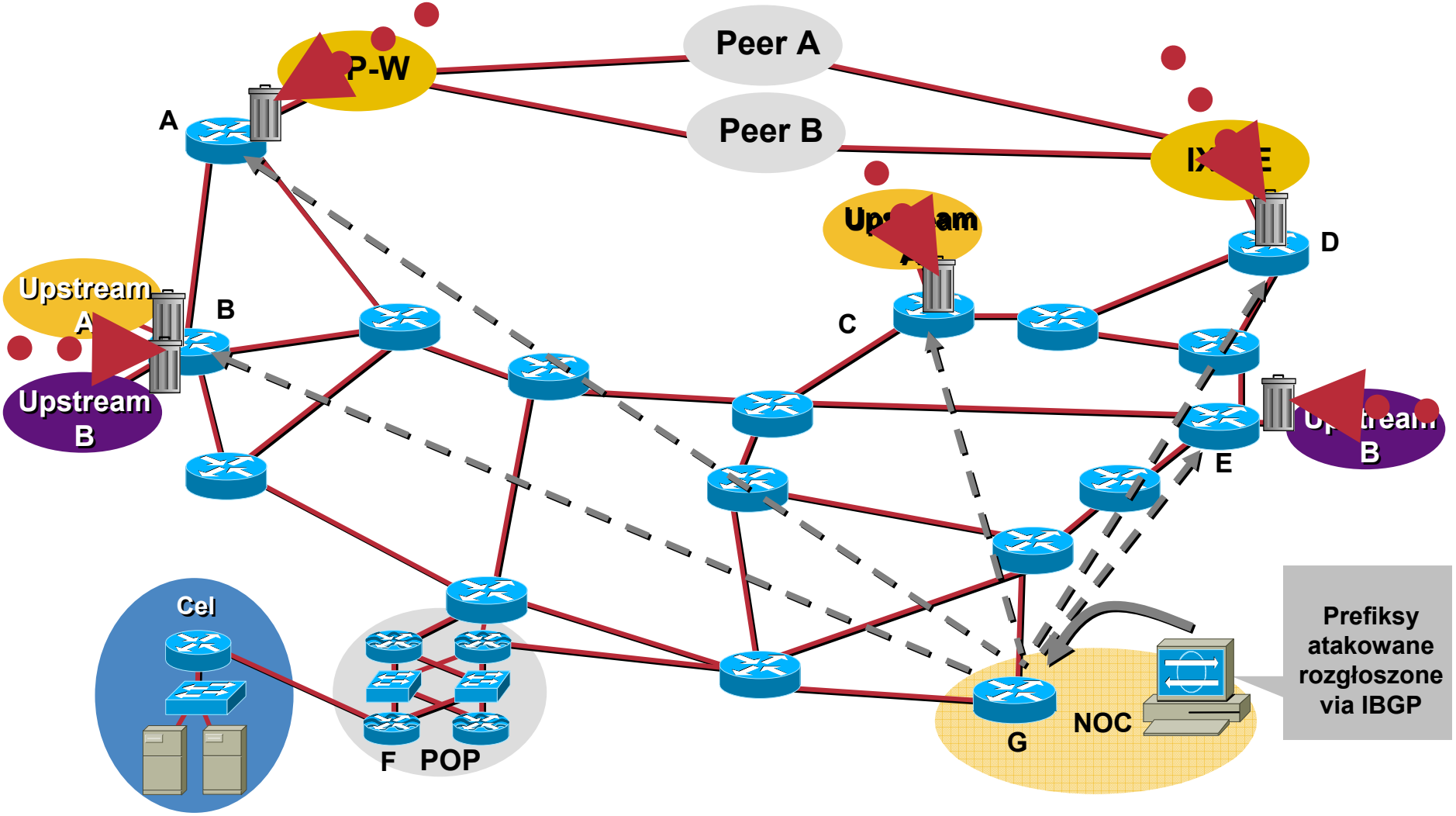
uRPF dla FreeBSD niezależny od filtra pakietów:
http://lukasz.bromirski.net/projekty/patches.html

Interfejs w stronę klienta
Przykład konfiguracji – Cisco IOS

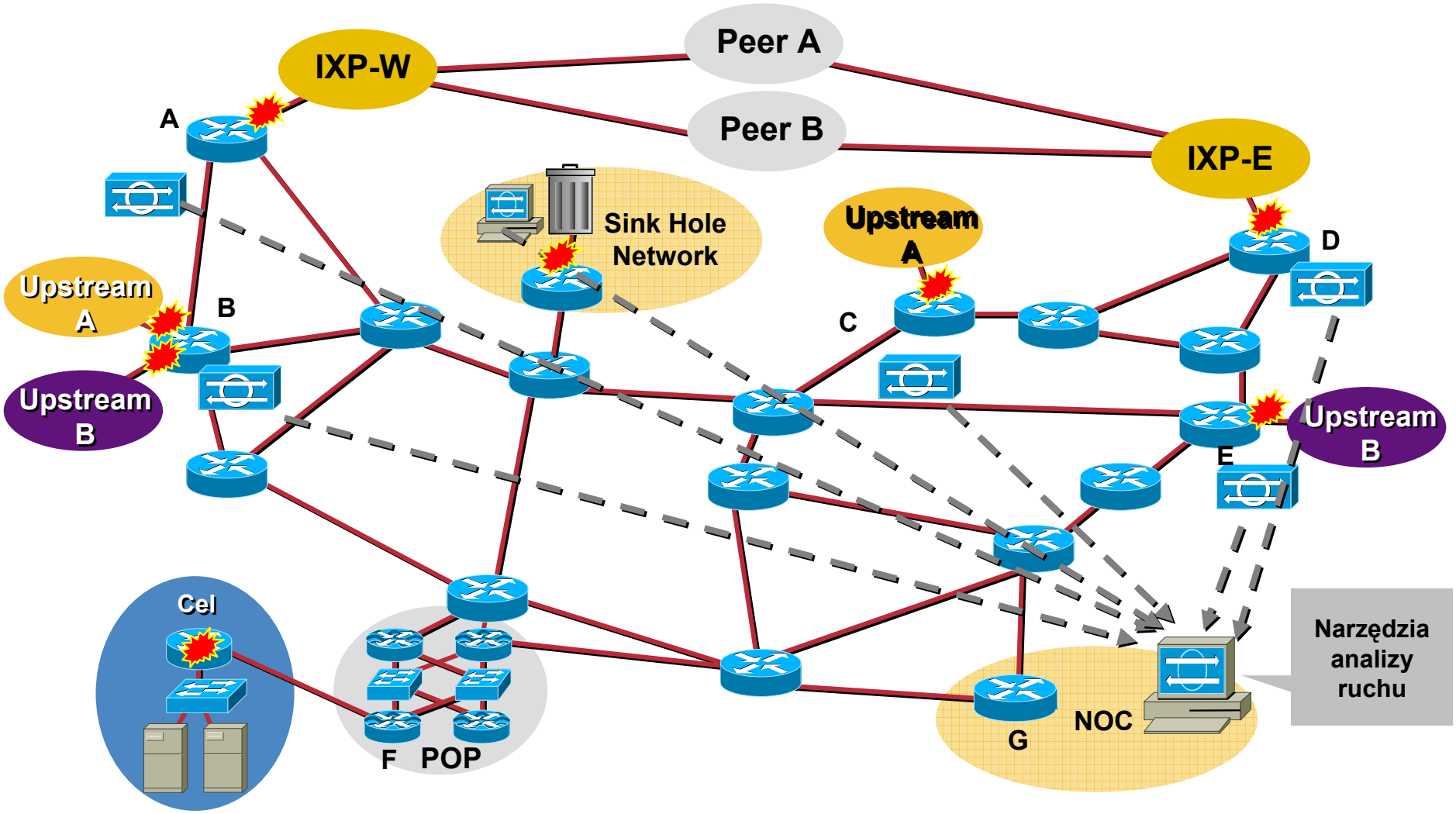
```
interface fastethernet 0/0.10
 desc Klient usługi 1024/1024
 encapsulation dot1q
 ip address 10.10.10.1 255.255.255.252
 ip verify unicast source reachable via rx
 ip access-list Kill_Bad_Traffic

 rate-limit input access-group name R-TCP
   768000 64000 64000 conform-action transmit exceed-action drop
 rate-limit input access-group name R-UDP
   128000 64000 64000 conform-action transmit exceed-action drop
 rate-limit input access-group name R-ICMP
   64000 32000 2048 conform-action transmit exceed-action drop
 rate-limit input access-group name R-OTHER
   2048 2048 2048 conform-action transmit exceed-action drop
```

BGP Blackholing
Wewnątrz – jako mechanizm ochrony



BGP Blackholing
Wewnątrz – jako mechanizm monitoringu



Co można monitorować?

- **Skany ‘ciemnej’ przestrzeni adresowej IP (nieprzydzielone i przydzielone ale nieużywane)**

  Kto i dlaczego skanuje sieć—zwykle robaki...

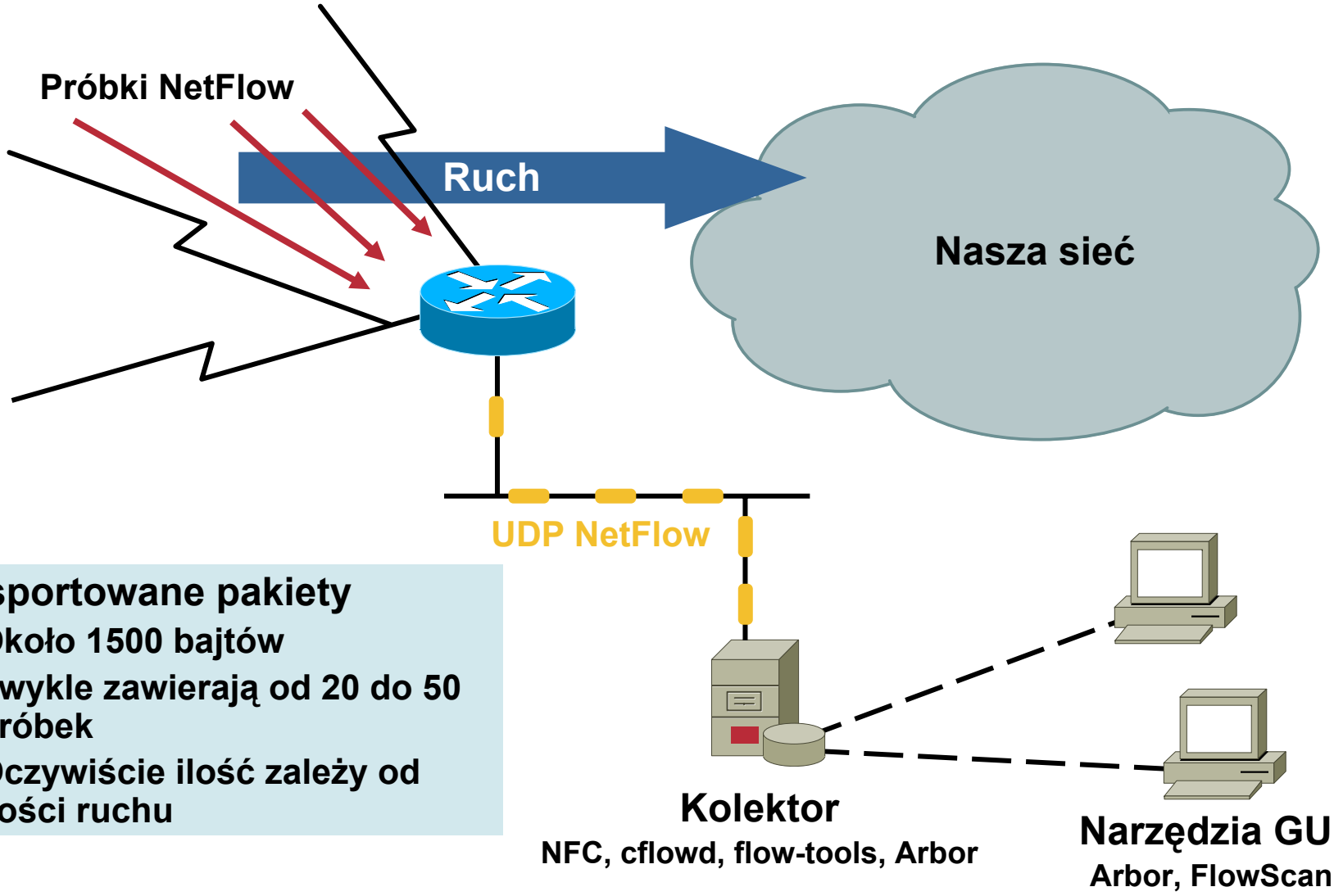
- **Skany przestrzeni nieużywanej lokalnie**

  Robaki, zainfekowne maszyny, prace badawcze...

- **Praca honeypot-alike**

  Jak dokładnie wygląda atak, z czego się składa, czy widzimy ruch C&C do botów?

Monitoring - NetFlow



Eksportowane pakiety

- Około 1500 bajtów
- Zwykle zawierają od 20 do 50 próbek
- Oczywiście ilość zależy od ilości ruchu

Co znajduje się w próbce NetFlow?
Przykład pakietu w wersji 5

- Ilość pakietów
- Ilość bajtów

- Źródłowy adres IP
- Docelowy adres IP

- Start sysUpTime
- End sysUpTime

- Źródłowy port TCP/UDP
- Docelowy port TCP/UDP

- Wejściowy ifIndex
- Wyjściowy ifIndex

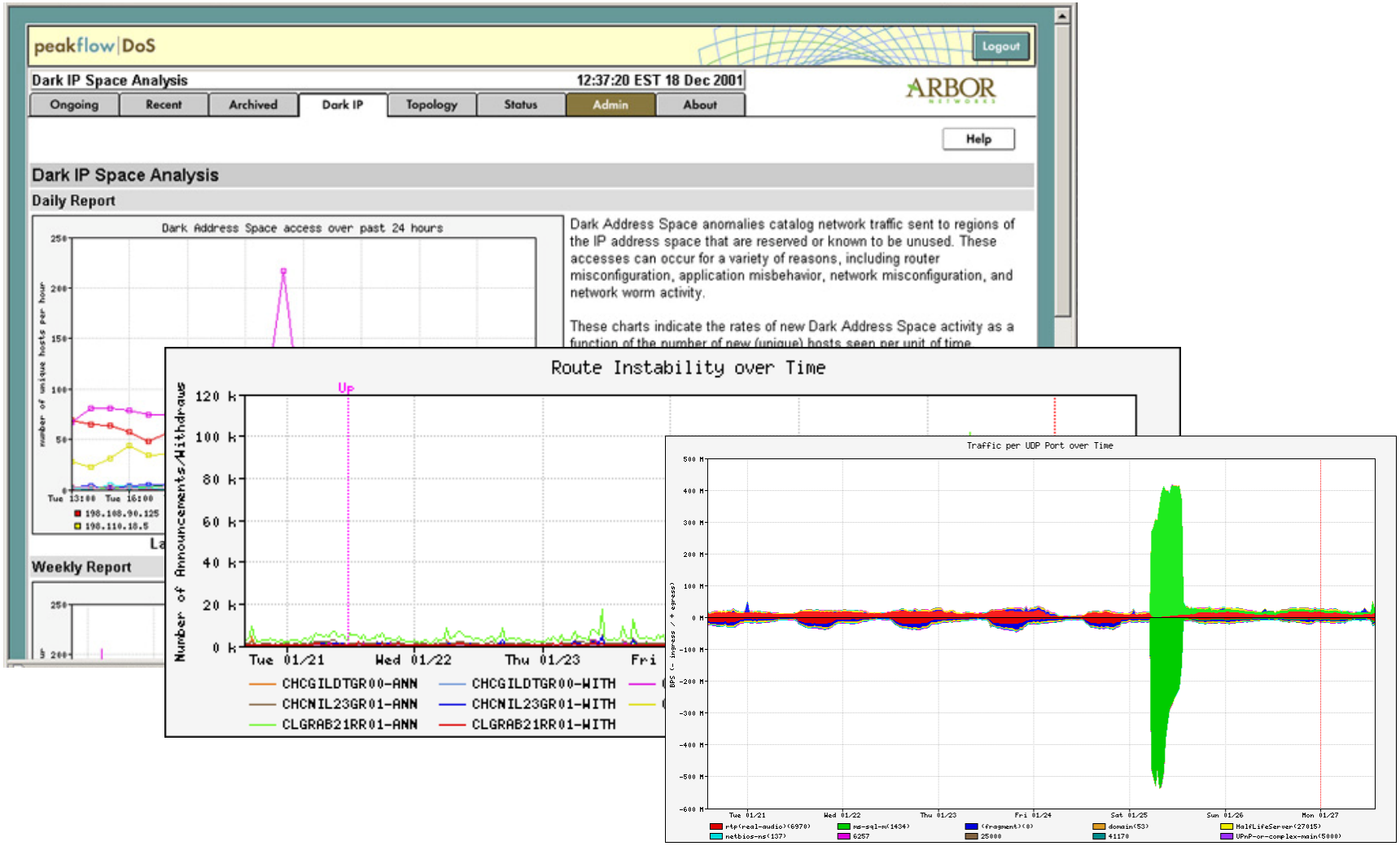
- Adres next-hop
- Źródłowy numer AS
- Docelowy numer AS
- Maska prefiksu źródłowego
- Maska prefiksu docelowego

- Pole ToS
- Flagi TCP
- Protokół

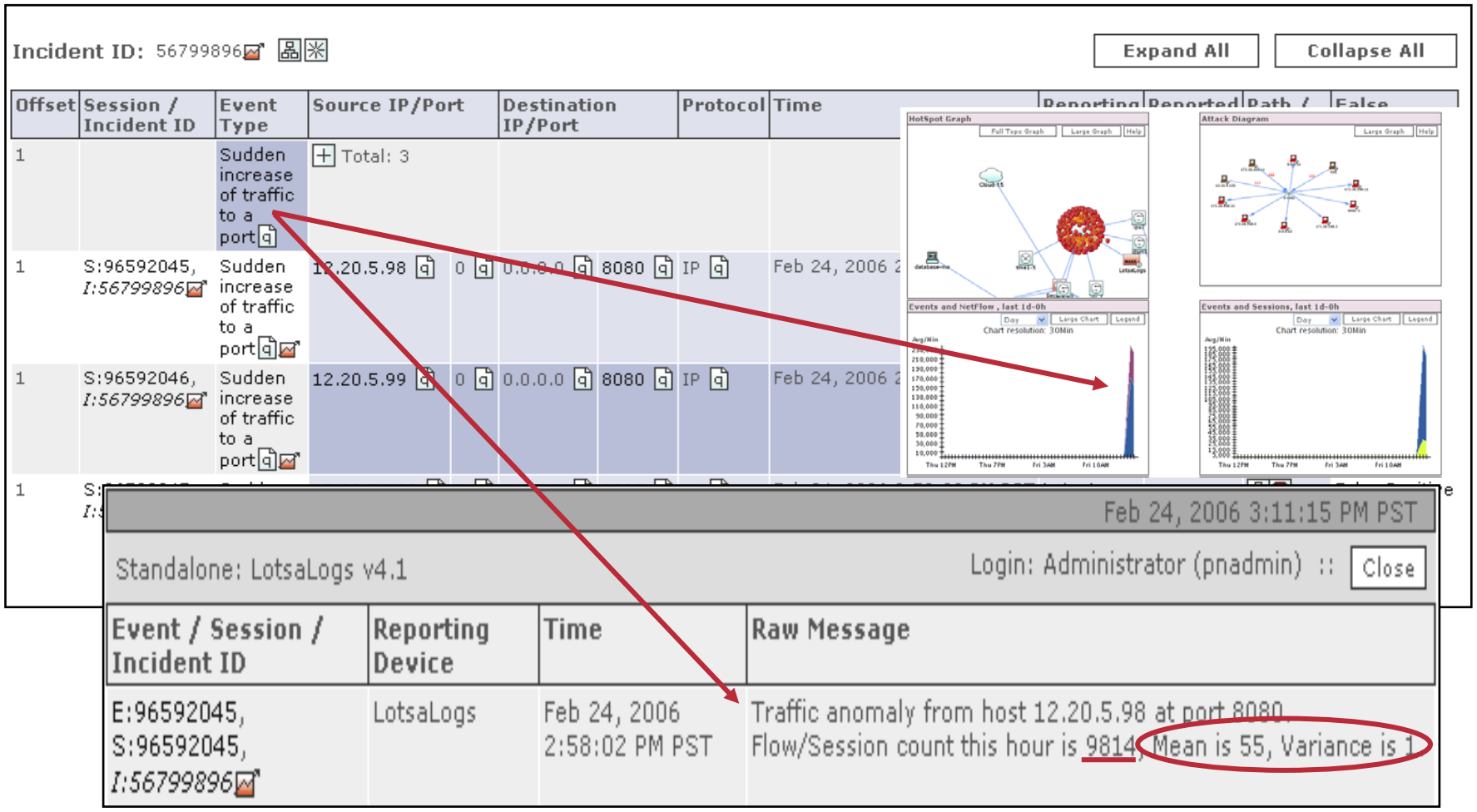
Zastosowania dla protokołu NetFlow

Dla operatora	Duże firmy
• Ustalenia dotyczące peeringu • Raportowanie SLA dla użytkowników VPN/MPLS • Billing na podstawie generowanego ruchu • Wykrywanie zagrożeń – ataków DDoS/robaków • Inżynieria ruchu • Rozwiązywanie problemów	• Monitoring ilościowy i jakościowy ruchu do Internetu • Podział kosztów dostępu pomiędzy różne działy przez IT • Zdecydowanie bardziej skalowalny niż RMON/SNMP • Wykrywanie zagrożeń – ataków DDoS/robaków • Rozwiązywanie problemów

Monitoring - NetFlow

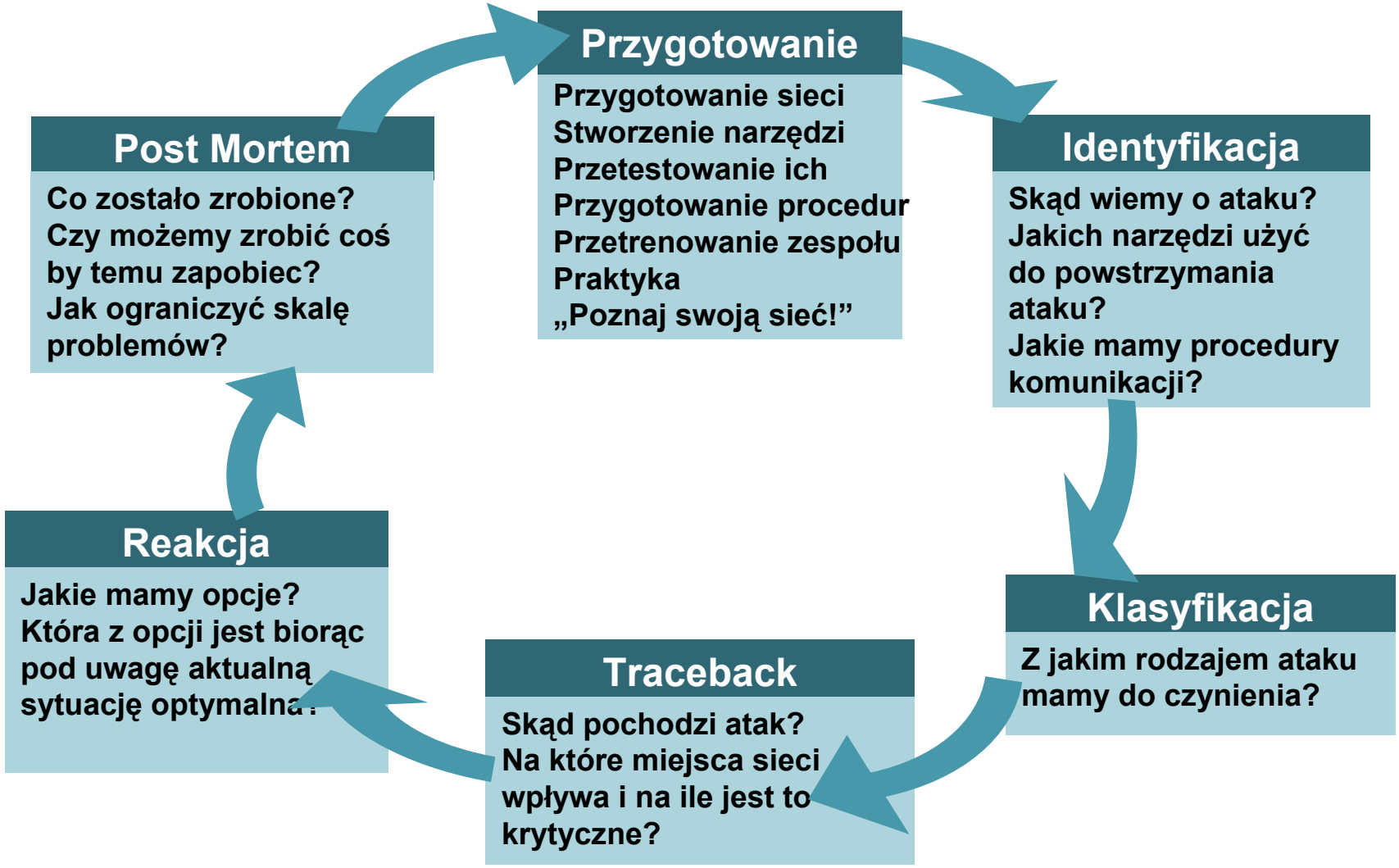


Monitoring – NetFlow w CS-MARS



Parę luźnych uwag

Procedury, procedury, procedury...



Q&A