

Směrovací démon BIRD

CZ.NIC z. s. p. o.

Ondřej Filip / ondrej.filip@nic.cz

Nov 6, 2009 – LinuxAlt / Brno



Test znalostí váženého publika

- Směrování – routing?
- IPv6?
- OSPF, BGP, AS?

Program

- Teoretický úvod
 - IP adresa, síťová maska směrovač
 - Směrování - externí, interní
 - Směrovací démon
 - Propojovací uzly
- BIRD
 - Historie
 - Vlastnosti, konfigurace, filtrování
 - BIRD vs Quagga vs OpenBGPD
 - Aplikace BIRDa - route server – NIX.CZ, LoNAP₃
 - Budoucí vývoj

IP protokol, IP adresa

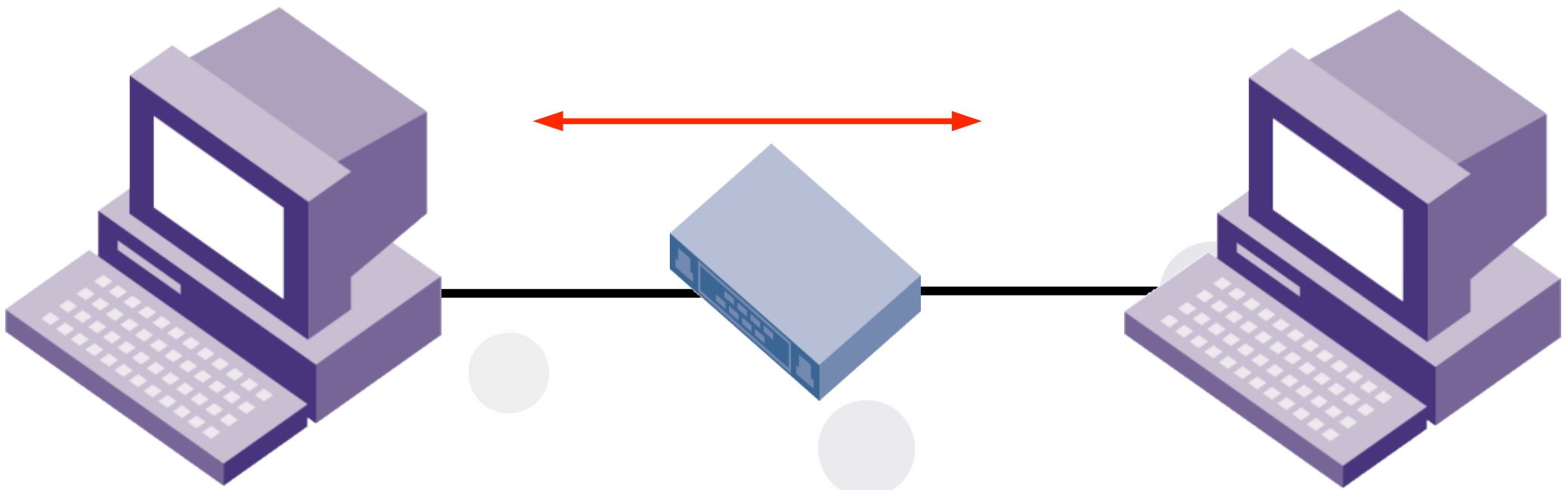
- Počítače nerozumí jménům
- Počítače rozumí číslům
- Každý počítač v Internetu má (nejméně jedno) číslo
- Jmenuje se IP adresa – 4 byte (nebo 16 byte)
- Kdysi byla unikátní... :-)
- 193.165.250.33
- 2001:1488:0:3::2

Network, netmask

- Počítač by si měl pamatovat k IP adrese i síťovou masku (anebo mu to někdo řekne)
- Síťová maska jsou jedničky a pak nuly (ve dvojkové soustavě)
- 255.255.255.0 (někdy taky /24)
- Logický bitový součin masky a adresu je číslo sítě
- Počítače se stejným číslem sítě by měli být připojeni ke stejné síti!

Komunikace na stejné lokální síti

- P1: 192.168.1.1 mask: 255.255.255.0
- P1: 192.168.1.2 mask: 255.255.255.0

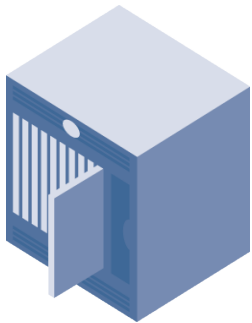


Gateway

- Co když číslo sítě neseďí?
- Měl by znát (default) bránu – cestu k někomu chytřejšímu
- Chytřejší se nazývá router a.k.a. směrovač
- Router je podfuk, ať žije počítač!!!!!!!!!!!!!!

Router

- Zařízení připojené k více sítím
- Má více IP adres
- Umí přeposlat „cizí“ zprávu - forwarding
- Cestu pozná podle směrovací (routovací) tabulky



Routery

- Malý a levný



Routery

- Velký a drahý



Routery

- Běžné PC s Unixem
- V Linuxu stačí jen zapnout routing a mít 2 rozhraní
- Sysctl – distribuce



Směrovací tabulka

Možný příklad:

217.31.201.0/24 dev eth0

217.31.202.0/24 dev eth1

217.31.203.0/24 via 217.31.201.1 dev eth0

default via 217.31.202.1 dev eth1

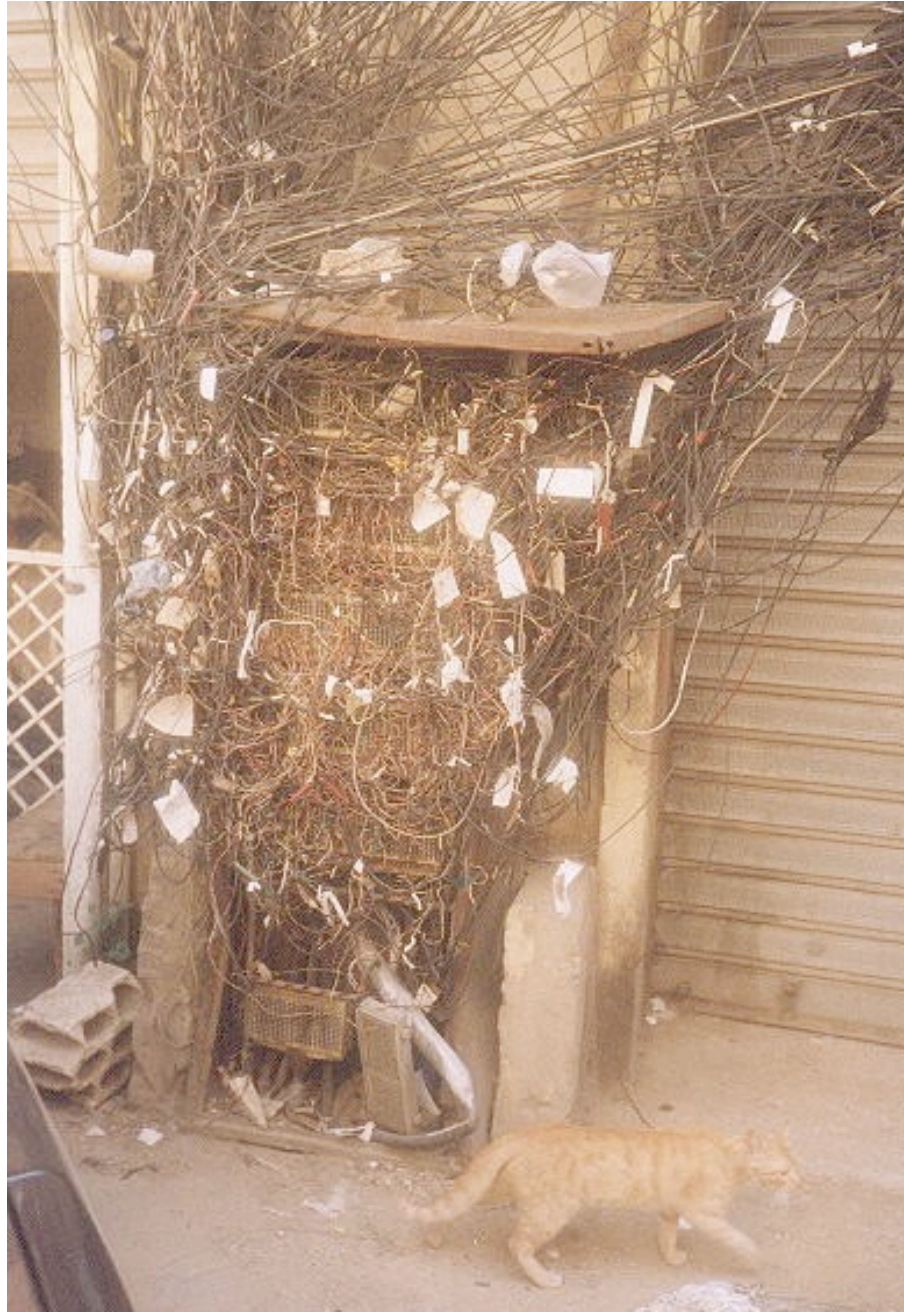
- Prioritu mají specifictější cesty

Konfigurace?

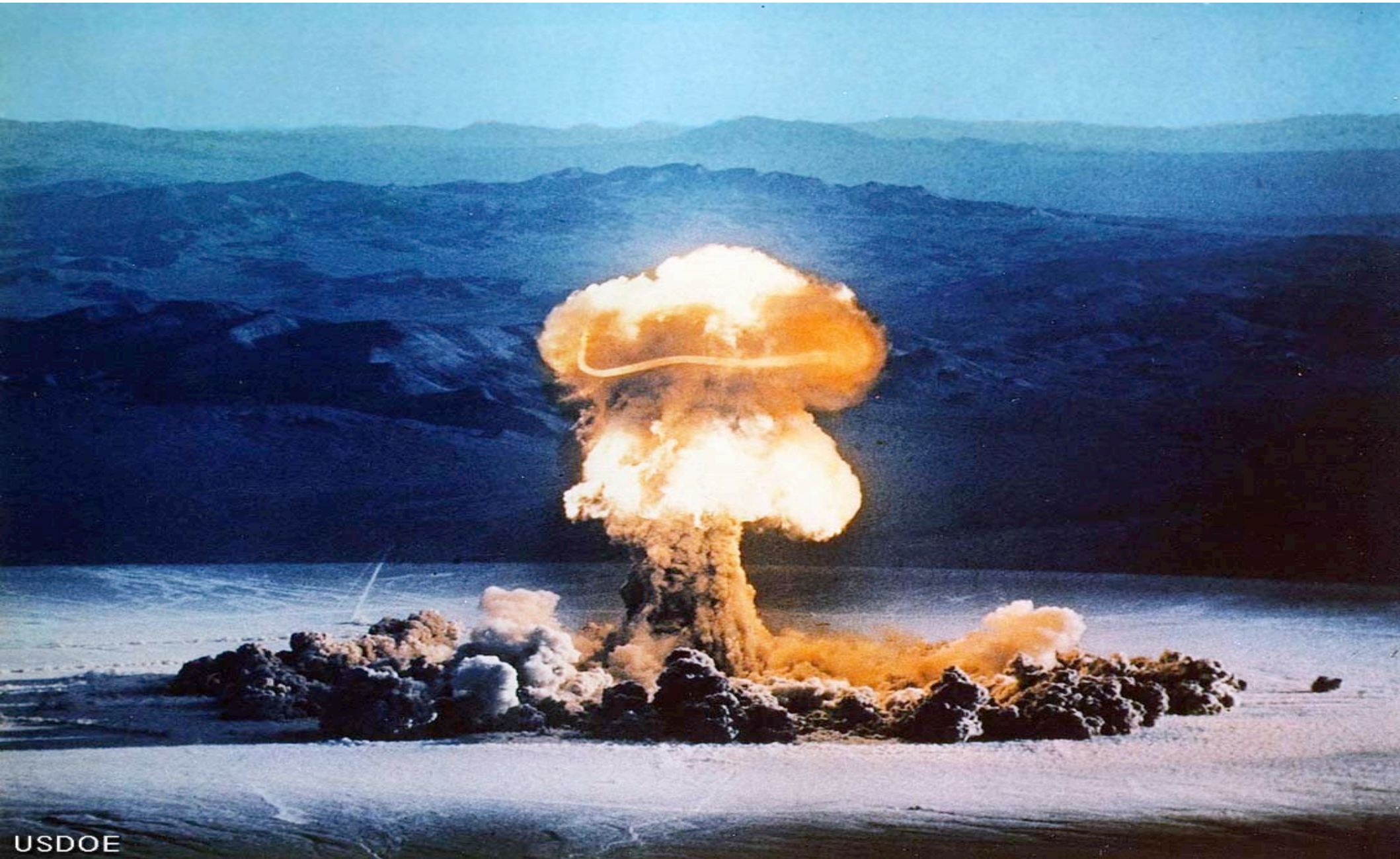
- Jak se router dozví topologii sítě?
- Jak sestaví směrovací tabulku?
- STATICKY (Ručně)

jenže.....

Někdy je síť složitější



Internet musí přežít!!!



Dynamické směrování

- Nic nového, jen se automaticky sestavuje směrovací tabulka
- Směrovače si „povídají“ o topologii (směrovací protokoly)
- Až na něco přijdou..... (a sestaví směrovací tabulku)
- Nikdy neví všechno! (někdo má globální pohled bez detailů, někdo zná detailně místní situaci)

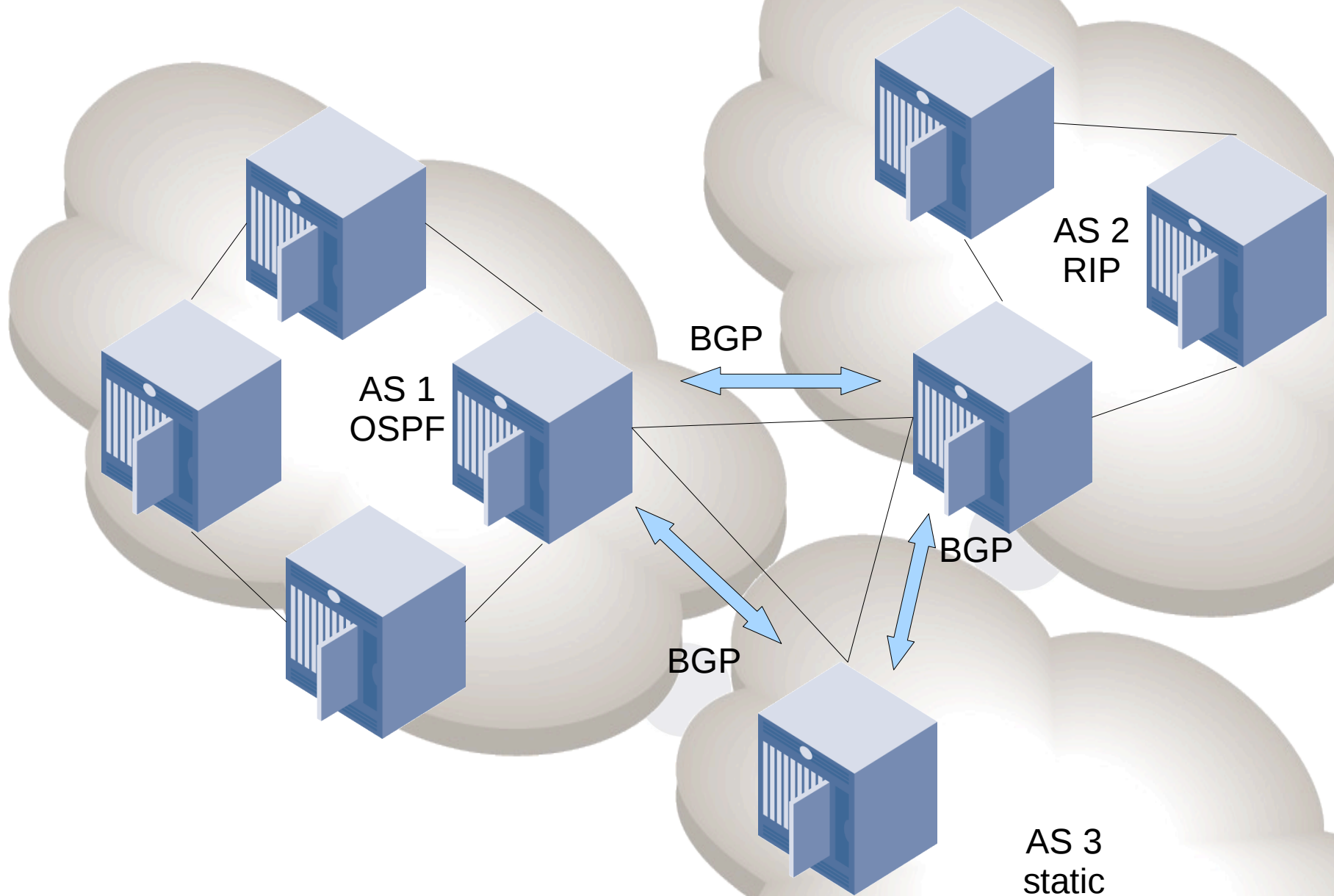
Rozdělení směrovacích protokolů

- Rychlé a pomalé, jednoduché a složité, ale hlavně kamarádské a nepřátelské neboli:

Interní a Externí

- Máme autonomní systémy (a jejich čísla)
- Množina směrovačů (kamarádů) pod jednotnou správou
- Na hranici jsou „velké“ směrovače – tlumočníci, novináři

Rozdělení směrovacích protokolů



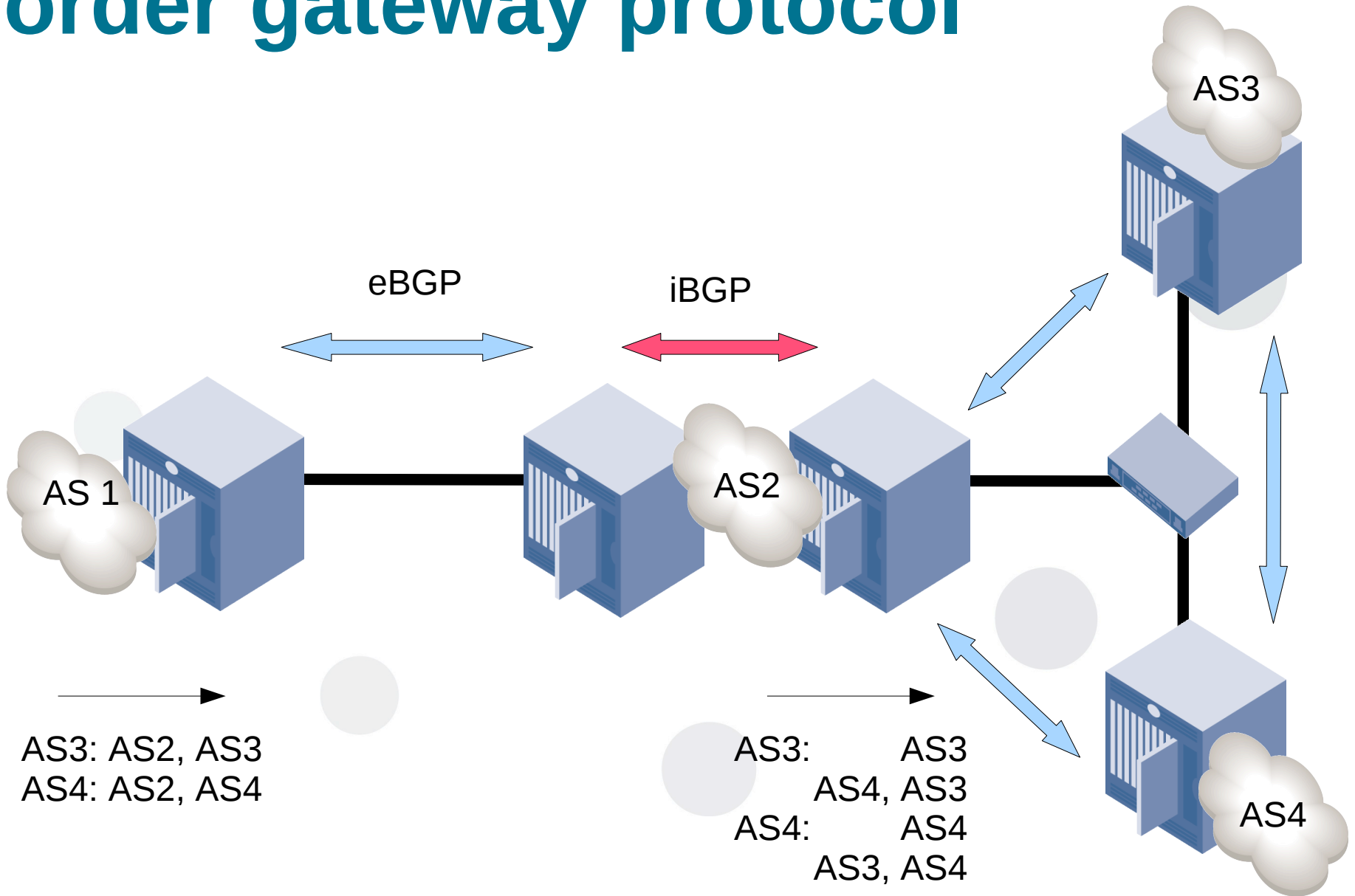
Border gateway protocol

- Jediný zástupce EGP
- BGP routery vidí „rozmazaně“ vidí celý svět
- Jsou nedůvěřivé (eBGP)
- I důvěřivější (iBGP)
- Často nemají default routu
- Ale mívají desetitisíce i statisíce položek v RT
- Posílá informace o sítích, které jsou přes něj dostupné (nejkratší cestou) a přidá své číslo AS (u eBGP)
- Preferuje se routa s nejmenším počtem AS v cestě

Border gateway protocol

- Každá routa (prefix) tedy obsahuje „cestu“ - seznam AS, přes které je nutno projít
- Preferuje se routa s nejmenším počtem AS v cestě
- Když předává informaci o síti (routa), může k přidat celou řadu dalších informací - atributy

Border gateway protocol



Interior Gateway Protocols

- RIP, RIPv2, RIPv6 – ne příliš používané
- OSPFv2, OSPFv3 – nejběžnější IGP
- IS-IS (proprietární Cisco)
- Statický
- Rychlé, důvěřivé (krom statického), obvykle menší tabulka, default route

Směrovací démon

- Na Linuxu (a ostatních UNIXech) – uživatelská aplikace mimo jádro, forwarding v jádře
- Obvykle implementuje více směrovacích protokolů
- Směrovací politika - filtrování
- Quagga (Zebra) – Cisco syntax <http://www.quagga.net>
- OpenBGPD - <http://www.openbgpd.org>
- GateD – zastaralý, ne volná licence
- BIRD - <http://bird.network.cz>

Peeringové uzly - IXP

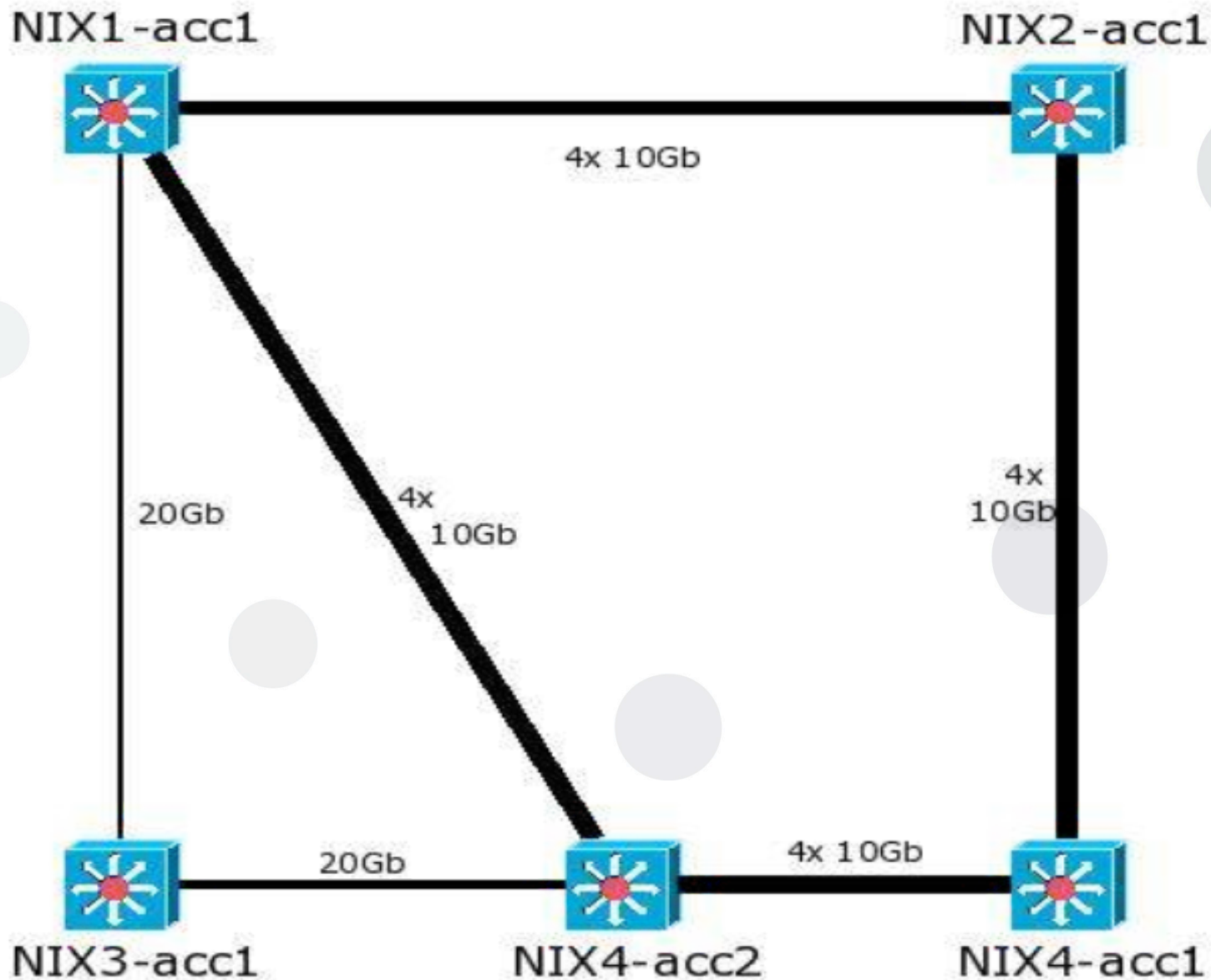
- Obvykle 1 ethernetový segment (jedna síť)
- Ale často více lokalit
- Vysoká redundance
- Všichni ISP připojeni (často vícekrát – do různých lokalit)
- Odpadá nutnost fyzického propojování se s každým
- Propojování protokolem BGP – každý s každým

Peeringový uzel NIX.CZ

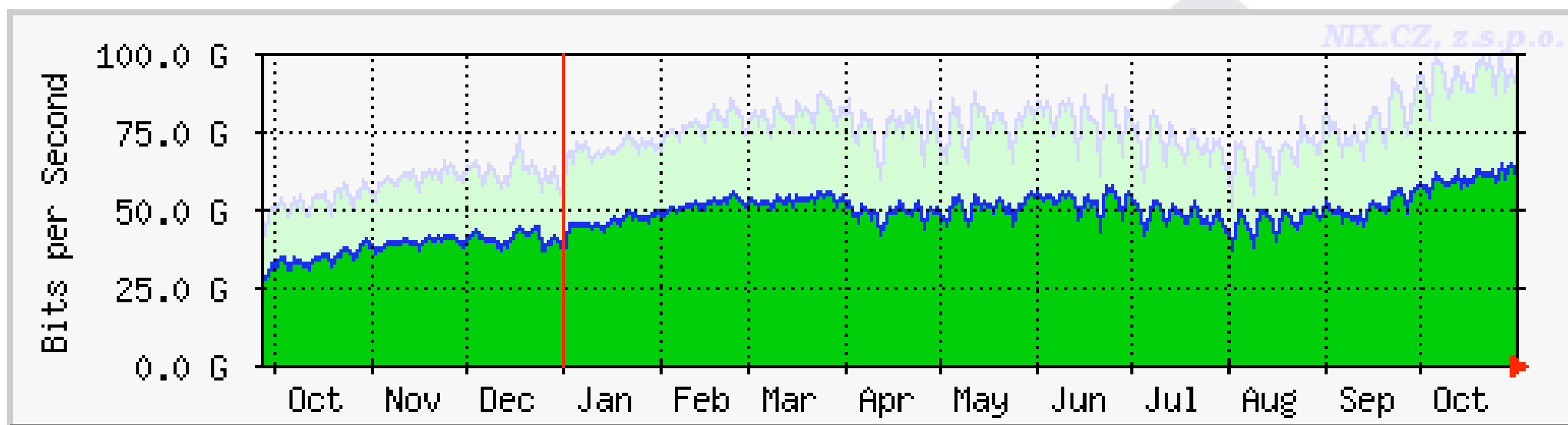
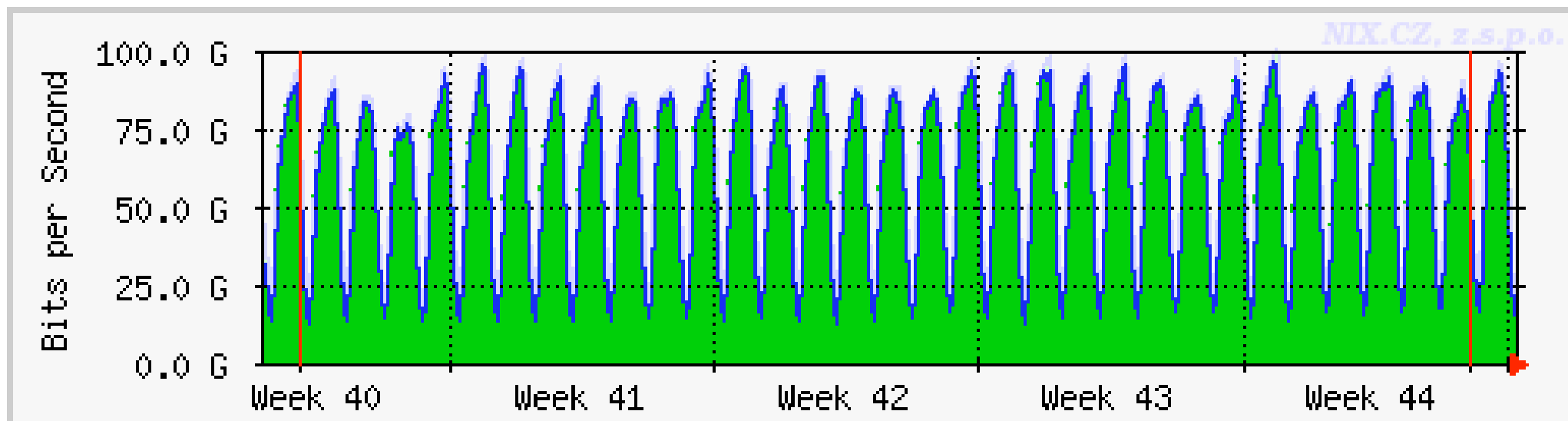


- 4 lokality v Praze, 5 přepínačů (switchů)
- Mezi největšími 10l v Evropě
- Špičkový tok cca 100 Gbps
- 100 připojených sítí
- Podpora IPv4 i IPv6
- Hlavní distribuční uzel pro kořenové DNS server L ICANNu pro Evropu
- <http://www.nix.cz>

Topologie NIX.CZ



Toky v NIX.CZ



Program

- Teoretický úvod
 - IP adresa, síťová maska směrovač
 - Směrování - externí, interní
 - Směrovací démon
 - Propojovací uzly
- BIRD
 - Historie
 - Vlastnosti, konfigurace, filtrování
 - BIRD vs Quagga vs OpenBGPD
 - Aplikace BIRDa - route server – NIX.CZ, LoNAP₂₉
 - Budoucí vývoj

Historie projektu

- Start projektu v roce 1999
- Seminární projekt – MFF UK Praha
- Projekt uspán
- Drobné probuzení v letech 2003 a 2006 (CESNET)
- Plně obnoveno na konci 2008 v rámci Laboratoří CZ.NIC - <http://labs.nic.cz>



Cíle projektu

- Opensource směrovací démon – alternativa k tehdejšímu démonu Quagga/Zebra (GateD)
- Rychlý a efektivní
- Portabilní, modulární
- Podpora současných směrovacích protokolů
- IPv6 a IPv4 v jednom zdrojovém kódu
- Snadná konfigurace a filtrování



Vlastnosti

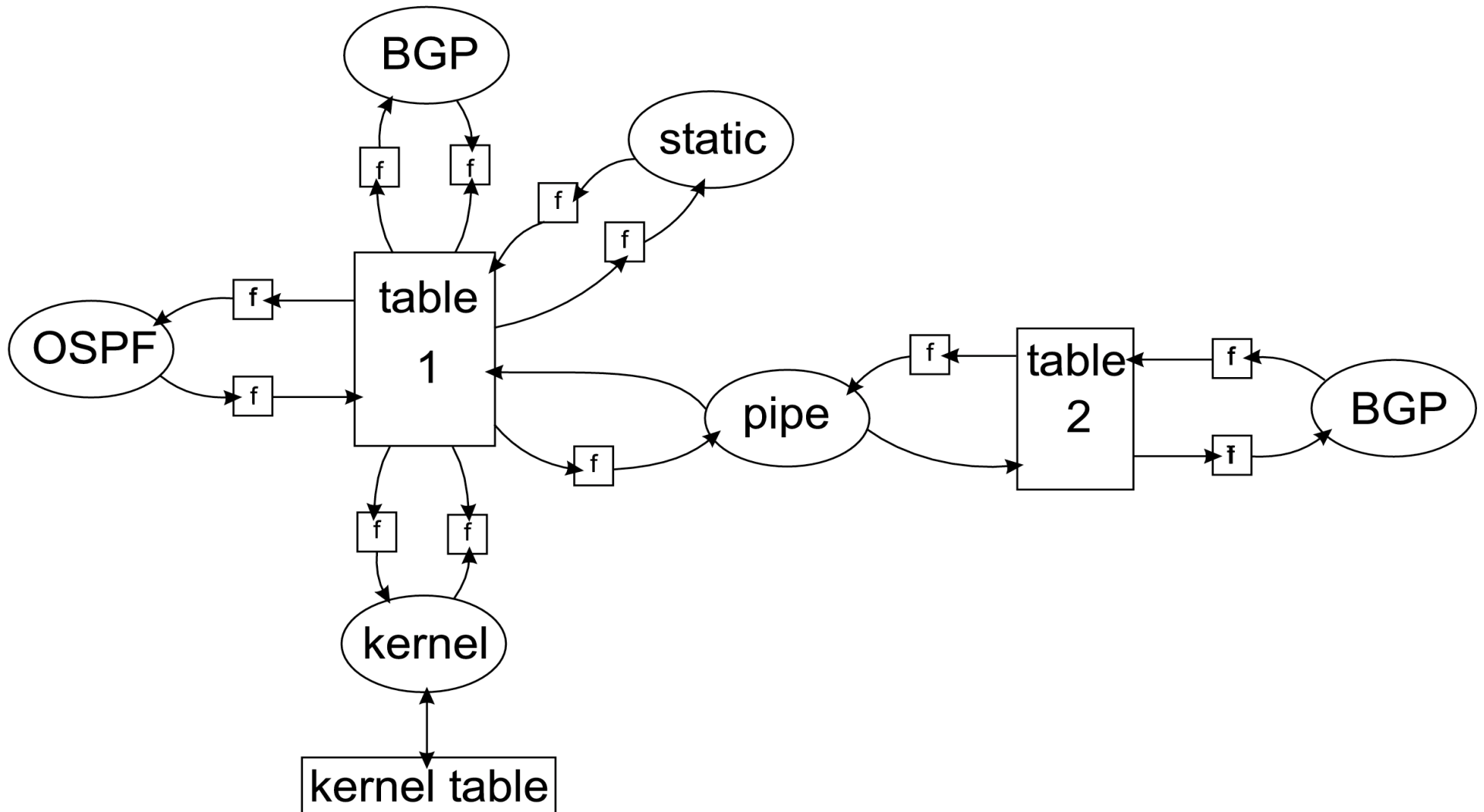
- Portabilní – Linux, FreeBSD, NetBSD, OpenBSD
- Podpora IPv4 i IPv6
- Statické směrování
- RIP, RIPv2, RIPng
- OSPFv2, (OSPFv3 je v alfa verzi, plnou verzi očekáváme do konce roku 2009)
- BGP (v4 a v6), route reflektor
- Směrovací server (Route server)
- ASN32 (ASPLAIN), MD5



Vlastnosti

- Více směrovacích tabulek (RIBs) (interně, ale také synchronizace s OS, pokud to podporuje)
- Protokol PIPE (!)
- Více směrovačů, route reflektorů a serverů na jednom systému
- Efektivní, silná konfigurace
- Silný jazyk pro filtrování
- Příkazová řádka (show, restart, ...)
- Automatická rekonfigurace

Design



Příklad konfigurace

```
log "/var/log/bird.log" all;

router id 193.51.100.238;

protocol static {
    route 10.0.0.0/8 drop;
    route 172.16.0.0/12 drop;
    route 192.168.0.0/16 drop;
}
filter bgp_out {
    if (net = 192.175.48.0/24 ) &&
        (source = RTS_DEVICE) then accept;
    else reject;
}
protocol bgp NIX_1 {
    local as 112;
    neighbor 193.51.100.235 as 6981;
    import all;
    export filter bgp_out;
}
```

Příkazová řádka



```
bird> show protocols
name      proto      table      state  since  info
direct1   Direct     master     up     Apr11
kernel1   Kernel     master     up     Apr11
device1   Device     master     up     Apr11
static1   Static     master     up     Apr11
NIX_2     BGP        master     up     Apr11  Established
NIX_1     BGP        master     up     Apr25  Established
ospf1     OSPF       master     up     Apr11  Running
bird>
bird> show status
BIRD 1.1.3
Current server time is 06-08-2009 22:01:06
Last reboot on 11-07-2009 22:54:12
Last reconfiguration on 30-07-2009 06:25:25
Daemon is up and running
bird>
```

Příkazová řádka



```
bird> show route
10.0.0.0/8      via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
127.0.0.0/8    dev lo [direct1 13:09] (240)
200.30.20.0/24 via 200.30.10.3 on eth2 [ospf1 13:10] I (150/10)
200.30.10.0/24 dev eth2 [direct1 13:09] (240)
                dev eth2 [ospf1 13:10] I (150/5)
200.0.10.0/24  dev eth0 [direct1 13:09] (240)
                dev eth0 [ospf1 13:09] I (150/5)
172.16.0.0/16  via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
195.47.235.0/24 via 194.50.100.246 on eth1 [NIX2 Apr11] (100)[AS688i]
                via 194.50.100.245 on eth1 [NIX1 Apr25] (100)[AS688i]

bird>
bird> show route protocol ospf1
10.0.0.0/8      via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
200.30.20.0/24  via 200.30.10.3 on eth2 [ospf1 13:10] I (150/10)
200.30.10.0/24  dev eth2 [ospf1 13:10] I (150/5)
200.0.10.0/24   dev eth0 [ospf1 13:09] I (150/5)
172.16.0.0/16   via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
```

Příkazová řádka



```
bird> show route for 127.0.0.1
127.0.0.0/8          dev lo [direct1 13:09] (240)
```

```
bird> show route filter bgp_out
192.175.48.0/24     dev dummy0 [direct1 Apr1] (240)
```

```
bird> show route count
1469 of 1469 routes for 849 networks
```

```
bird> show route export NIX_1
192.175.48.0/24     dev dummy0 [direct1 Apr1] (240)
```

```
bird> show route where 127.0.0.5 ~ net
0.0.0.0/0           via 195.47.235.1 on eth0 [static1 Apr1](200)
127.0.0.0/8        dev lo [direct1 Apr1] (240)
```

```
bird> show route filter {if 127.0.0.5 ~ net then accept;}
0.0.0.0/0           via 195.47.235.1 on eth0 [static1 Apr1](200)
127.0.0.0/8        dev lo [direct1 Apr1] (240)
```



Quagga

- Cisco styl konfigurace – výhoda i nevýhoda
- Více procesů – OSPF, BGP, zebra – nezávislé konfigurace
- Pořád ještě chybová – hlavně BGP
- Obtížné nasazení s automatickým generátorem konfigurace
- Slabší filtrování
- Málo efektivní kód
- Ne příliš aktivní vývoj

OpenBGPd



- Vlastní styl konfigurace
- Pouze protokol BGP (sesterský projekt OpenOSPFd)
- Efektivnější než Quagga
- Svázáno s BSD
- Nepodporuje více směrovacích tabulek

BIRD vs Quagga vs OpenBGPD

- Import plné IPv4 BGP tabulky ~300k položek
- Porovnání na Linuxu
- Více měření
- Spotřeba CPU (CPU time, sec) a paměti (MB)
- Test 1 – se synchronizací s směrovací tabulkou OS (Quagga - bgpd+zebra)
- Test 2 – bez synchronizace (i OpenBGPD)

BIRD vs Quagga vs OpenBGPD

- Test 1

Daemon	Memory (MB)	CPU (sec)
Zebra	90 + 77 = 167	32 + 120 = 152
BIRD	30	14

- Test 2

Daemon	Memory (MB)	CPU (sec)
Zebra	87	30
BIRD	30	7
OpenBGP	33 + 18 = 51	10 + 7 = 17

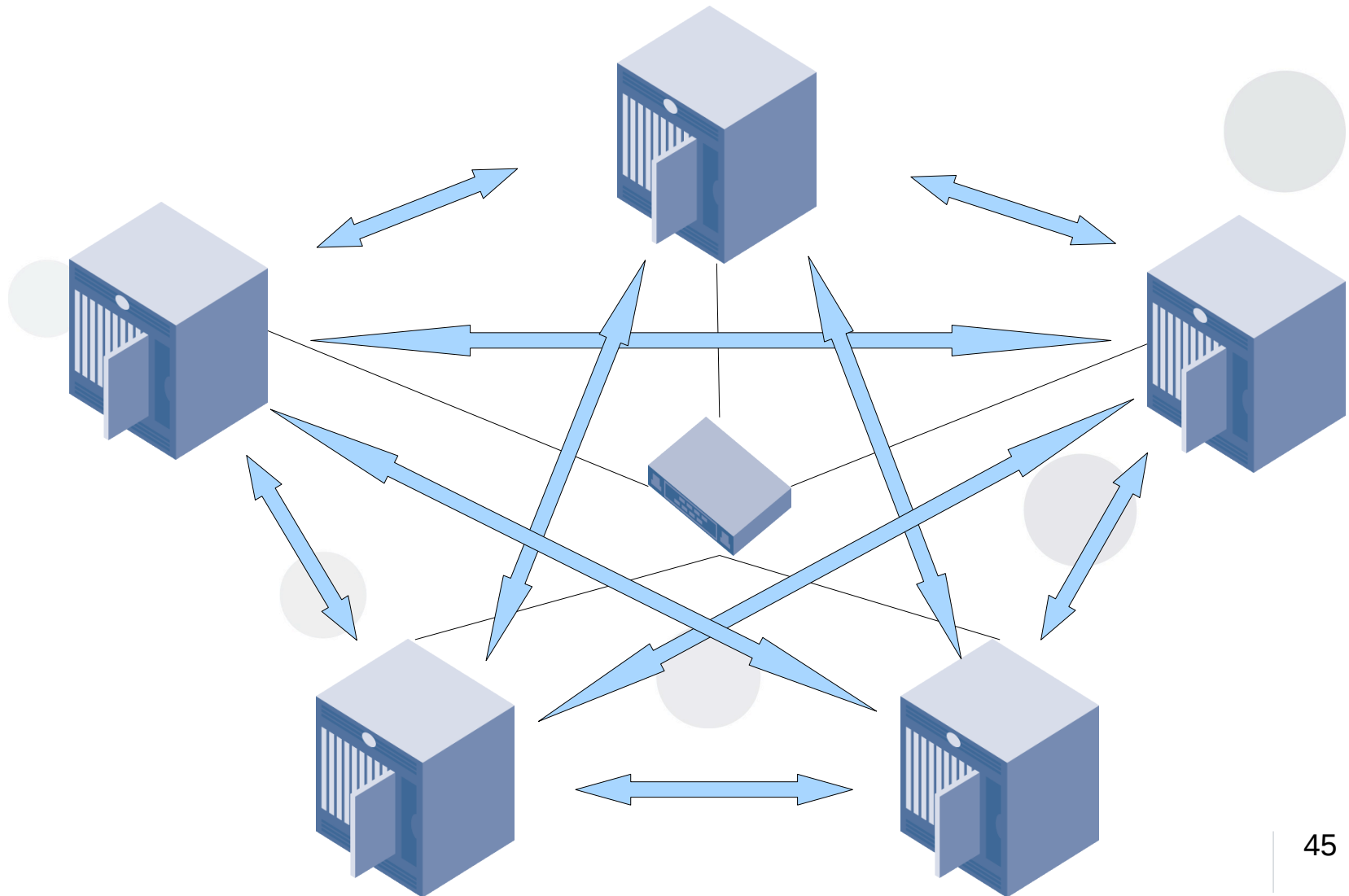
Příklad filtru

```
function avoid_martians()  
prefix set martians;  
{  
    martians = [ 169.254.0.0/16+, 172.16.0.0/12+,  
                192.168.0.0/16+, 10.0.0.0/8+, 224.0.0.0/4+,  
                240.0.0.0/4+, 0.0.0.0/32-, 0.0.0.0/0{25,32},  
                0.0.0.0/0{0,7}  ];  
  
    # Avoid RFC1918 networks  
    if net ~ martians then return false;  
  
    return true;  
}
```

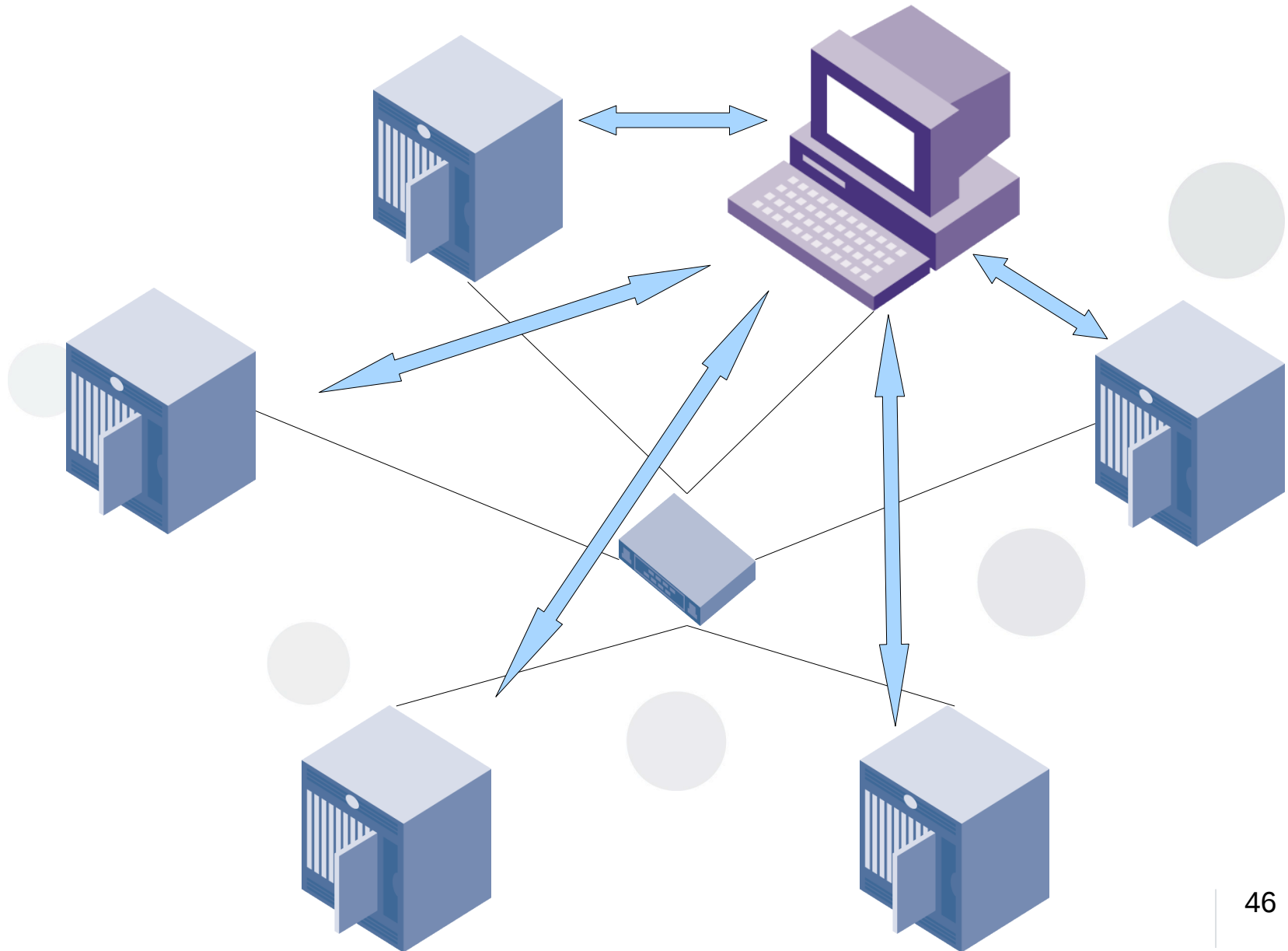
Router server

- Route server – snížení počtu BGP relací
- Každý člen IXP – $n-1$ BGP relací - celkem tedy $n*(n-1)/2$
- Nutnost konfigurace s každým novým členem
- Zátěž CPU router
- Řešením je route server – všichni jsou pouze připojeni k RS – 1 relace na router – n celkem
- RS musí skrýt svou existenci
- Ačkoliv jsou všichni propojeni k route serveru, nemusí všichni být propojení se všemi
- Kvalita RS, efektivita RS

IXP bez route serveru



IXP s route serverem



Příklad filtru – router server

- Politika route serveru - NIX.CZ

Pořadí	Komunita	Akce
1	0:<peer-as>	Nepropaguj routu <peer-as>
2	47200:<peer-as>	Propaguj routu <peer-as>
3	0:47200	Nepropaguj nikomu
4	47200:47200	Propaguj všem

Příklad filtru – router server

- Každý ISP zapojený do route serveru neztrácí svobodu definice své vlastní směrovací politiky
- Příklad 1 – chci posílat svou síť pouze CZ.NICu
 - 10.0.0.0/8 community 47000:25192, 0:47000
- Příklad 2 – chci posílat všem krom CZ.NICu
 - 10.0.0.0/8 community 0:25192, 47200:47200

Router server - Quagga

```
ip community-list standard C-0-10001 permit 0:10001
ip community-list standard C-0-47200 permit 0:47200
ip community-list standard C-47200-10001 permit 47200:10001
```

```
route-map Policy10001 deny 10
  match community C-0-10001
!
route-map Policy10001 permit 20
  match community C-47200-10001
!
route-map Policy10001 deny 30
  match community C-0-47200
!
route-map Policy10001 permit 40
!
```

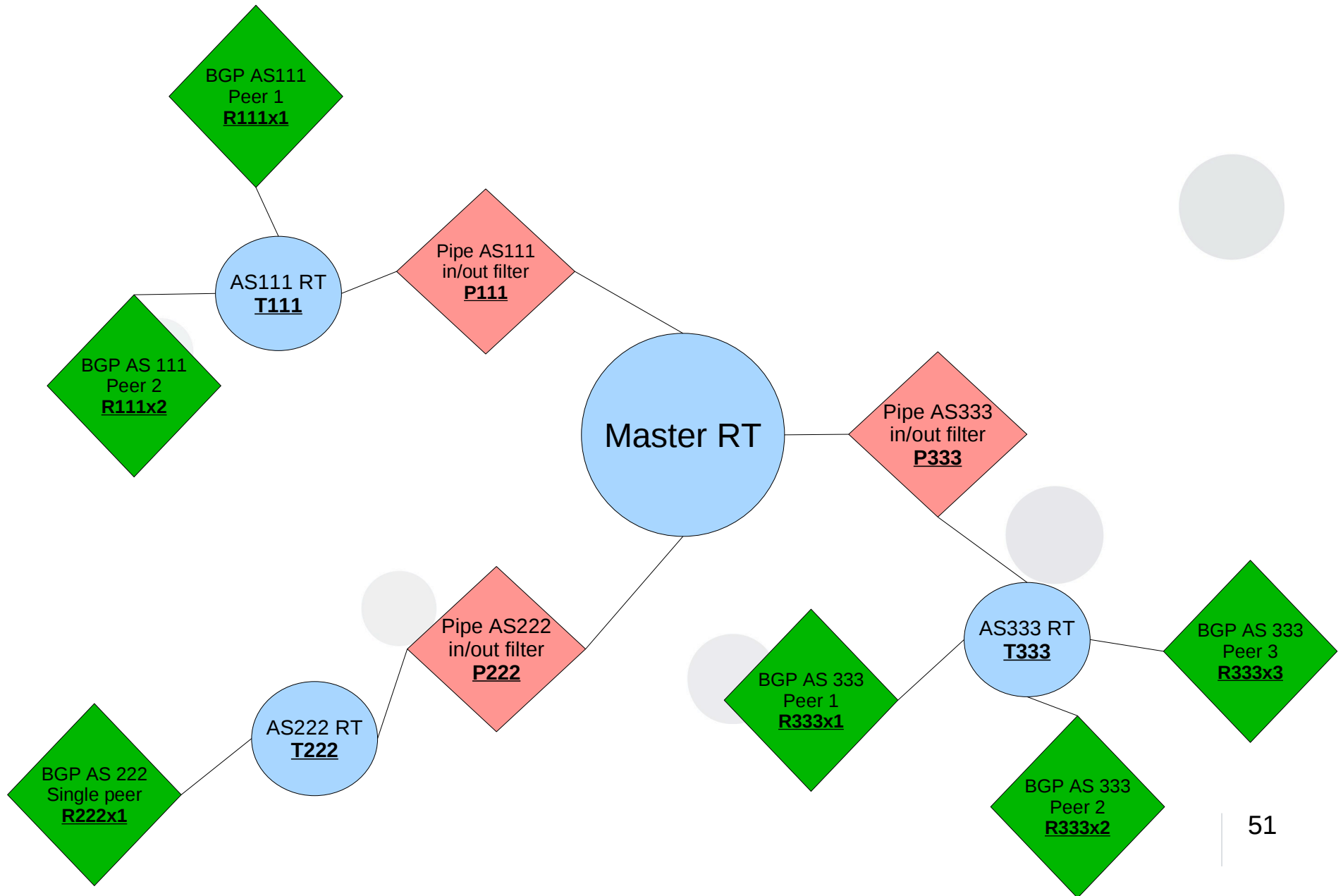
Příklad filtru – route server

```
define myas = 47200;
```

```
function bgp_out(int peeras)  
{  
    if ! (source = RTS_BGP ) then return false;  
    if (0,peeras) ~ bgp_community then return false;  
    if (myas,peeras) ~ bgp_community then return true;  
    if (0, myas) ~ bgp_community then return false;  
    return true;  
}
```

```
protocol bgp R25192x1 {  
    local as myas;  
    neighbor 194.50.100.13 as 25192;  
    import where bgp_in(25192);  
    export where bgp_out(25192);  
    rs client;  
}
```

BIRD as a route server



BIRD v LoNAPu

- IXP v Londýně
- První nasazení BIRDa jako route server
- Současný běh 2 route serverů
- BIRD (Linux) a OpenBGPD (FreeBSD)
- BIRD in má více RIBs (směrovacích tabulek)
- 25 BGP relací
- Cca 1000 IPv4 rout
- <http://www.lonap.net>

BIRD v NIX.CZ



- Také více směrovacích tabulek RIBs
- BIRD (Linux) a Quagga (FreeBSD)
- Žádný pád BIRDa od implementace (v. 1.1.3)
- Přibližně 100 IPv4 relací (30 na IPv6)
- Přibližně 6000 IPv4 rout (60 IPv6)
- Spotřeba paměti – 60MB (Quagga 260MB)
- Automatické generování konfigurace – 2x denně – rekonfigurace BIRDa
- Quagga – poloautomatická rekonfigurace



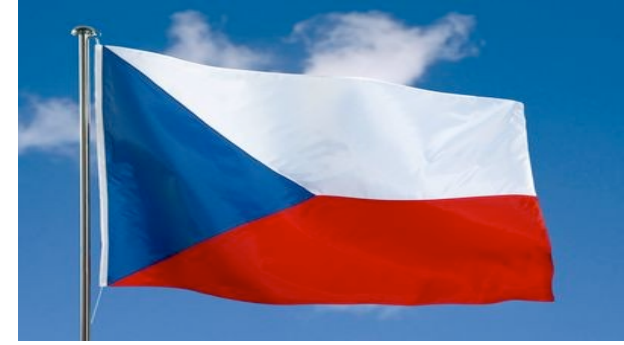
Jiné aplikace

- AS112 server v NIX.CZ
- BGP/OSPF router u menších ISP
- Router pro některé anycastové uzly CZ.NICu
- Používán v malých embedded systémech – součást firmware některých WiFi AP
- Nasazení v některých jiných IXP je v běhu – VIX do konce roku, testování MSK-IX, LINX
- Součást analyzátoru BGP tabulek

Budoucí vývoj

- Nová verze cca jednou měsíčně
- Distribuce (v současnosti deb, rpm)
- OSPFv3 – konec roku 2009
- Vylepšení OSPF (Opaque LSA, ...)
- Route flap dampening
- Další porty
- ...
- Záleží na přáních uživatelů

Závěr



- Český projekt – ale pro celosvětovou komunitu – kvalitní dokumentace v AJ
- Alternativa k dnes nejrozšířenějšímu démonu Quagga
- Nasazen v důležitých centrech internetu
- Jeden z OpenSource projektů Laboratoří CZ.NIC – příspěvek ke stabilitě a rozvoji Internetu

Závěr

Klady

- Velice silná konfigurace a filtrování
- Efektivní, úsporný
- Protokol PIPE
- Aktivní vývoj
- Rekonfigurace
- Modularita

Zápory

- Jiný styl konfigurace
- Chybí některé vlastnosti či další napojení – Looking Glass – www přístup k BGP informacím
- Separace IPv4 a IPv6

¿Dotazy?

Děkuji za pozornost!

<http://bird.network.cz>, <http://labs.nic.cz>

