

12. Въведение в маршрутизацията.

Маршрутни алгоритми.

Софтуер за маршрутизация
Quagga Routing Suite с отворен
код. Инструментариум iproute2.

Централизирана маршрутизация.
Софтуерно дефинирани мрежи.

Маршрутни алгоритми

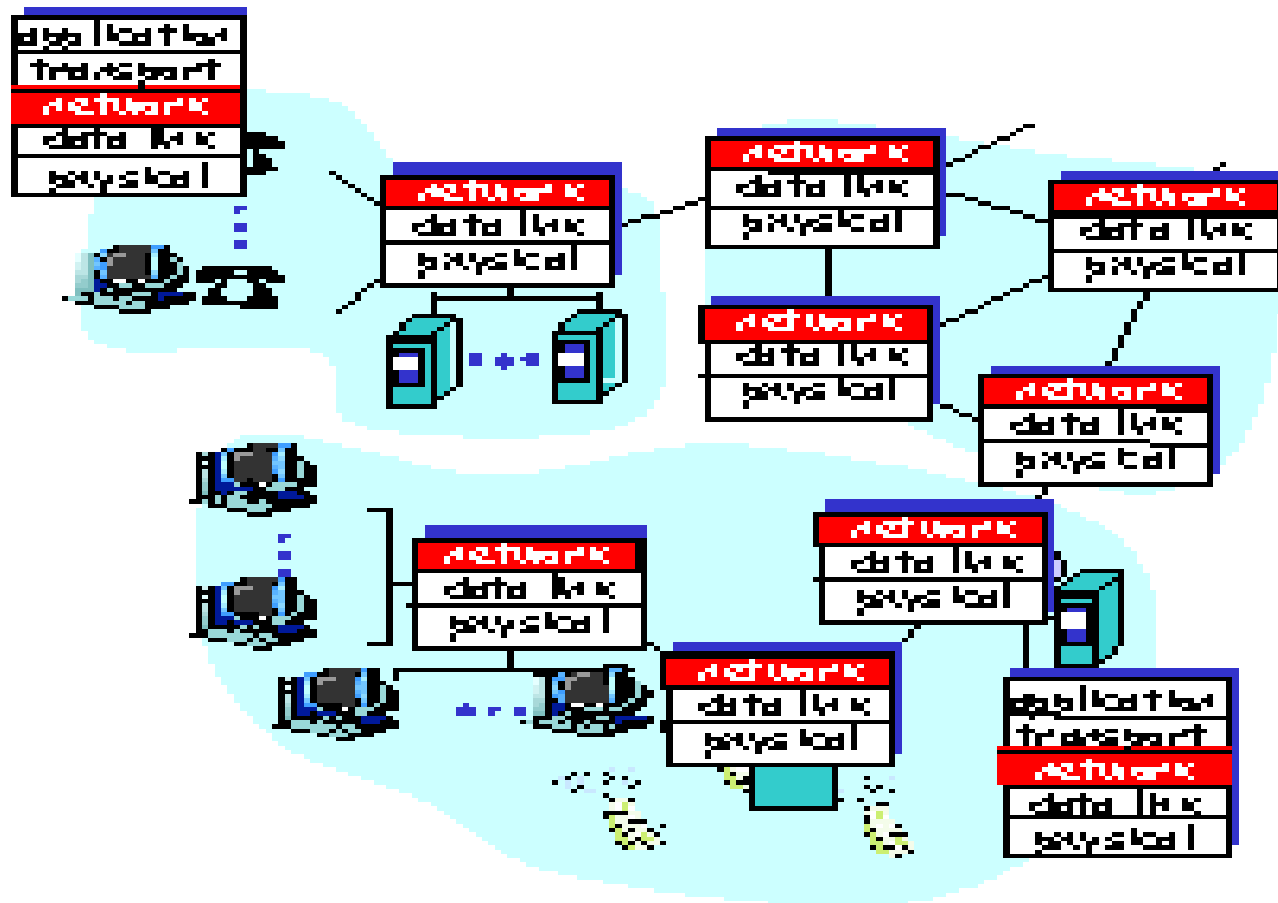
Основната функция на мрежовото ниво е да маршрутизира пакетите от източника към получателя - през няколко хопа.

Маршрутен алгоритъм е част от софтуера на мрежовото ниво, която определя по коя от изходните линии да се изпрати пристигнал пакет. За целта всеки маршрутизатор притежава **маршрутна таблица**.

Ако мрежата е с **пакетна комутация** (дейтаграми), решението трябва да се взема наново за всеки пристигнал пакет, тъй като оптималният маршрут може да се е променил.

Ако се използва **виртуален канал**, решенията по маршрутизацията се взимат при създаването му.

Функции на мрежовия слой



Маршрутизиращи протоколи

Маршрутизиращите протоколи трябва да отговарят на множество изисквания.

- Да са достатъчно прости и лесни за конфигуриране и да осигуряват надеждна и стабилна работа на мрежата.
- Да реагират своевременно на отпадане на маршрутизатори или връзки между тях.
- Да бъдат в състояние да открият алтернативни пътища за доставяне на пакетите, ако такива съществуват.

Маршрутизиращи протоколи

Две други цели на маршрутизиращите протоколи си противоречат (на пръв поглед):

- **минимизиране на времето за закъснение** (по-малък престой на пакетите в междинните възли);
- **максимизиране на общия поток** предполага буферите в маршрутизаторите да работят на максимален капацитет.

Освен това максимизирането на общия поток може да влезе в противоречие с изискването мрежовите ресурси да могат да се използват от всички потребители в мрежата.

Маршрутизиращи алгоритми

Маршрутизиращите алгоритми са два вида - **неадаптивни** и **адаптивни**.

При **неадаптивните** маршрутизацията не се извършва на базата на текущата топология на мрежата.

Маршрутите между всеки два възела в мрежата се изчисляват предварително и **се записват ръчно от мрежовите администратори**, след което влизат в маршрутните таблици.

При промяна на топологията на мрежата (например при отпадане на възел или на връзка), администраторите ръчно трябва да променят маршрутите.

Това прави неадаптивните алгоритми приложими само в малки мрежи, при които рядко настъпват промени.

Неадаптивните алгоритми се наричат още **статични**.

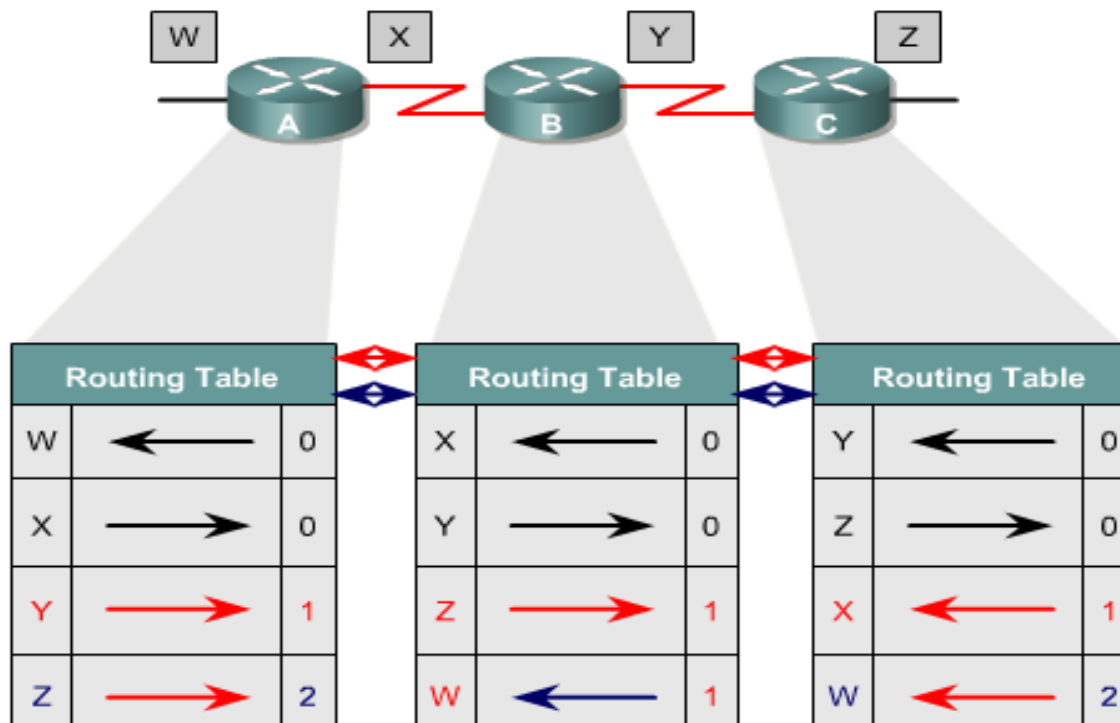
Попълване на маршрутна таблица. Пример.

Мрежата представяме като **граф** - върховете са възлите в мрежата, а дъгите са комуникационните линии.

Метриката в графа се определя на базата на разстоянието до крайната точка (**хопове**), време-закъснението за минаване на един пакет, надеждност на линията, цена, брой хопове и др. Възможно е да се комбинират една или повече от изброените характеристики.

Попълване на маршрутна таблица. Хопове.

Метриката се определя от това през колко маршрутизатора (хопа) ще премине пакета до крайната точка (мрежа).



Адаптивни алгоритми. Скорост на сходимость (конвергенция)

При **адаптивните алгоритми** маршрутните таблици се променят **динамично**, за да отразяват промени в топологията и натовареността на трафика.

Важна характеристика на адаптивния алгоритъм е неговата **скорост на сходимость**:

времето, което е необходимо да се **преизчислят маршрутните таблици** на всички маршрутизатори в мрежата при промяна в топологията или трафика (**конвергенция**).

Адаптивни алгоритми.

Принцип за оптималност

Оптималните пътища между всеки два възела в мрежата се изчисляват по някои от алгоритмите за намиране на **най-къс път в граф** (след като е въведена метрика в графа, представящ мрежата).

Всички тези алгоритми се базират на **принципа за оптималност**:

всяка част от оптимален път е също оптимален път между съответните два върха.

Като следствие от този принцип, **оптималните пътища от един връх към всички останали образуват дърво** (sink tree).

Алгоритъм на Дейкстра

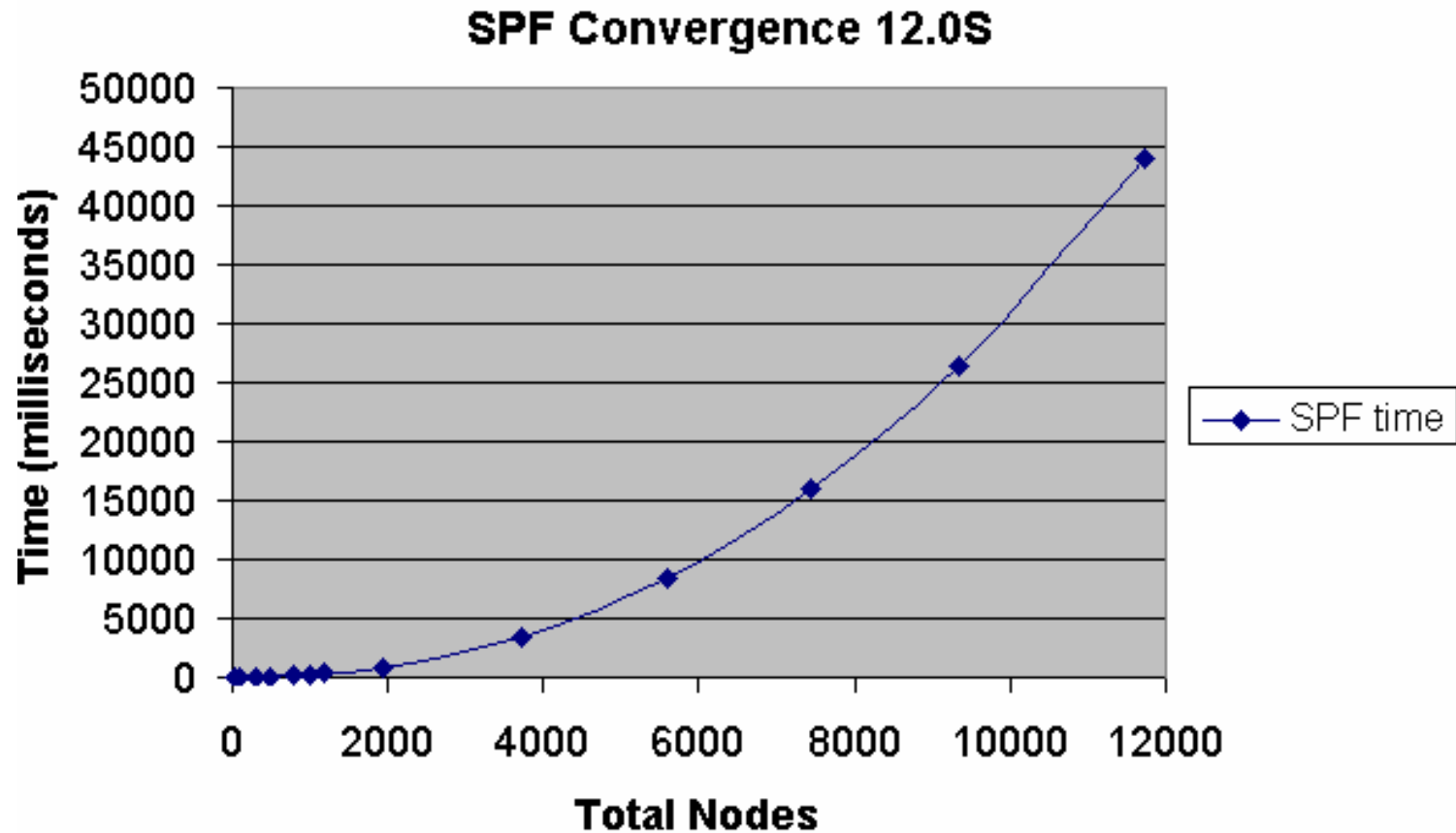
Алгоритъмът на Дейкстра е алгоритъм за намиране на най-къс път в граф от даден връх до всички останали върхове.

Важно е да се отбележи, че при алгоритъма на Дейкстра теглата на ребрата трябва да са положителни.

Резултатът от алгоритъма е дърво на оптималните пътища от дадения връх до всички останали.

Т.е. Shortest Path Tree (SPF).

Сходимость (конвергенция)



Вътрешни и външни протоколи

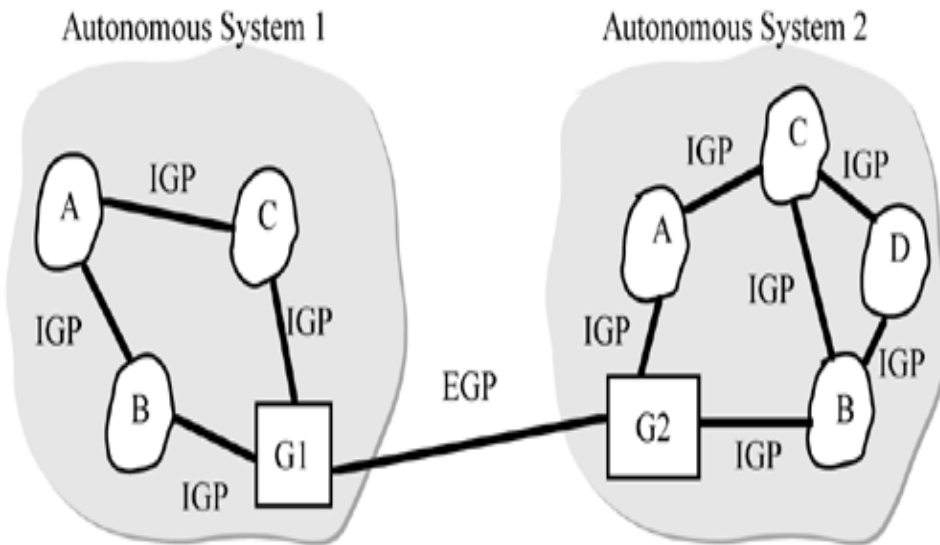
IGP vs EGP

IGP:

- **DV**: RIP, (E)IGRP
- **LS**: OSPF, IS-IS

EGP:

BGP 4/4+



Quagga



Quagga е **open source** софтуерен пакет за маршрутизация.

Поддържа: RIPv1, RIPv2, RIPvng, OSPFv2, OSPFv3, BGP-4 и BGP-4+ (т.е. **IPv4** и **IPv6**)

Quagga рутер

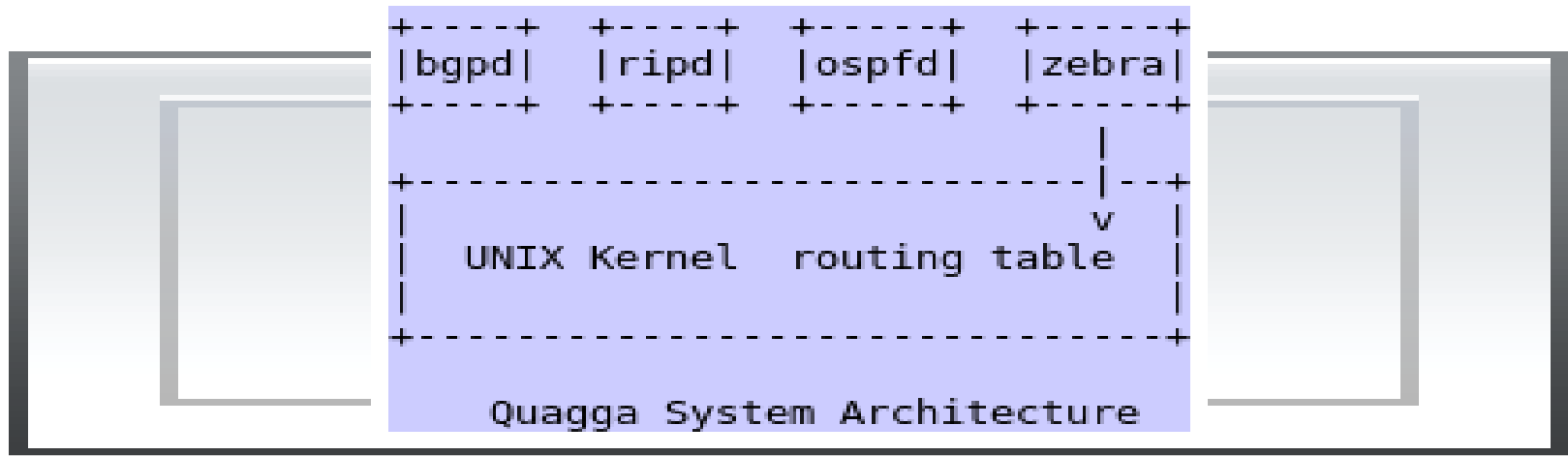


Компютър с Quagga си е рутер с **Cisco CLI**.

Обменя информация за маршрутите с помощта на маршрутни протоколи.

Quagga я използва, за да обновява таблицата с маршрутите в ядрото.

Quagga. Архитектура.



Мултипроцесна архитектура.

(все още няма **multi-thread**)

Всеки демон - `.conf` файл и терминал.

Статичен маршрут – `zebra.conf`

BGP - `bgpd.conf`

Ядро на Linux рутер

```
less /etc/sysctl.conf
```

```
...
```

```
# Controls IP packet forwarding
```

```
net.ipv4.ip_forward = 1
```

```
...
```

```
net.ipv6.route.max_size = 15000000
```

```
net.ipv4.route.max_size = 15000000
```

XORP



XORP рутер с интерфейс на **Juniper**.

Също е **open source** платформа за IPv4 и IPv6 маршрутизация.

Поддържа OSPF, RIP, BGP, OLSR, VRRP (Virtual Router Redundancy Protocol), PIM, IGMP (Multicast).

iproute2

Iproute2 – сбор от средства за контрол на TCP/IP мрежи и трафик в Linux.

Пример: добавя адрес 10.0.0.1 с префикс 24 (255.255.255.0) и стандартен broadcast към интерфейс eth0

```
[root@XXX]#ip addr add 10.0.0.1/24 brd + dev eth0
```

```
[root@XXX]#ifdown eth0
```

```
[root@XXX]#ifup eth0
```

iproute2

Показване на състоянието:

```
[root@XXX]# ip address show dev eth0
```

или

```
[root@XXX]# ip a [ls eth0]
```

Изтриване на адрес:

```
[root@XXX]# ip addr del 10.0.0.1/24 dev eth0
```

Централизирани адаптивни алгоритми

При централизираните адаптивни алгоритми в мрежата се създава един маршрутен управляващ център.

Той изчислява маршрутните таблици на всички възли и им ги изпраща.

За да се адаптират маршрутните таблици към текущата топология и текущия трафик, всички възли трябва да изпращат информация към маршрутния център.

На базата на получените сведения, маршрутният център изчислява теглата на ребрата и след това пресмята оптималния маршрут между всеки два възела.

Добре е да се поддържат алтернативни пътища между възлите.

Централизирани адаптивни алгоритми

Информацията от по-близките до маршрутния център възли ще пристигне по-бързо отколкото от по-далечните. Поради това **периодът на обновяване** на маршрутните таблици трябва да е поне два пъти по-голям от времето за преминаване на пакет от маршрутния център до най-отдалечения от него възел.

Преизчислена маршрутна таблица, получена в един възел, не трябва да се използва веднага, тъй като маршрутните таблици пристигат по различно време в различните възли.

Ако по някаква причина маршрутният център отпадне, мрежата остава без управление.

За целта може да се дублира маршрутният център, но тогава служебният трафик би се увеличил твърде много.

...ОТНОВО централизирана. SDN и OpenFlow

Системите преминаха:

- централизирани - големи машини (mainframes);
- разпределени – PCs;
- пак централизирани – VMs, Cloud.

Същото и с мрежите. Днес **Cloud Networking Services (CNS)**:

- SaaS и/или
- IaaS и/или NaaS.

Проектът OpenFlow

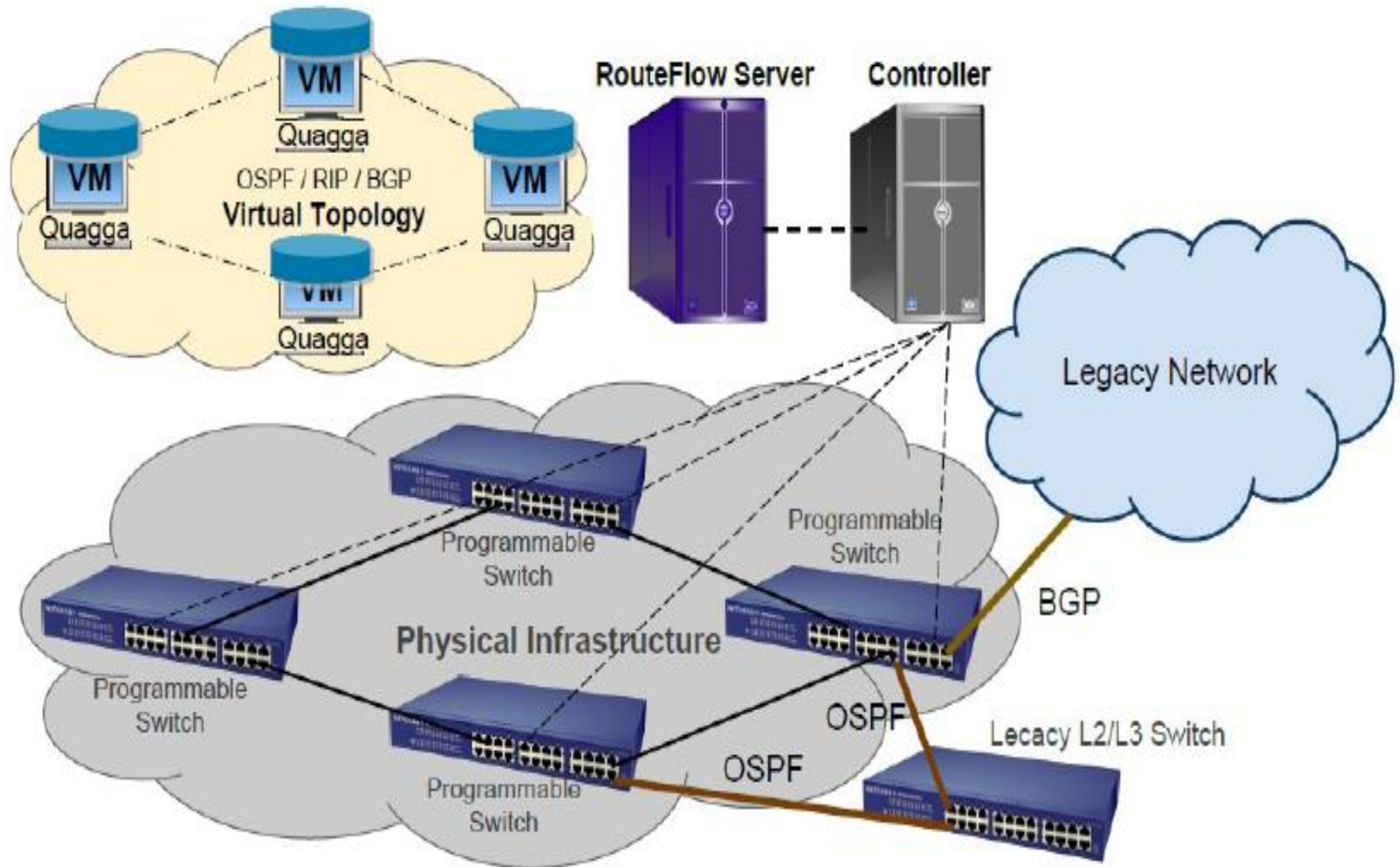
(www.openflow.org)



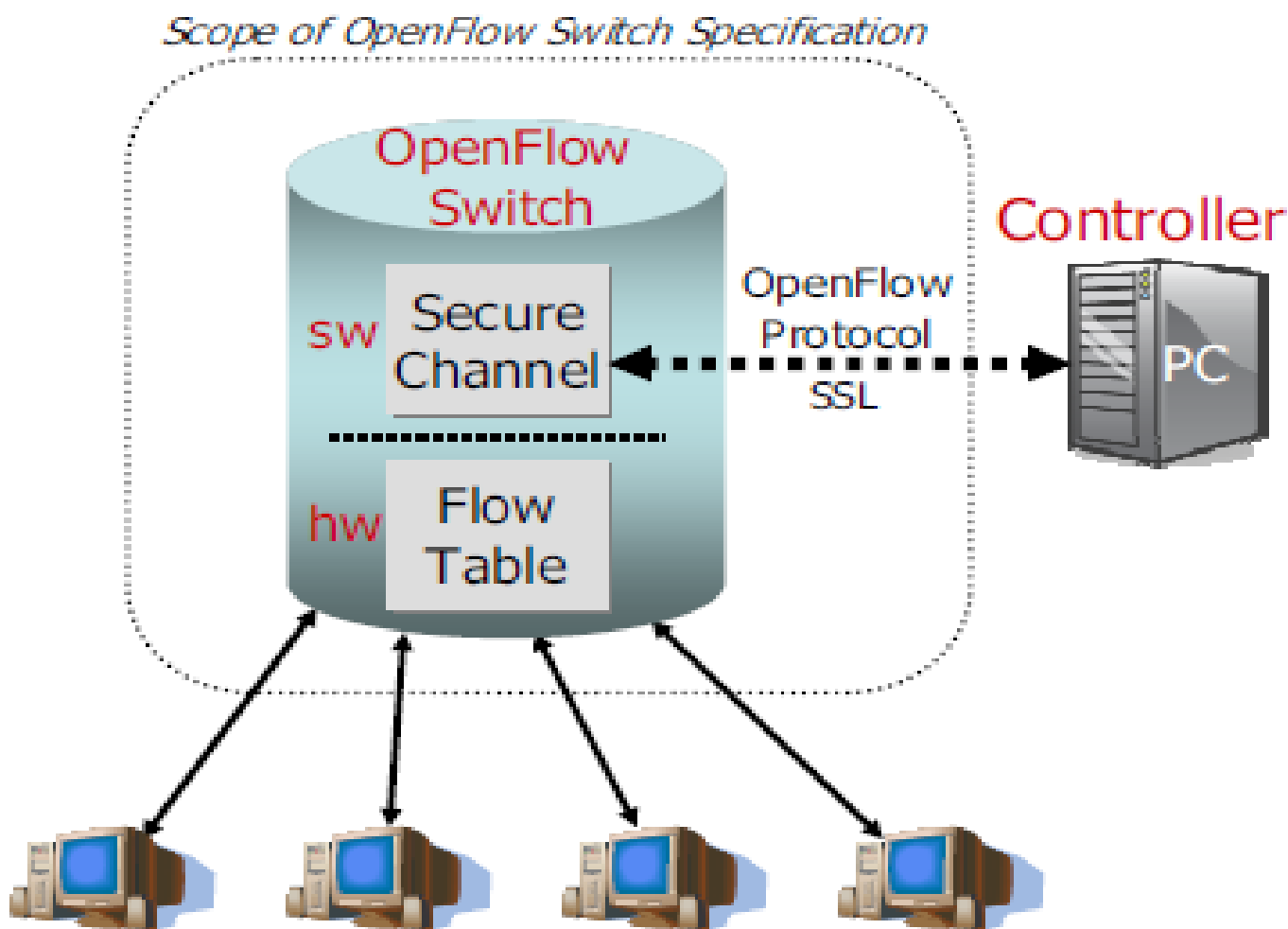
OpenFlow е в основата на software-defined networking (**SDN**). Потребителите дефинират потоците данни и пътищата им независимо от инфраструктурата под тях – маршрутизатори и комутатори.

Проект с **отворен код** – сътрудничество между Stanford University и University of California at Berkeley.

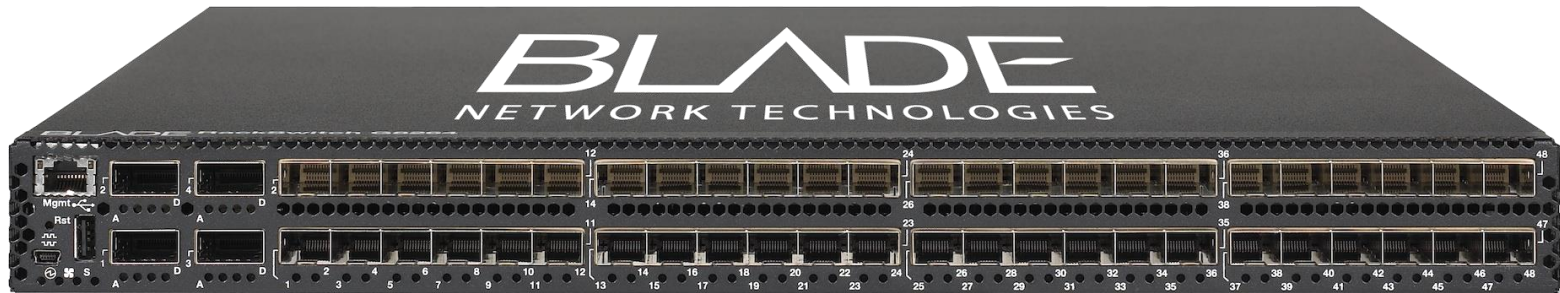
RouteFlow (QuagFlow)



Идеален OpenFlow суич



OpenFlow суичове. IBM G8264.



IBM OpenFlow суич **G8264**:

48 × 10 GbE SFP+ порта и

4 × 40 GbE QSFP+.

OpenFlow суичове. HP 8200 zl.



Освен това:

HP 6600, HP 6200-24G-mGBIC, HP 5400 и
HP 3500

RouteFlow (QuagFlow)

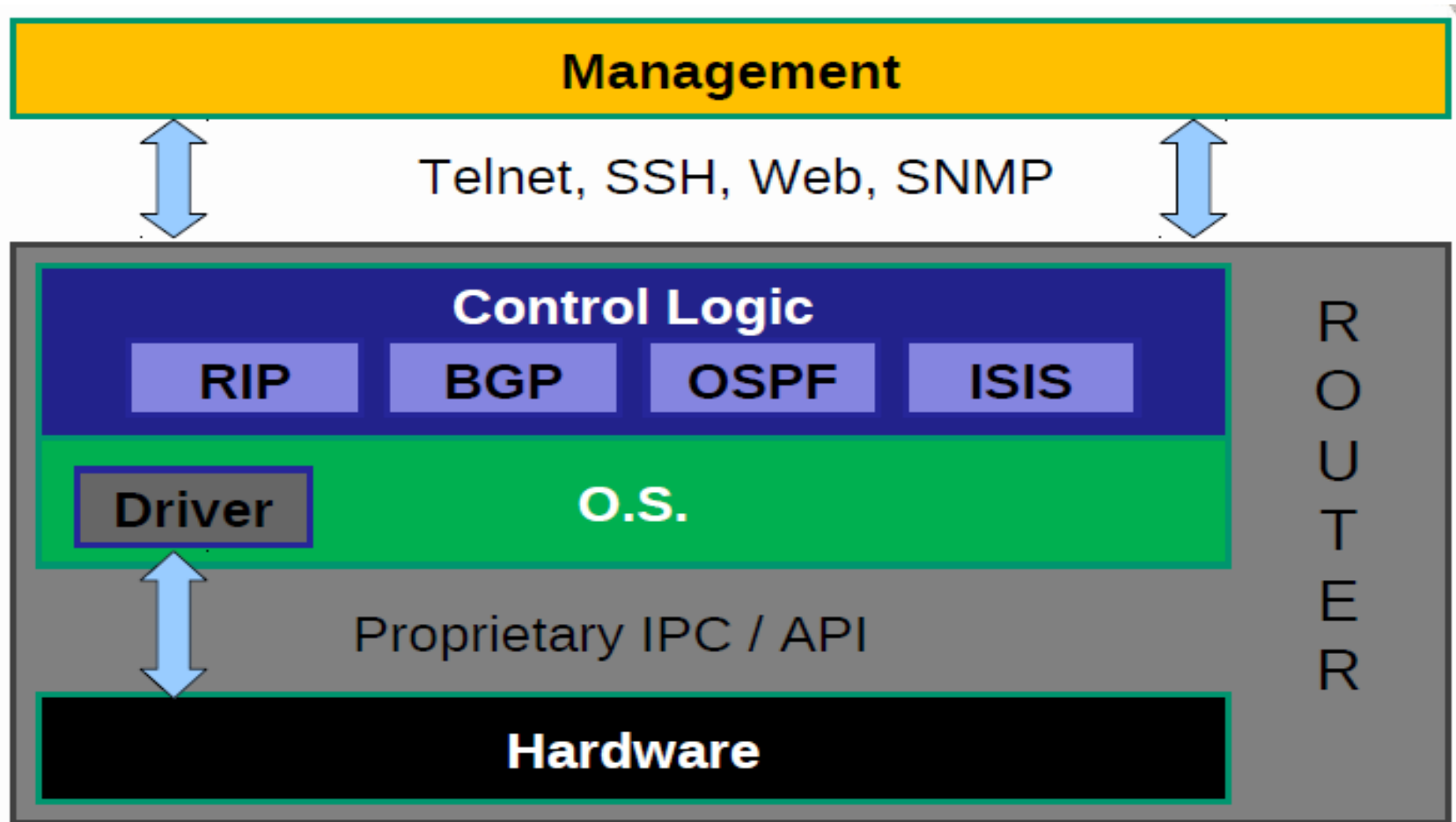
Параметри:

- **open-source** протоколни стекове (напр. **Quagga**)
- комерсиален мрежов хардуер с отворени API-та (напр. OpenFlow суичове от HP, IBM)

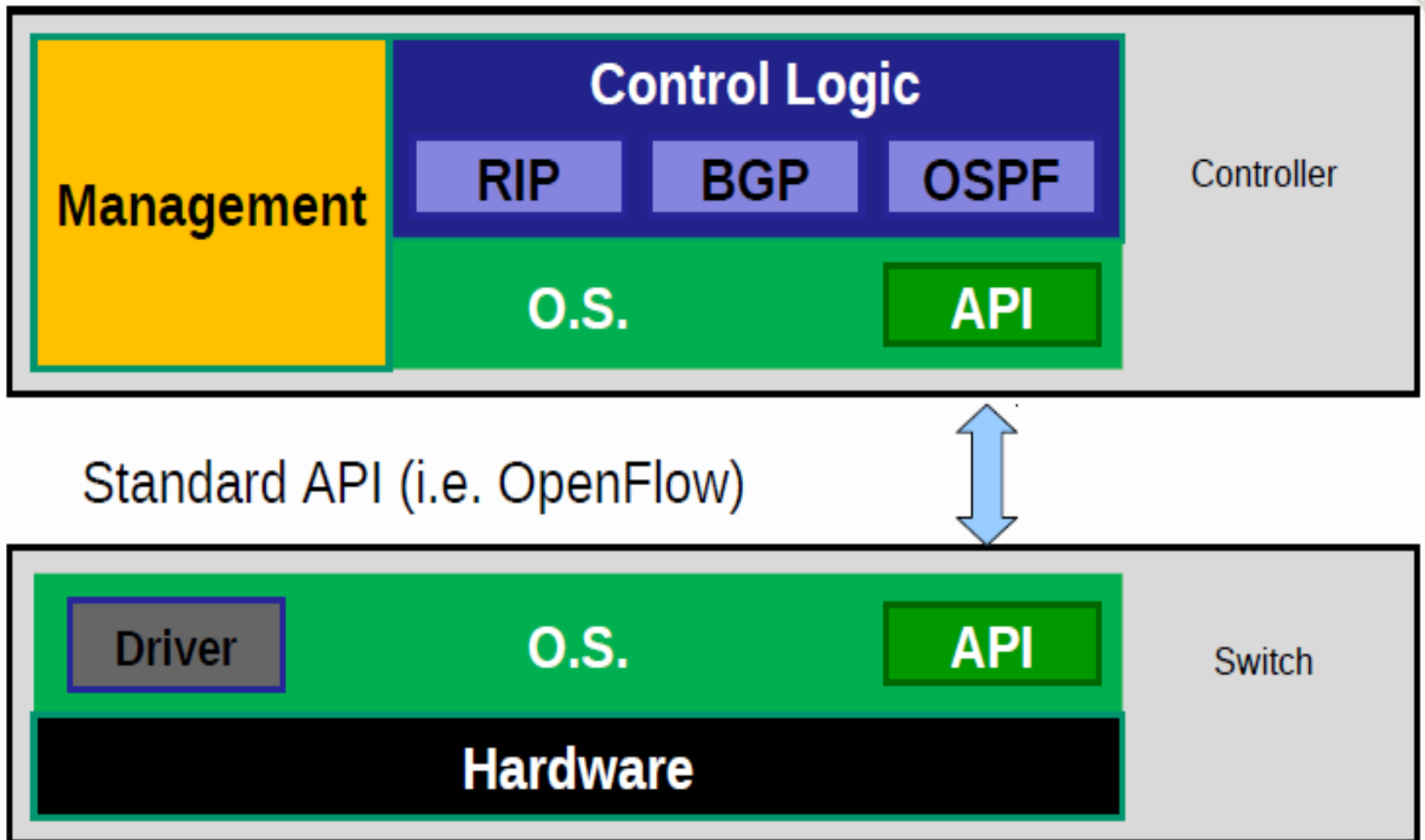
Постига се:

- производителност, съизмерима със скоростта на линията;
- оптимално съотношение цена/производителност;
- гъвкавост.

Архитектура на маршрутизатор (сега)



OpenFlow модел



Open Networking Foundation

Версия 1.1 на OpenFlow - на 28.02.2011, поддържана от openflow.org.

Развитието на стандарта се поема от Open Networking Foundation (ONF).

www.opennetworking.org

Пред от ONF е OpenFlow v. 1.3.4.

OpenFlow v.1.2 суиц

