12. Маршрутни алгоритми

Обхождане, наводняване, метод на Берън.

Маршрутни алгоритми

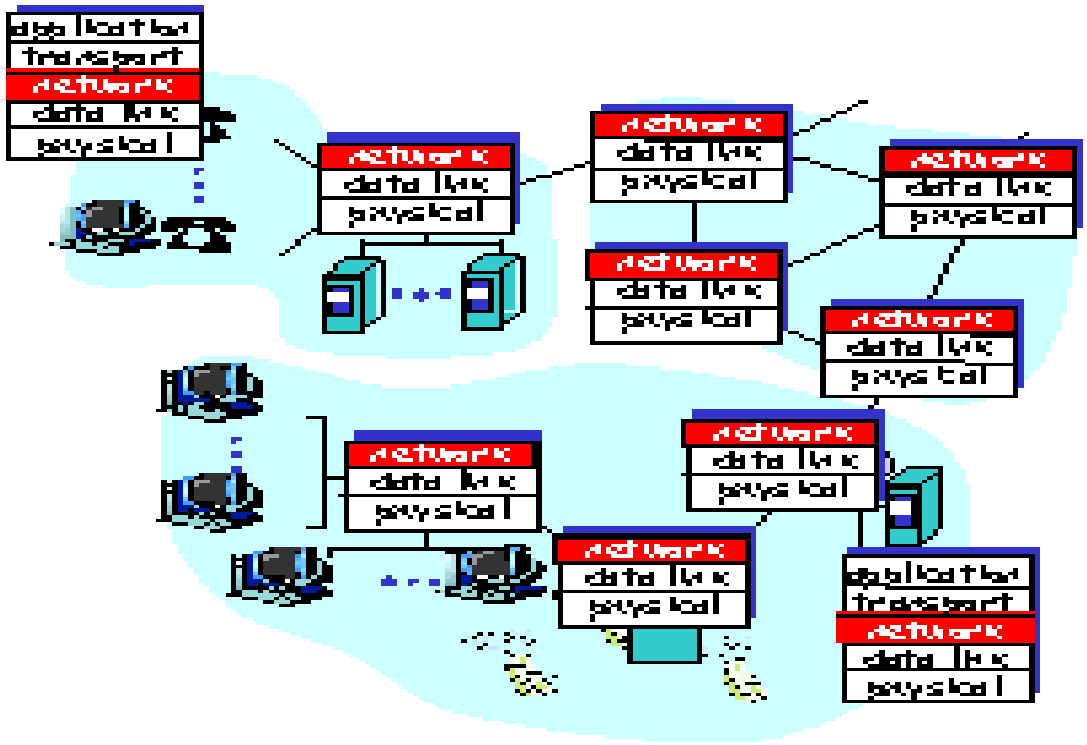
Основната функция на мрежовото ниво е да маршрутизира пакетите от източника към получателя - през няколко хопа.

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 взима наново за всяка 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.

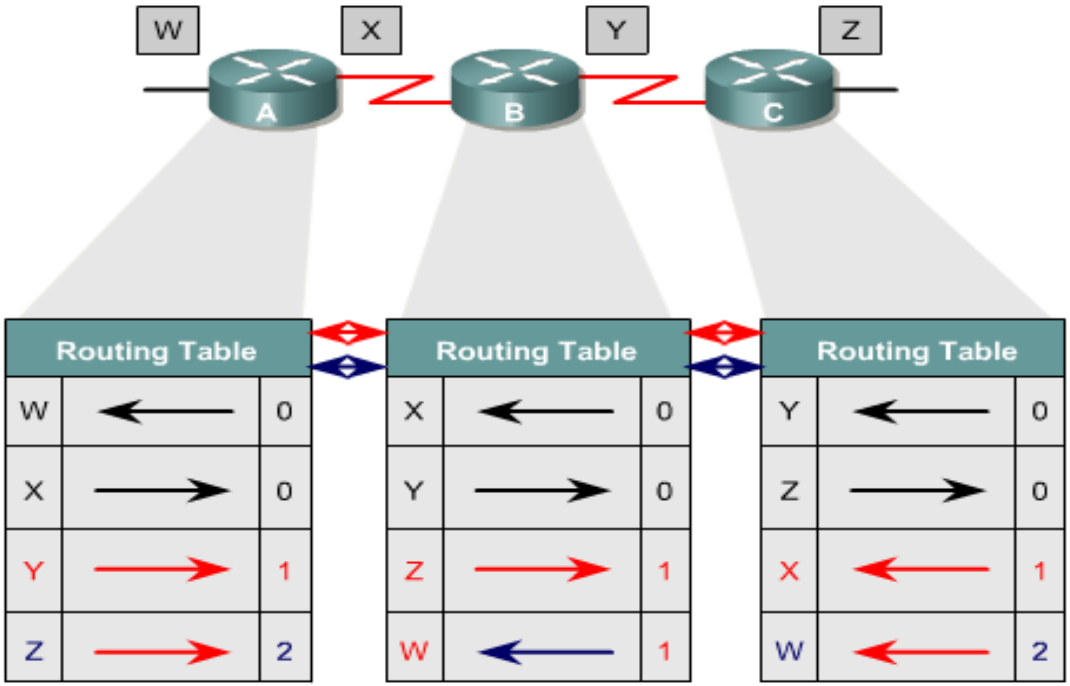
Мрежата представяме като граф - върховете са възлите в мрежата, а дъгите са комуникационните линии.

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 мине пакета до крайната точка (мрежа).



Адаптивни алгоритми. Скорост на сходимост (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 сходимост**:

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

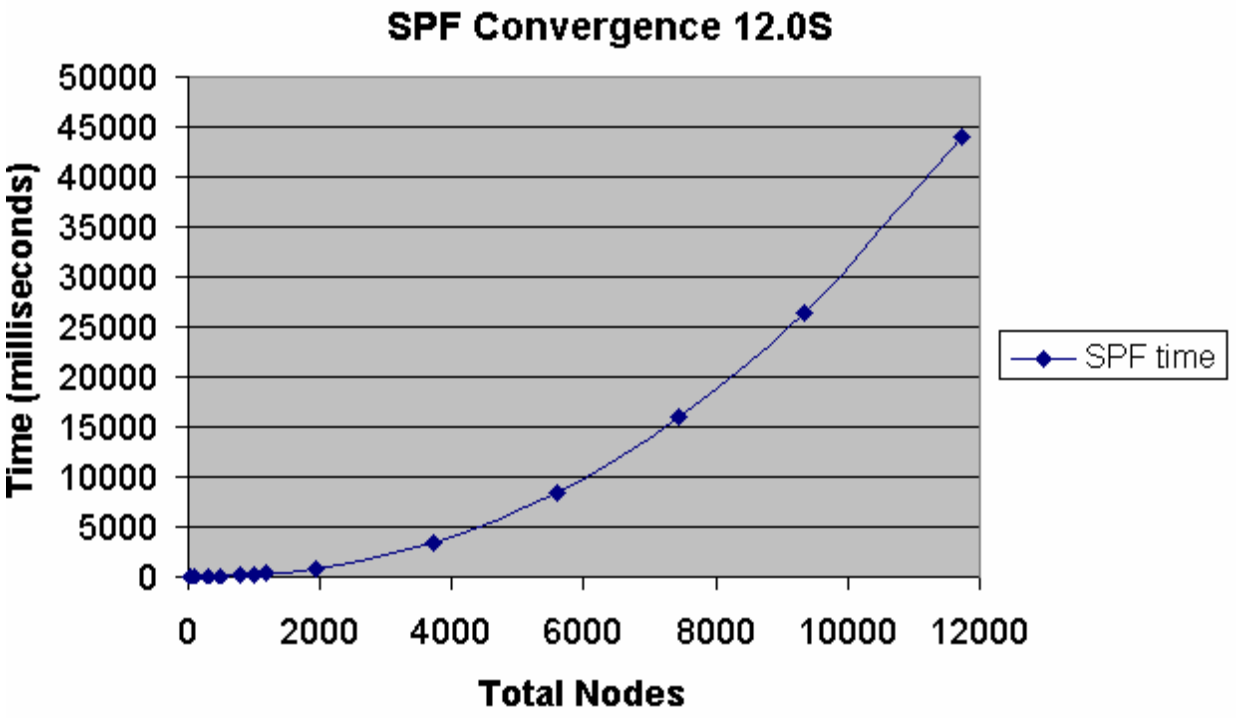
Алгоритъмът на Дейкстра е алгоритъм за намиране на най-къс път в граф от даден връх до всички останали върхове.

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).

Сходимост (конвергенция)



Метод на наводняването

Методът на наводняването (flooding) е статичен алгоритъм за маршрутизация.

Когато един пакет пристигне по дадена линия, той се изпраща по всички останали линии.

Получават се дубликати на пакетите (тъй като между два възела обикновено има повече от един път).

За да се избегне този проблем, се въвежда идентификация на пакетите - всеки пакет съдържа идентификация на възела-източник и пореден номер.

Метод на наводняването

Всеки маршрутизатор поддържа по един списък за всеки възел-източник, съдържащ номерата на пакетите, които са минали през него.

Ако пристигнал пакет присъства в списъка, той не се препредава, в противен случай списъкът се актуализира и пакетът се предава.

Списъците може да нарастнат много - при една голяма мрежа алгоритъмът не е приложим.

Метод на наводняването

За намаляване на дължината на списъка може да се добави допълнителен брояч *k*, който указва, че всички пакети до номер *k* са предадени.

Пакетите с номера по-малки от *k* може да не присъстват в съответния списък.

Метод на случайното обхождане

Това е неадаптивен алгоритъм.

Когато един пакет пристигне на някоя линия, той се изпраща по случаен начин по една от останалите линии.

Този метод няма сигурна сходимост.

Ако всеки възел помни историята си, т.е. кой пакет в коя посока е изпратил, то може да се предотврати повторно изпращане на един и същи пакет в една и съща посока, което ще доведе до сходимост.

Такава информация, обаче, ще има твърде голям обем.

Изолирана маршрутизация. Алгоритъм на Берън

При изолираната маршрутизация всеки възел се разглежда сам за себе си.

За всяка линия възелът поддържа по една изходяща опашка.

При алгоритъма на Берън всеки пристигнал пакет се добавя към най-късата изходяща опашка в момента на пристигането.

Този алгоритъм се адаптира към пиковете на предаване, но маршрутите не са оптимални.

Quagga

Quagga е open source софтуерен пакет за маршрутизация.

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

Quagga рутер

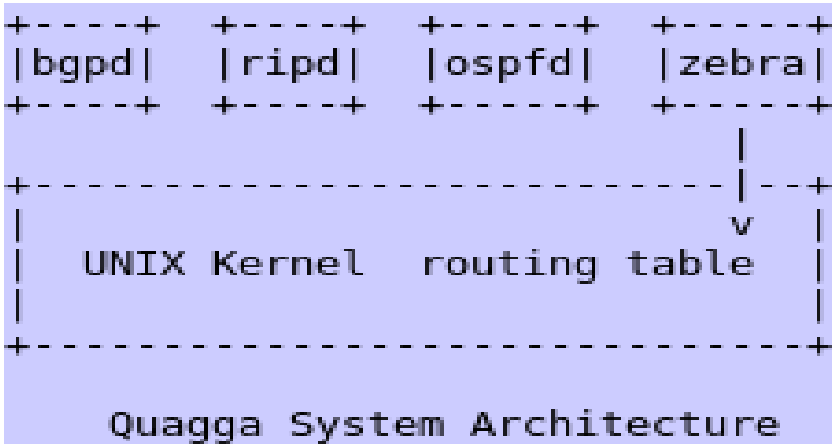
Quagga рутер

Компютър с Quagga си е рутер със **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