**Майкъл Зарков, ф.н. 45655, спец. Информатика**

**Задача 1**

Сравнете динамичните маршрутизиращи протоколи: “distance vector” и „link-state“.

**Distance vector routing**

При този маршрутизиращ протокол всеки рутер пази таблица (вектор) от дължината на най-късия път (за който рутера знае до момента) до всеки друг участник в мрежата. Рутерите в мрежата обменят информация за тези таблици с директните си съседи и при наличие на по-къси пътища обновяват своите рутинг таблици. След достатъчно повторения на обмена и обновяването на таблиците, в даден момент най-късите пътища са открити и рутерите спират да пращат рутинг таблици, тъй като в тях няма промяна. Алгоритъмът на Белман-Форд, стои зад обновяването на таблиците. За „дължина“ на връзката често се ползва „hop count” (броят на връзките между изпращача и получателя) или закъснението на сигнала на всяка връзка.

**Link-state routing**

При този маршрутизиращ протокол основните стъпки за попълване на рутинг таблиците са следните:

1. Всеки рутер открива кои са му съседите, чрез пращане на HELLO пакети по всяка връзка.
2. На всяка връзка се поставя цена(тегло). Това може да стане автоматично или човек ръчно да я въведе. Обикновено цената е обратно пропорционална на пропускателната способност на връзката. Ако връзките са много дълги (има голямо закъснение), може и това да се отчете в цената.
3. След като горните две неща са установени, всеки рутер изпраща на всички останали пакет със информацията за съседите си и теглото на връзките.
4. На края всеки рутер трябва да е получил всички пакети с информацията пратена от другите. Сега всеки рутер чрез алгоритъма на Дейкстра изчислява най-късите пътища до всяка дестинация и попълва рутинг таблицата си.

**Разлики между двата протокола**

Първо се появява протоколът **“distance vector”**, но в момента **“link-state”** е много по-разпространен. Една много основна причина за това е проблемът за броенето до безкрайност при “distance vector”. Той може да се появи, когато връзка е прекъсната или някой рутер е в неизправност. Отнема много време докато информацията за неизправност достигне до всички рутери в мрежата и това е защото при този протокол всеки рутер пази само локална информация за цялата топология на мрежата (връзките със съседите). При протокола “link-state” всеки рутер пази информация за цялата топология на мрежата и този проблем не съществува.

От друга страна “distance vector” е много по-пестелив от към памет и изчисление. Паметта нужна при “distance vector” е линейна в броя на рутерите докато при “link-state”, ако имаме N рутера всеки с K съседа, нужната памет е пропорционална на N\*K.

**Задача 2**

Опишете стъпките за процеса на договаряне при транспортния протокол TCP.

TCP използва “three-way-handshake” със следните стъпки:

1. Клиентът, който иска да отвори връзката, изпраща към дестинацията TCP сегмент със „сетнат“ SYN бит, указан порт, към който иска да се свърже, и началния номер на клиента.
2. Когато горният сегмент стигне до дестинацията или се изпраща сегмент за отхвърляне или установяване на връзка. При отхвърляне се праща сегмент със „сетнат“ RST бит. При установяване на връзка сървърът изпраща сегмент със своя начален номер и началния номер на клиента + 1.
3. На края клиентът трябва да потвърди получаването на този сегмент от сървъра като изпрати сегмент с началния номер на сървъра + 1.

**Задача 3**

Посочете транспортния протокол и портовете по подразбиране (well-known

ports) на следните услуги: SMTP, SNMP, DNS, DHCP, HTTP, HTTPS, FTP, SSH, POP3, IMAP, TELNET.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Услуга** | **Транспортен протокол** | **Портове по подразбиране** |
| 1. | SMTP | TCP | 25, 587, 465 |
| 2. | SNMP | UDP | 161, 162 |
| 3. | DNS | UDP | 53 |
| 4. | DHCP | UDP | 67, 68 |
| 5. | HTTP | TCP | 80 |
| 6. | HTTPS | TLS | 443 |
| 7. | FTP | TCP | 20, 21 |
| 8. | SSH | TCP | 22 |
| 9. | POP3 | TCP, TLS, SSL | 110, 995 |
| 10. | IMAP | TCP | 143 |
| 11. | TELNET | TCP | 23 |