

Мрежов слой

Маршрутизация

Маршрутизация

- Определяне/изчисляване на оптимален маршрут за пренасяне на пакети през дадена мрежа от даден подател към определен/и получател/и, въз основа на определен/и критерий/и за оптималност.
- Видове**
 - Определяне на следващия маршрутизатор/скок (*next-hop routing*) – най-често използван вид!
 - Определяне на целия маршрут от подателя до получателя (*source routing*) – използва се от хостовете в определени случаи, например:
 - Еталонно тестване на мрежи (*benchmarking*)
 - Маршрутизация в някои видове безжични мрежи (multi-hop ad hoc networks, MANETs)
- Маршрутизиращи алгоритми**
 - Изчисляват оптимален маршрут (най-кратък път) през мрежата
 - Попълват и актуализират маршрутизиращи таблици
- Маршрутизиращи таблици**
 - Поддържани в мрежовите възли
 - Съдържат информация за най-краткия път до всяка дестинация, с адреса на следващия маршрутизатор, към който трябва да се предадат пакетите.
 - Могат да се актуализират чрез обмен на информация между възлите за състоянието на мрежата

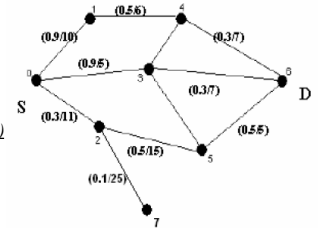
Маршрутизация: Критерии и метрики

Критерий	Метрика
MIN транзитно време / MIN време на закъснение	Закъснение на предаването (<i>msec</i>) / Географско разстояние (<i>km</i>)
MIN брой скокове	Брой на скоковете
MIN брой на пакетите в опашките по маршрута	Средно закъснение поради изчакване в опашките (<i>msec</i>)
MAX пропускателна способност	Скорост на предаване (<i>bps</i>)
MAX надеждност / сигурност	BER / ниво на шифриране
MIN комуникационни разходи	€ / \$

Обикновено се използват съставни метрики, чрез комбинирание на прости метрики в претеглена формула, с цел образуване на елинна метрика (*cost*).

Маршрутизация: Типове метрики

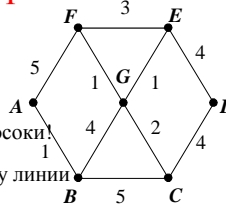
- Добавящи се (*additive*)**
(напр., закъснение, вариация на закъснението (*jitter*), (логаритъм на) брой успешни предавания, брой скокове, \$/€ цена)
– Метриката на даден път е сума от метриките на съставните му комуникационни линии.
 $M_{i,j}$ is the metric for link $\{i, j\}$
 $M(P)$ = metric of path
Additive : $M(P) = M_{i,j} + M_{i,k} + \dots + M_{i,m}$
- Умножаващи се (*multiplicative*)**
(напр., надеждност, вероятност за успешно предаване)
– Метриката на даден път е произведение от метриките на съставните му комуникационни линии.
Multiplicative: $M(P) = M_{i,j} \times M_{i,k} \times \dots \times M_{i,m}$
- С MIN-MAX стойност (*concave*)**
(например, скорост)
– Метриката на даден път е MIN или MAX стойност на метриките на съставните му комуникационни линии.
Concave : $M(P) = \min\{M_{i,j}, M_{i,k}, \dots, M_{i,m}\}$



Маршрутизация:

Определяне на най-краткия път

- Всяка линия има цена на използване в зависимост от нейните метрики:
 - Дължина
 - Закъснение
 - Ниво на задръстване
 - \$/€
- Цената може да е различна в двете посоки!
- Цената за използване на даден път е функция на цените на съставните му линии
- Най-кратък е пътят с MIN цена!**
- Определяне на най-краткия път:
 - Построяване на граф на мрежата
 - Възлите му са маршрутизаторите
 - Дъгите му са комуникационните линии
 - Изчисляване на цената на всяка дъга като претеглена функция на всички метрики, използвани от маршрутизиращия алгоритъм.
 - $\text{Hop cost} = [(K1 * \text{bandwidth}) + (K2 * \text{bandwidth}) / (256 - \text{load}) + (K3 * \text{delay})] * [(K5 / (\text{reliability} + K4))] * 256$
 - Избиране на маршрут, съответстващ на най-краткия път през мрежовия граф.

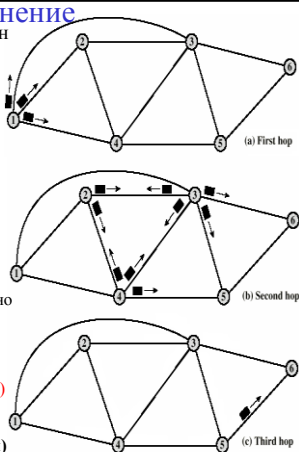


Класически маршрутизиращи стратегии

- Наводнение (*flooding*)
- По случаен начин
- Статична/фиксирана
- Динамична

Наводнение

- Всеки пакет се комутира от даден маршрутизатор към всички негови портове с изключение на този, на който е пристигнал.
 - Няколко копия на пакета пристигат до получателя
 - Всеки пакет е уникално номериран за установяване и отстраняване на копията (маршрутизаторите помнят пакетите вече изпратени от тях и премахват повторните копия за да намалят натоварването на мрежата)
 - Пакетите могат да съдържат и броя на скоковете, направени от тях.
- Вариация
 - Всеки пакет се предава само към няколко линии, водещи приблизително в правилната посока.
- **Предимства**
 - MIN настройка
 - Оцеляване (подходяща за военни приложения)
- **Недостатъци**
 - Задръствания (огромен брой дублирани пакети)



Маршрутизация по случаен начин

- Всеки маршрутизатор избира само 1 изходящ порт за препредаване на входящ пакет
 - Селекцията може да бъде по случаен признак или по ред на номерата в кръг (*round robin*)
- Подобрение
 - Избиране на изходящ порт въз основа на вероятности
- **Не е необходима информация за мрежата**
 - Няма маршрутизиращи таблици
- Проста стратегия с много по-малко натоварващ трафик от наводнението
- **Не се минимизират разходите/цената!**

Статична маршрутизация

- **Не е адаптивна**
 - Не се основава на измервания или оценки на текущия мрежов трафик и топология
- Маршрути
 - Предварително изчислени (*off-line*)
 - Зареждат се в маршрутизаторите при инициране на мрежата
- Маршрутизиращи таблици
 - Съдържат статична информация
 - Създадени и дефинирани преди функциониране на мрежата
 - Рядко актуализирани
 - Само при значителна промяна в топологията на мрежата
 - Въз основа на очаквания трафик или капацитет
- **Предимства**
 - Простота
- **Недостатъци**
 - Липса на гъвкавост
 - Липса на реакция при повреда или претоварване/задръстване на мрежата
- Приложения
 - Малки мрежи
 - Гръбнаци на големи мрежи

Динамична маршрутизация

- Използвана от почти всички видове мрежи с комутация на пакети
- **Адаптивна**
 - Променя се във времето с цел отразяване на промените в топологията и трафика на мрежата, причинени от повреди и претоварвания/задръствания.
 - Изисква информация за мрежата
 - Маршрутизиращите таблици се актуализират често и автоматично
- **Предимства**
 - Подобрена производителност
 - Подпомагане контрола на задръстванията (*congestion control*)
- **Недостатъци**
 - Решенията са по-сложни
 - Компромис между качеството на информацията за мрежата и допълнителните разходи (режийните)
 - Твърде бързата реакция може да предизвика осцилации (колебания) в маршрутизирането
 - Твърде бавната реакция може да доведе до използване на информация, която вече е остаряла.
 - Теоретичните ползи може да не се осъществят!
- 2 вида контрол
 - Централизиран
 - Разпределен

Динамична маршрутизация: Централизиран контрол

- 1 управляващ възел изчислява всички маршрути (по зададен алгоритъм) и актуализира маршрутизиращите таблици на всички останали мрежови възли
- Нова информация за мрежата (необходима за актуализация) се получава чрез специални пакети за поддръжка (*maintenance packets*)
 - Информират за закъснения, откази на линии/възли, задръствания и др.
- **Не е толкова надежден колкото разпределения контрол**
 - Единична точка на отказ
 - Необходимост от резервиране
- **Няма задръствания!**

Динамична маршрутизация: Разпределен контрол

- **Всеки** маршрутизатор сам изчислява маршрути (по зададен алгоритъм) и актуализира своя собствена маршрутизираща таблица, въз основа на информация:
 - Получена от другите маршрутизатори
 - Извлечена от преминаващите пакети
- **По-надежден от централизиран контрол**
- Може да предизвика задръствания

Динамични разпределени маршрутизиращи алгоритми: Видове

- **С използване на дистанционен вектор (distance vector)**
 - Вектор, съдържащ разстоянието/цената и пътя (следващия маршрутизатор) до всички известни дестинации
 - Съхранява се във всеки маршрутизатор и се обменя само със съседните маршрутизатори
 - **Недостатъци:**
 - Бавна сходимост на маршрути
 - Нестабилност
 - Тенденция за създаване на (затворени) маршрутни кръгове в мобилни мрежи
- **С използване на състоянието на линиите (link state)**
 - Поддържане на информация за глобалната топология на мрежата в маршрутизаторите чрез периодично наводнение от страна на всеки маршрутизатор с информация за състоянието на неговите линии
 - **Недостатъци:**
 - Голям трафик, предизвикван от наводненията.
 - Огромно количество ресурси за изчисляване на маршрути в маршрутизаторите
 - Мобилността води до чести наводнения!

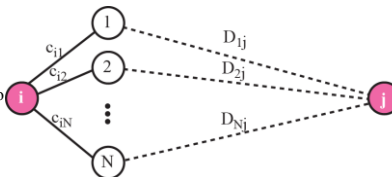
Дистанционно-векторна маршрутизация (distance vector routing)

- Споделяне на информация:
 - За **цялата** мрежа (интернет)
 - Само между **съседите**
 - На **редовни** интервали
- Определяне на най-краткия път чрез използване на:
 - Алгоритъм на **Bellman** (1957)
 - Алгоритъм на **Ford-Fulkerson** (1962)

Алгоритъм на Bellman-Ford

- Ако всеки съсед k на възел i знае най-краткото разстояние D_{kj} до възел j , тогава най-краткото разстояние D_{ij} между възлите i и j може да бъде намерено чрез добавяне на разстоянието c_{ik} между възел i и всеки негов съсед k към съответното разстояние D_{kj} и избиране след това на **MIN** разстояние измежду всичките.
- Въз основа на **принципа за оптималност**
 - Ако k се намира на оптималния път от i и j , тогава оптималният път от k до j също попада на този маршрут.

$$D_{ij} = \text{minimum} \{ (c_{i1} + D_{1j}), (c_{i2} + D_{2j}), \dots, (c_{iN} + D_{Nj}) \}$$



Legend

D_{ij} Shortest distance between i and j
 c_{ij} Cost between i and j
 N Number of nodes

Figure 11.4

Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010

Дистанционно-векторна маршрутизация: Във взаимосвързани мрежи (интернет)

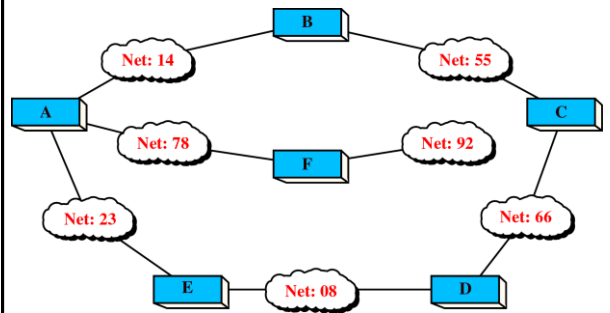


Figure 21-17

Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Дистанционно-векторна маршрутизация: Концепция

Концепция

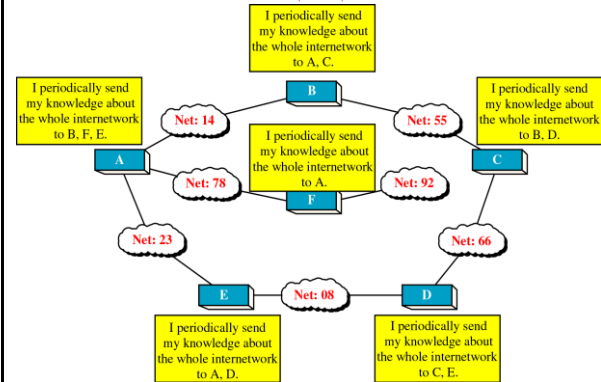


Figure 21-18

Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Дистанционно-векторна маршрутизация: Маршрутизиращи таблици

➤ **ВСЕКИ МАРШРУТИЗАТОР ПОДДЪРЖА СВОЯ СОБСТВЕНА ТАБЛИЦА**

Network ID	Distance	Next Hop
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

➤ **КОЛОНИ**

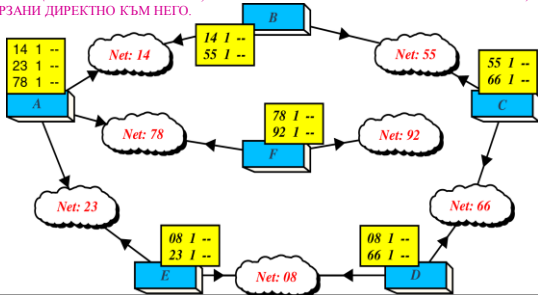
• **NETWORK ID** - АДРЕС НА **ВСЯКА МРЕЖА** ИЗВЕСТНА НА МАРШРУТИЗАТОРА

• **Distance** - **НАЙ-КРАТКОТО РАЗСТОЯНИЕ** ДО СЪОТВЕТНАТА МРЕЖА (измерено с приетата метрика, т.е. *брой скокове, закъснение, € цена, ...*)

• **NEXT HOP** - АДРЕС НА **СЛЕДВАЩИЯ МАРШРУТИЗАТОР**, КЪМ КОЙТО ТРЯБВА ДА БЪДАТ ПРЕДАДЕНИ ПАКЕТИТЕ ЗА ДА ДОСТИГНАТ ДО КРАЙНОТО СИ МЕСТОНАЗНАЧЕНИЕ.

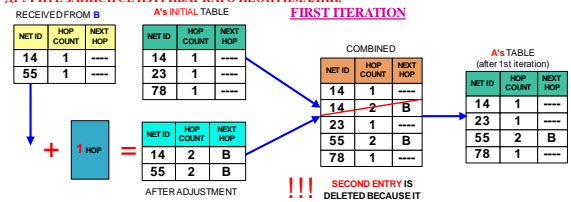
ПЪРВОНАЧАЛНО ЗАПЪЛВАНЕ НА МАРШРУТИЗИРАЩИТЕ ТАБЛИЦИ (0. итерация)

- ВСЕКИ МАРШРУТИЗАТОР ИМА ТОЛКОВА ЛОГИЧЕСКИ АДРЕСА, КОЛКОТО СА МРЕЖИТЕ, КЪМ КОИТО Е СВЪРЗАН.
- ЧРЕЗ ИНСПЕКТИРАНЕ НА ЛОГИЧЕСКИТЕ СИ АДРЕСИ, МАРШРУТИЗАТОРЪТ ОТКРИВА КЪМ КОИ МРЕЖИ Е СВЪРЗАНИ И КАКВИ СА СЪОТВЕТНИТЕ ЗНАЧЕНИЯ НА МЕТРИКИТЕ (в този случай се използват скокове; броят на скоковете към директно свързаните мрежи е 1).
- ТРЕТАТА КОЛОНА "СЛЕДВАЩ СКОК" Е ПРАЗНА – ТЪЙ КАТО В НАЧАЛОТО (ПРИ ИНИЦИИРАНЕ НА ИНТЕРНЕТА) МАРШРУТИЗАТОРЪТ ПОЗНАВА САМО МРЕЖИТЕ, СВЪРЗАНИ ДИРЕКТНО КЪМ НЕГО.



ЗАПЪЛВАНЕ НА ТАБЛИЦАТА НА МАРШРУТИЗАТОР А (1. итерация)

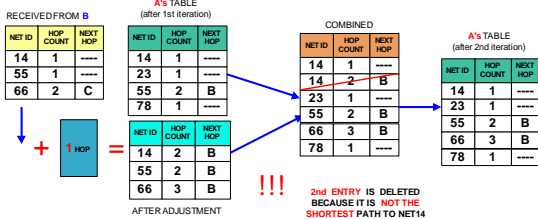
- СЪСЕДНИТЕ МАРШРУТИЗАТОРИ НА А (т.е. В, Е и F) ИЗПРАЩАТ ПЪРВОНАЧАЛНИТЕ СИ МАРШРУТИЗИРАЩИ ТАБЛИЦИ НА А
- А НАГЛАЖДА ИНФОРМАЦИЯТА, СЪДЪРЖАЩА СЕ В ПОЛУЧЕНИТЕ ТАБЛИЦИ, Т.Е. ДОБАВЯ РАЗСТОЯНИЕТО ОТ НЕГО ДО СЪОТВЕТНИЯ СЪСЕД (В СЛУЧАЯ 1) КЪМ ВСЕКИ ПЪТ, ПОЛУЧЕН ОТ ТОЗИ СЪСЕД.
- А ОСТАВЯ САМО 1 ЗАПИС ЗА ВСЯКА МРЕЖА В ТАБЛИЦАТА СИ (В СЛУЧАЯ, ЗАПИСЪТ С МИНИМАЛЕН БРОЙ СКОКОВЕ).
- В СЛУЧАЙ НА 2 (ИЛИ ПОВЕЧЕ) ЗАПИСА СЪС СЪЩИ БРОЙ СКОКОВЕ ДО ДАДЕ НА МРЕЖА, САМО ЗАПИСЪТ С МИНИМАЛНА СТОЙНОСТ НА СЛЕДВАЩАТА КООРДИНАТА (Т.Е. "NEXT HOP") АДРЕС НА СЛЕДВАЩИЯ МАРШРУТИЗАТОР ОСТАВА В ТАБЛИЦАТА; ДРУГИТЕ ЗАПИСИ СЕ ИЗТРЯВЯТ КАТО НЕОПТИМАЛНИ.



ЗАПЪЛВАНЕ НА ТАБЛИЦАТА НА МАРШРУТИЗАТОР А (2. итерация)

- ПО ВРЕМЕ НА 1. ИТЕРАЦИЯ, В Е ОБМЕНИЛ ПЪРВОНАЧАЛНАТА СИ ТАБЛИЦА СЪС СВОЯ ДРУГ СЪСЕД С И СЕГА ИМА НОВА ТАБЛИЦА, КОЯТО ИЗПРАЩА НА А.
- ПРИЛАГАТ СЕ СЪЩИТЕ ПРАВИЛА КАКТО ПРИ 1. ИТЕРАЦИЯ

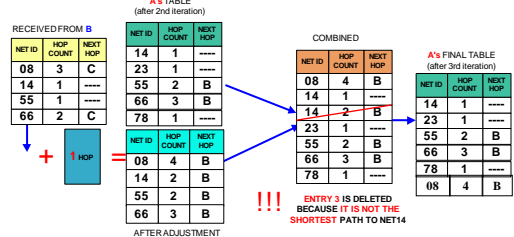
SECOND ITERATION



ЗАПЪЛВАНЕ НА ТАБЛИЦАТА НА МАРШРУТИЗАТОР А (3. итерация)

- ПО ВРЕМЕ НА 2. ИТЕРАЦИЯ В Е ОБМЕНИЛ ИНФОРМАЦИЯ СЪС СВОЯ ДРУГ СЪСЕД С И ИМА НОВА ТАБЛИЦА, КОЯТО СЕГА ИЗПРАЩА НА А.
- ПРИЛАГАТ СЕ СЪЩИТЕ ПРАВИЛА КАКТО ПРИ ДРУГИТЕ ИТЕРАЦИИ

THIRD ITERATION



Окончателни маршрутизиращи таблици

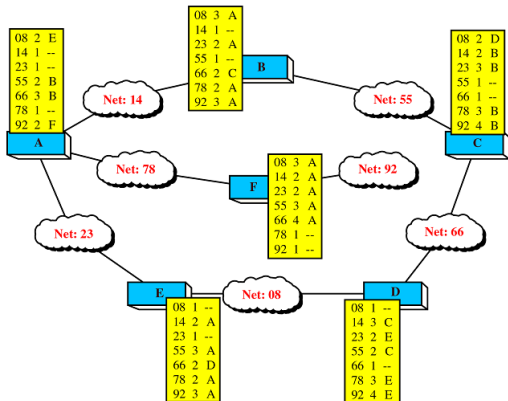
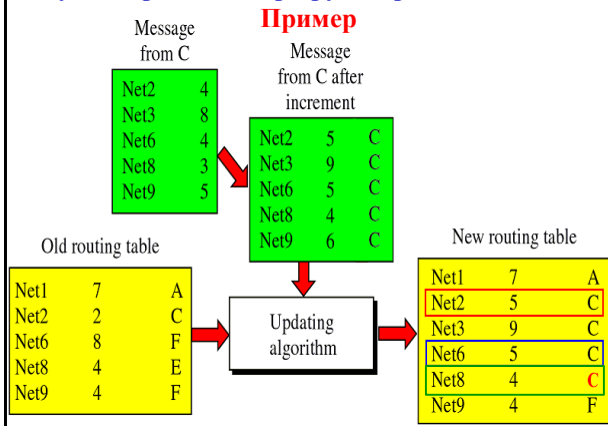


Figure 21-22 Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Актуализиране на маршрутизиращите таблици

- **периодично** (по ПРЕДВАРИТЕЛНО ЗАДАДЕНИ ИНТЕРВАЛИ) СЪСЕДНИТЕ МАРШРУТИЗАТОРИ ПРОДЪЛЖАВАТ ДА ОБМЕНЯТ СВОИТЕ ТЕКУЩИ ТАБЛИЦИ, СЪДЪРЖАЩИ ИНФОРМАЦИЯ ЗА ВСИЧКИ ИЗВЕСТНИ ИМ МРЕЖИ.
- ИЗПОЛЗВАЙКИ ТАЗИ ИНФОРМАЦИЯ, ВСЕКИ МАРШРУТИЗАТОР АКТУАЛИЗИРА СВОЯТА ТАБЛИЦА ЧРЕЗ ПРИЛАГАНЕ НА СЛЕДНИЯ **АКТУАЛИЗИРАЩ АЛГОРИТЪМ**:
 - ✓ Ако няма информация за рекламираната дестинация, маршрутизаторът я **добавя** незабавно в таблицата си;
 - ✓ В противен случай:
 - 1) Ако значението на полето **NEXT HOP** е същото (което означава, че информацията идва от същия рекламиращ маршрутизатор), новата информация **заменя** стария запис в таблицата, дори ако новият запис е с по-голямо **РАЗСТОЯНИЕ**!
 - 2) В противен случай (т.е. информацията идва от друг маршрутизатор):
 - Ако **РАЗСТОЯНИЕТО** в новия запис е **по-малко** от това в съществуващия (стар) запис, новият запис **измества** стария.
 - Ако **РАЗСТОЯНИЕТО** в новия запис е **по-голямо** от това в съществуващия (стар) запис, старият запис **остава** (т.е. таблицата **не се променя**).
 - Ако **РАЗСТОЯНИЕТО** в новия запис е **равно** на това в съществуващия (стар) запис, записът с **по-малко NEXT HOP** значение **остава** в таблицата.

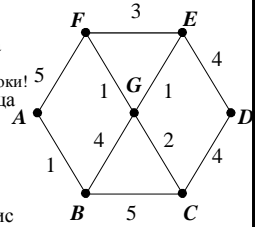
Актуализиране на маршрутизиращите таблици:



Дистанционно-векторна маршрутизация:

В мрежа

- Принципът е същият както при взаимосвързани мрежи
- Да предположим (за разнообразие), че разстоянието се измерва в **€ цените**.
- Всяка комуникационна линия има цена за изпращане на пакет през нея
 - Цена може да бъде различна в двете посоки!
- Всеки маршрутизатор поддържа таблица с информация за изпращане на пакети до всеки един друг маршрутизатор в мрежата, въз основа на минималната възможна цена.
- Маршрутизиращата таблица на даден маршрутизатор съдържа точно един запис за всеки един друг маршрутизатор (r) в мрежата, съдържащ:
 - Разстоянието (d) до съответния маршрутизатор;
 - Адреса/идентификатора на следващия маршрутизатор (n), към който трябва да се предават пакетите, предназначени за съответния маршрутизатор.



Първоначални маршрутизиращи таблици



- Първоначално всяка таблица съдържа информация само за съседите
 - Например, таблицата на маршрутизатор A съдържа информация само за маршрутизатори F и B
- Обменът на информация със съседите по време на следващите итерации завършва заглавяването на таблиците

Маршрутизиращи таблици след обмен на първоначалните таблици (1. итерация)

A	B	C	D	E	F	G
r d n	r d n	r d n	r d n	r d n	r d n	r d n
B 1 B	A 1 A	A 6 B	B 9 C	A 8 F	A 5 A	A 5 B
C 6 B	C 5 C	B 5 B	C 4 C	B 5 G	B 5 G	B 4 B
E 8 F	D 9 C	D 4 D	E 4 E	C 3 G	C 3 G	C 2 C
F 5 F	B 5 G	E 3 G	F 7 E	D 4 D	D 7 E	D 5 E
G 5 B	F 5 G	F 3 G	G 5 E	F 2 G	E 2 G	E 1 E
	G 4 G	G 2 G		G 1 G	G 1 G	F 1 F

- Всеки маршрутизатор изпраща своята таблица на съседите си
- Изпращаната таблица съдържа адреса на изпращащия маршрутизатор (записан в полето n), заедно с цената за изпращане на пакет от този маршрутизатор към всяка известна на него дестинация в мрежата.
- Маршрутизаторите променят таблиците си по такъв начин, че да сведат до минимум цената до всяка конкретна дестинация.
- След получаване на таблицата на съсед i , маршрутизатор j добавя цената на линията $i-j$ към всяка нова (получена) дестинация.
- В таблицата остават само най-кратките маршрути.

Окончателни маршрутизиращи таблици

A	B	C	D	E	F	G
r d n	r d n	r d n	r d n	r d n	r d n	r d n
B 1 B	A 1 A	A 6 B	A 10 C	A 6 G	A 5 A	A 5 B
C 6 B	C 5 C	B 5 B	B 9 C	B 5 G	B 5 G	B 4 B
D 10 B	D 9 C	D 4 D	C 4 C	C 3 G	C 3 G	C 2 C
E 6 B	E 5 G	E 3 G	E 4 E	D 4 D	D 6 G	D 5 E
F 5 F	F 5 G	F 3 G	F 6 E	F 2 G	E 2 G	E 1 E
G 5 B	G 4 G	G 2 G	G 5 E	G 1 G	G 1 G	F 1 F

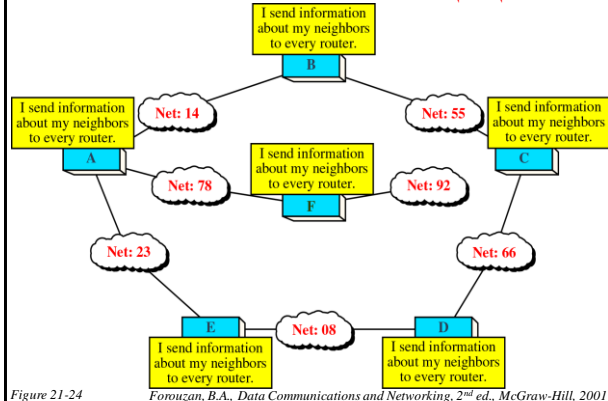
- По време на следващите итерации маршрутизаторите добавят новопридобитата информация за мрежата към своите таблици и ги изпращат на съседите си.
- По този начин, всеки маршрутизатор получава обратно своята собствена информация заедно с нова информация за другите съседни на своите съседни.
- След определен брой итерации всеки маршрутизатор ще получи информация за всеки друг маршрутизатор в мрежата.
- След това периодично всеки маршрутизатор продължава да изпраща информацията си за цялата мрежа към своите съседи. Това споделяне на информация се извършва, независимо дали мрежата се е променила от последния път, или не.

Маршрутизация с използване на

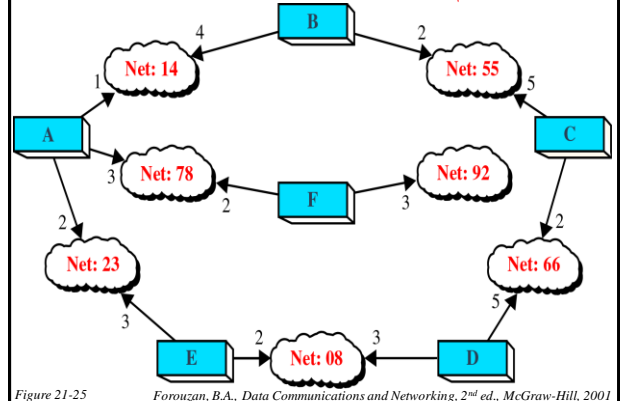
состоянието на линиите (link state routing)

- Споделяне на информация:
 - За **съседство**
 - Откриване на всички съседи чрез изпращане на пакети HELLO по всяка една комуникационна линия
 - Научаване на техните мрежови адреси
 - Оценяване состоянието на линията към всеки един от тях (например, какво е закъснението, скоростта, € цената и т.н.)
 - Например, закъснението се измерва чрез изпращане на ECHO пакети.
 - С **всички маршрутизатори** в мрежата (или интернет)
 - Създаване на Link State Packet (LSP), съдържащ текущото състояние на всички локални линии.
 - Изпращане на LSP до всички маршрутизатори, използвайки наводнение (flooding).
 - Всички LSP се потвърждават по линиите между всеки два маршрутизатора за предотвратяване на грешки
 - Веднага** когато има значителна промяна в състоянието на дадена линия
 - И периодично за да се гарантира премахването на стара информация
 - Период $LSP >$ Период pur (по-дългият период предпазва от създаването на прекалено голям трафик от наводненията)
- Формиране на база данни LSD (Link State Database), създаване на дърво с най-краткия път (с използване на алгоритъма на **Dijkstra**), изграждане/обновяване на маршрутизиращата таблица.

Маршрутизация с използване на състоянието на линиите: Концепция



Маршрутизация с използване на състоянието на линиите: Цена



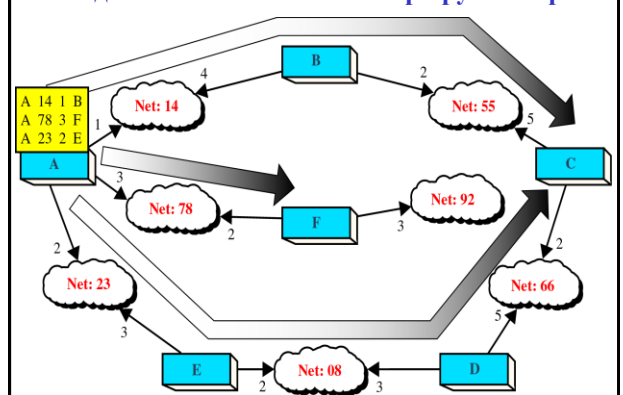
Пакет за състоянието на линиите (Link State Packet, LSP)

- Увеличава се с 1 за всеки нов изпратен LSP
- Използва се за отличаване на нови от стари LSP

- Предотвратява съществуването на стари LSP за дълго време в мрежата
- Подобно на TTL полето в IP протокола

Advertiser	Network	Cost	Neighbor	Sequence No	Age
.....
.....
.....

Наводнение с LSP пакет на маршрутизатор A

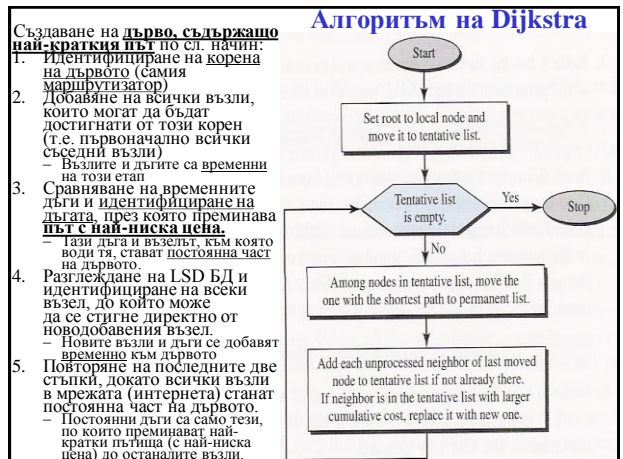


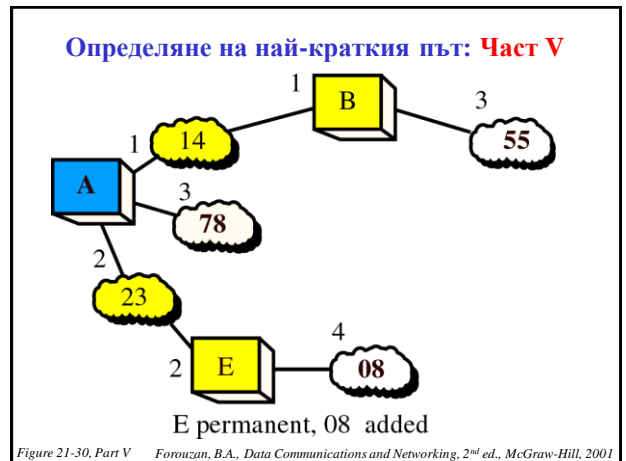
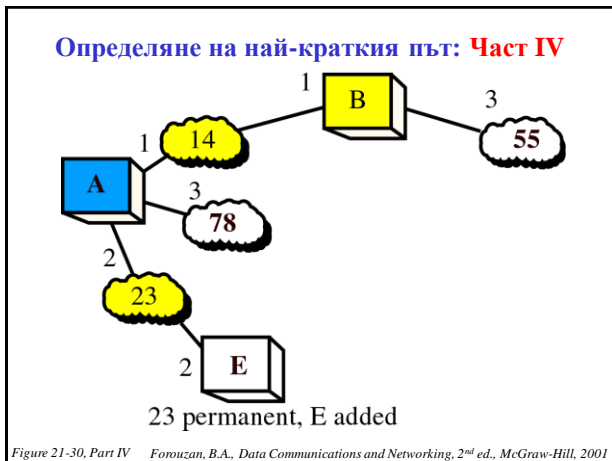
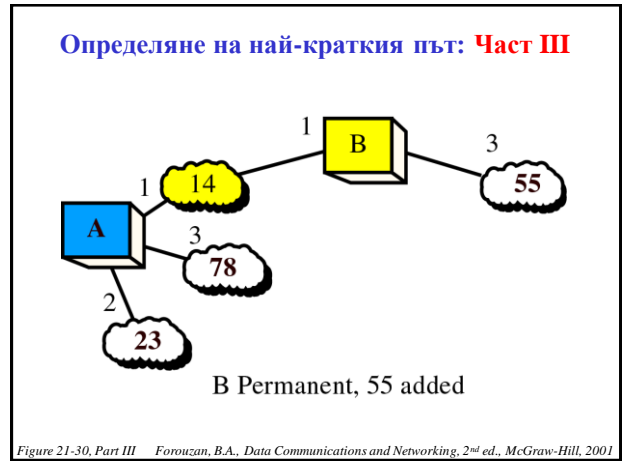
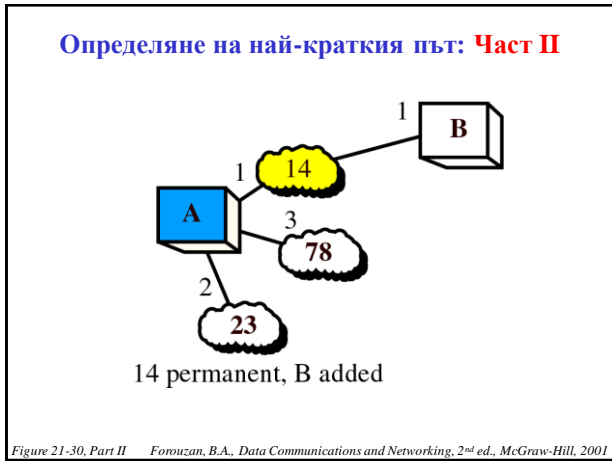
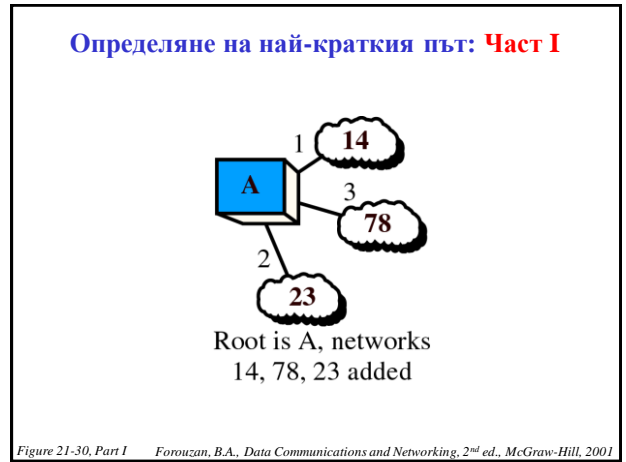
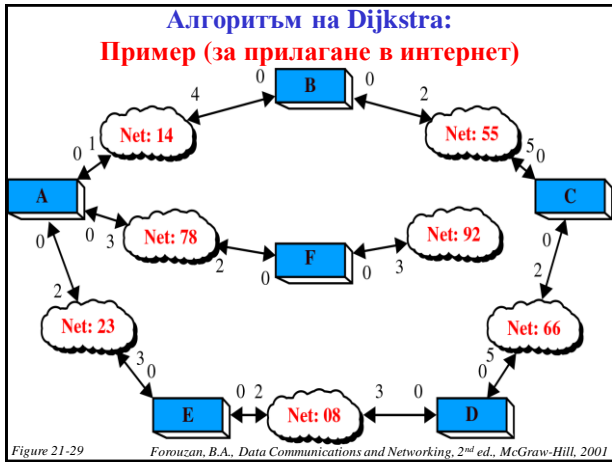
База данни за състоянието на линиите (Link State Database, LSD)

Advertiser	Network	Cost	Neighbor
A	14	1	B
A	78	3	F
A	23	2	E
B	14	4	A
B	55	2	C
C	55	5	B
C	66	2	D
D	66	5	C
D	08	3	E
E	23	3	A
E	08	2	D
F	78	2	A
F	92	3	—

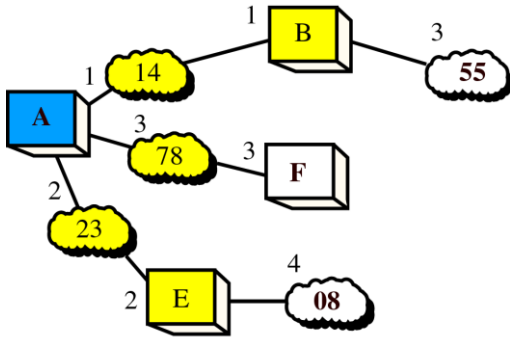
Figure 21-28

Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001





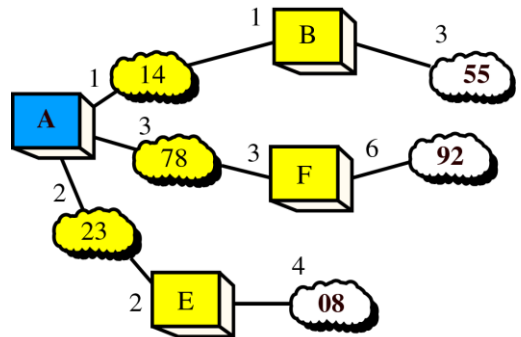
Определяне на най-краткия път: **Част VI**



78 permanent, F added

Figure 21-30, Part VI Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

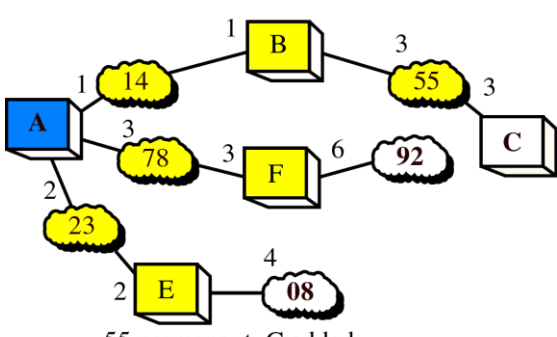
Определяне на най-краткия път: **Част VII**



F permanent, 92 added

Figure 21-31, Part VII Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

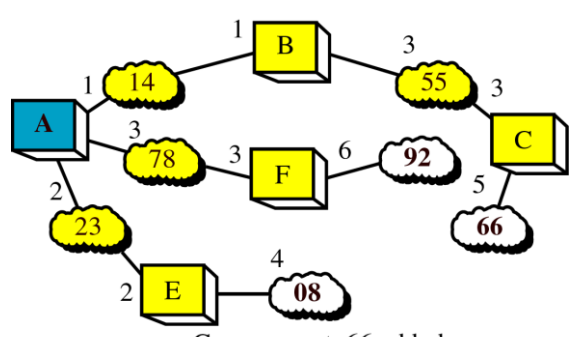
Определяне на най-краткия път: **Част VIII**



55 permanent, C added

Figure 21-31, Part I Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

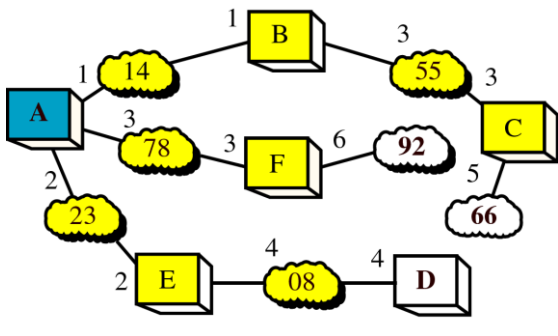
Определяне на най-краткия път: **Част IX**



C permanent, 66 added

Figure 21-31, Part II Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

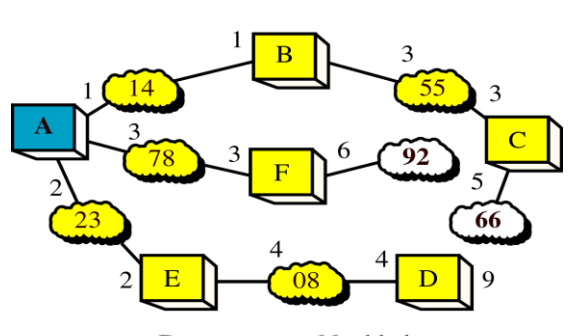
Определяне на най-краткия път: **Част X**



08 permanent, D added

Figure 21-31, Part III Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Определяне на най-краткия път: **Част XI**



D permanent, 66 added.
But $9 > 5$, so that link deleted

Figure 21-31, Part IV Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Определяне на най-краткия път: **Част XII**

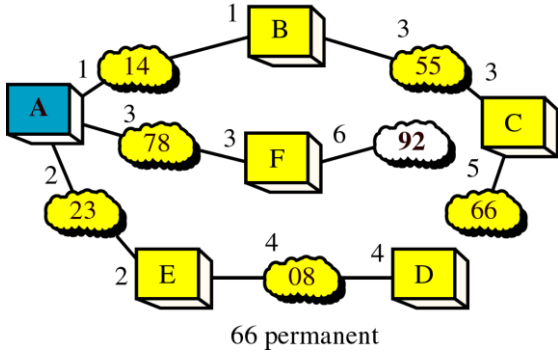


Figure 21-31, Part V Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Определяне на най-краткия път: **Част XIII**

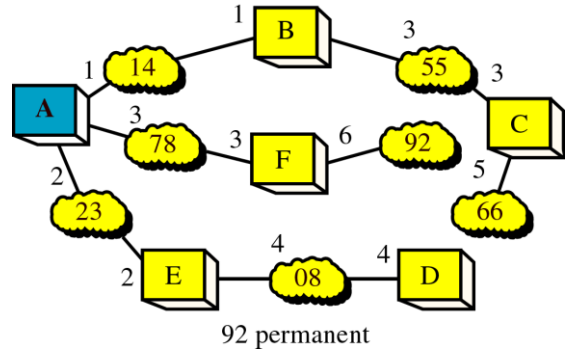


Figure 21-31, Part VI Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Окончателна маршрутизираща таблица
на маршрутизатор A

Net	Cost	Next router
08	4	E
14	1	--
23	2	--
55	3	B
66	5	B
78	3	--
92	6	F

Figure 21-32 Forouzan, B.A., Data Communications and Networking, 2nd ed., McGraw-Hill, 2001

Алгоритъм на Dijkstra:
Пример (за прилагане в мрежа,
за маршрутизатор A)

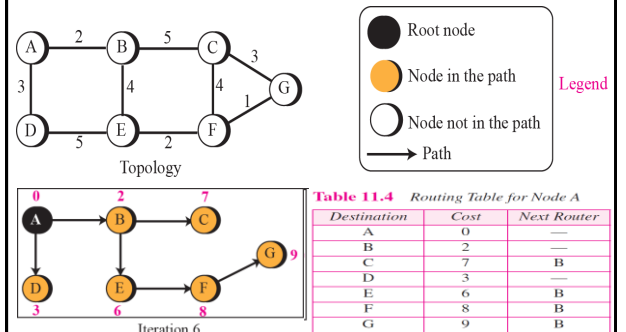


Figure 11.19

Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010

Алгоритъм на Dijkstra:
Пример (за прилагане в мрежа,
за маршрутизатор C)

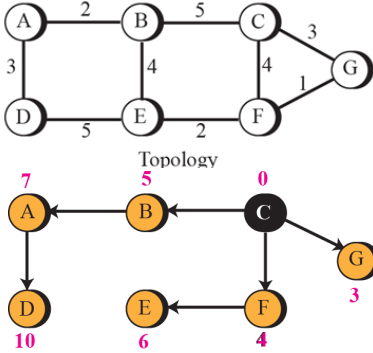


Figure 11.20 Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010